A Theory of Interactions: Unifying Qualitative and Quantitative Algebraic Reasoning*

Brian C. Williams
Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304

to appear in Artificial Intelligence

Abstract

The apparently weak properties of a purely qualitative algebra have led some to conclude that researchers 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, along with symbolic techniques for manipulating this algebra. We have constructed a hybrid algebra, called $SR1$ which allows intermediate abstractions to be selected between traditional qualitative and quantitative algebras.

$SR1$ and the symbolic algebra system Minima demonstrate substantial progress towards a theory of continuous interactions between quantities — one that allows just the interesting features of interactions to be represented, and that captures skills for composing and comparing interactions. This theory is sufficiently expressive to determine the behaviors that a variety of fluid regulation devices *will achieve* — not just what is impossible. It embodies in a simple manner many existing algebraic formalisms for describing and individually manipulating interactions, including confluences, inequalities, and monotonicity operators, as well as many of the individual inferences of qualitative arithmetic, composition of monotonicity, inequality algebra, transition analysis, qualitative resolution and traditional algebra.

*This is based on a paper that won the Artificial Intelligence Journal Best Paper Award at AAAI-88, St. Paul, MN.

1 Introduction

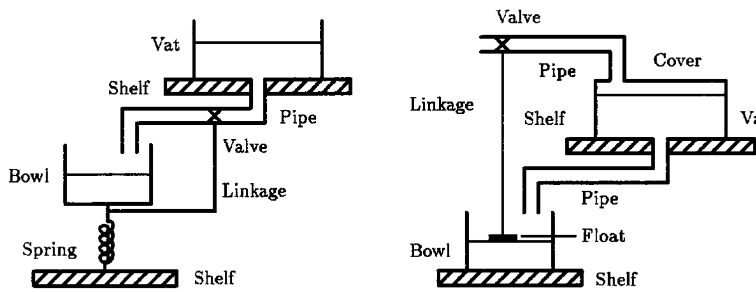

Figure 1: Do these devices, similar to ones developed by Heron (*circa* 100 A.D.) and Philon (*circa* 300 B. C.)[14], regulate the height of fluid in the bowl to a specific level? Showing that this is necessarily the case, not just possible, requires symbolic algebra on a hybrid sign/real algebra.

The research challenge of engineering problem solving, and more recently qualitative and model-based reasoning, is to develop a computational theory of the core skills underlying an engineer or scientist's ability to hypothesize, test, predict, create, optimize, diagnose and debug physical mechanisms. The ability to perform these tasks builds on general skills for answering questions like: What will the mechanism do (*prediction*), and how does it do it (*causal explanation*)? What happens if I change this parameter and why (*differential analysis*)? And what explains these observations (*interpretation*)? The task-oriented skills, such as diagnosis, and the question-oriented skills, such as causal explanation, provide two well defined layers that many researchers are trying to combine in to a computational theory. We argue that a still deeper layer needs to be unified — the level of interactions.

If we think of aggregate devices (e.g., circuits) and primitive mechanisms (e.g., components and processes) as the molecules and atoms of an overall theory, then the subatomic particles are the primitive interactions relating a mechanism's state variables. A device works by propagating changes through a network of local interactions; diagnosis involves identifying what interactions might have changed, and design involves constructing a network of interactions that achieves the desired behavior. To facilitate reasoning at the higher levels, our descriptions of interactions (e.g., represented algebraically as confluences[6] and monotonicity relations[10, 12]) attempt to highlight only the interesting features of each interaction relative to the questions at hand. The primitive skills at this subatomic level involve manipulating interactions individually or in pairs by composing, comparing and abstracting them. The coordination of these basic manipulations is guided at the higher levels to perform tasks like design verification, model abstraction and explanation.

Our goal is a theory of continuous interactions that allows just the interesting features of interactions to be represented and which captures the above composition, comparison and abstraction skills. There are two objectives: First, for tasks like design we must show how the interactions of a design combine to achieve the intended behavior. Thus to be useful the theory must be sufficiently expressive to determine what behaviors a set of interactions *will achieve* — not just what is impossible[12, 23].[1] Second, to provide a general framework for studying interactions the theory should unify in a simple manner the many existing formalisms for describing and individually manipulating interactions.

This paper presents the component of our theory of interactions that captures the algebraic properties of interactions (e.g., as opposed to aspects involving causality and time[27]). We have constructed and analyzed three hybrid algebras, called $SR1$, $PI1$ and $ZI1$, that capture qualitative and quantitative information about interactions. $SR1$ combines the signs and the reals, $PI1$ combines odd and even (parity) with the integers, and $ZI1$ combines the integers with zero and non-zero. We have implemented a qualitative symbolic algebra system based on these hybrid algebras, called *Minima*, that provides facilities for composing and comparing equations, and which builds on our earlier work on simplifying, canonicalizing and factoring equations[28]. $PI1$ and $ZI1$ are discussed primarily in [31], we concentrate here on $SR1$. Minima has been implemented on a Symbolics 3600, using Symbolics and DOE Macsyma.

With respect to our first objective, the expressiveness of our theory of interactions depends on three distinctive features of $SR1$, and Minima's ability to manipulated $SR1$ symbolically.[2] First, $SR1$ is a hybrid qualitative-quantitative algebra that merges the signs and reals. Using the reals addresses the over-abstraction problem of qualitative algebras with finite land-

[1] Qualitative simulation and envisionment over finite landmarks only exclude behaviors that are impossible, but do not guarantee that the predicted behaviors can occur (i.e., they over-abstract)[12, 23]. While these techniques aren't sufficient for verification, they may be useful for filtering out invalid designs.

[2] Researchers have incorporated quantitative operators into qualitative equations [10, 21, 27, 18, 19, 32, 13, 1] in the past. These algebras are typically used to perform arithmetic and inequality reasoning. In a few instances restricted forms of traditional quantitative algebraic manipulation have been used. Also a few systems have manipulated qualitative equations symbolically: the composition of monotonicity functions[11], the qualitative resolution rule [9, 4], QMR[18] and recently the symbolic manipulation of qualitative differential equations[5] and dimensional analysis[2]. Until recently[23, 24] work on formalizing these qualitative algebras has focused only on sign addition and subtraction.

marks. Adding signs avoids the under-abstraction problem of quantitative algebras and traditional interval algebras, which include all closed intervals over the reals (section 4). Second, $SR1$ embodies a complete sign algebra, including multiplication and division. This allows any operations over the reals to be selectively abstracted, permitting a problem solver to select the appropriate abstraction level to describe an interaction along a qualitative-quantitative spectrum (section 5). While operators over signs have been used in other reasoning systems (primarily for qualitative arithmetic), much of Minima's power is derived from exploiting unique properties of the complete sign algebra, for example, to perform simplification and factorization, or to determine equivalence and subsumption. Third, unlike other qualitative algebras[6, 26, 23, 9, 4], $SR1$ provides true equality and subset, as well as qualitative equality. The first two allow much stronger relations to be expressed than the qualitative equality of traditional sign algebras (section 6). This makes it possible to substitute equals for equals, or supersets with subsets — some of the basic rules of inference used by Minima to compose interactions. The power of qualitative symbolic algebra has been demonstrated through its central role in a theory of design from physical principles, called interaction-based design[30], and a technique, called *critical abstraction*, for generating "simplest" qualitative models from quantitative equations[29]. This paper demonstrates this use of $SR1$ and Minima, for design/verification, through a simple fluid regulation example (section 7).

Consider our second objective: unifying $SR1$ and Minima with other formalisms for describing and individually manipulating interactions. While we draw from and substantially extend our earlier work [28] on simplification, canonicalization and factoring (sections 11 and 12.1), our emphasis is on a general theory of comparison and composition (sections 12 and 13), and a formal framework for hybrid algebras (section 8). By extending $SR1$ to $=$, $\subset$ and $\approx$, it embodies a wide variety of existing representations for interactions, including pure and mixed confluences[6], inequalities and quantity spaces[10] (section 12.3), monotonicity constraints[11](appendix A), qualitative proportionalities, correspondences, direct influences [10] (see [31]), and quantitative equations. Through simplification (section 11.3) and two general composition rules — qualitative composition and the hybrid resolution (sections 12.2 and 12.3) — Minima embodies the basic rules of inference underlying qualitative and quantitative arithmetic, composition of monotonicity functions[11] (appendix A), qualitative resolution[9](section 12.2), inequality reasoning and parts of transition analysis[26, 10, 12, 6] (section 12.3), as well as the rational arithmetic, factorization and differentiation of traditional quantitative symbolic algebra[16].

There are two things our theory is not. First we do not incorporate order of magnitude[17, 25] and traditional interval algebras[15, 8, 21, 19], or the solution of differential equations. Second, we do not address the strategies used to perform sequences of algebraic manipulations (but see [30, 29]), nor do we attempt to unify the strategies used in existing approaches. Given the expressivity of $SR1$, a general but tractable equation solver seems infeasible (the problem is a superset of solving systems of non-linear inequalities). Instead we believe that strategies for guiding algebraic manipulations are most fruitfully explored in the context of particular tasks, and by exploiting the properties of particular classes of mechanisms[9, 4].

The first part of this paper (through section 7) characterizes $SR1$ and the operations of Minima, demonstrates their use, and argues for their necessity by examining the limitations of other formalisms. The remainder is a formal analysis of $SR1$ and Minima, with a discussion of how several existing formalisms are embodied.

2 Example: Regulator Design

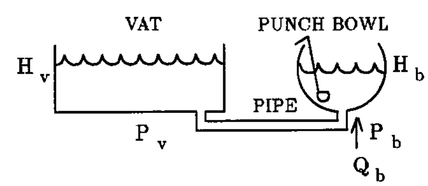

Figure 2: A Pipe used to restore the bowl level.

The work described in this paper arose from the goal of developing a model-based theory of design for continuous, lumped parameter devices [30]. This theory focuses on the early stages of the design process, where different device topologies are entertained that achieve qualitative aspects of a design specification (e.g., aspects expressible as qualitative equations or qualitative state descriptions), and concentrates on designs using new technologies where little is known beyond the models of a few simple devices. $SR1$ and Minima were developed to capture the individual steps of the type of qualitative argument evident in the following design of a fluid regulator:

> 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, thus you would like the level of the bowl to be restored automatically. That is, whenever there is a height difference between the bowl's level and what is desired, a device should automatically change the bowl height in the direction of the desired level. To construct this device you have available an array of plumbing, such as pipes, containers, floats, valves, levers, linkages and springs.
>
> You reason as follows: First, the height of the punch in the bowl can be raised or lowered by having punch flow in or out of it, either

through the top or the bottom. Second, if the desired level is specified as the height of fluid in the vat, then the punch level being too low corresponds to a difference between the vat and bowl's punch levels. Third, 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 the desired behavior can be achieved by having punch flow into the bowl whenever its pressure drops relative to the vat. Considering the available parts, you know for a pipe that fluid flows to the end that has the lower pressure. Thus a minimalist solution is to simply attach a pipe between the bottoms of the vat and bowl (figure 2).

Not excited by this solution, you observe that the bowl's pressure also corresponds to a downward force. Thus the height can also be sensed by weighing the bowl, for example, by using a spring (figure 1, left). The downward movement of the spring can then be used to proportionately open a valve that lets fluid flow into the bowl. Alternatively, the fluid level can be sensed directly with a float that controls the valve. And if the vat is closed, then its outflow can be be controlled indirectly by controlling its air inflow (figure 1, right).

Minima has been used to verify each solution.[3] The first is seemingly trivial; however, we see in section 4 that its verification is beyond the abilities of most qualitative reasoning approaches — because they tell us what is impossible[12], but not what is necessary. The example is not an artifact; it is the simplest representative of a fundamental phenomena — continuous regulation and feedback control[14]. The second two solutions are quite novel, and correspond to two of the earliest feedback control devices [14].

The qualitative vocabulary used above to describe interactions is similar to that found in the literature, and involves 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 you reasoned about the composition of interactions which are described intensionally as qualitative relations. This involved composing a desired relation with known relations to identify a new one that achieves the goal and 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[6]; for example, the exact quantitative re-

[3]During this verification each device was described using a lumped-parameter model with a single operating region. Detailed geometry and kinematic constraints were not considered.

lationships among fluid density, height and pressure must be known to determine that height and pressure difference have the same sign.

The latter is an issue of representational expressivity, and the former is one of inferential power. Given these two observations, capturing this reasoning process requires hybrid qualitative-quantitative algebras for describing interactions coupled with a theory of the designer's algebraic skills for manipulating these interactions. These are the two elements of our theory of interactions. The next three sections address the concern of expressivity — why the hybrid algebra $SR1$ is necessary — turning then to the issue of inferential power and the operations that Minima must support.

3 Qualitative Algebras

The idea behind most qualitative reasoning research[3, 25] is that engineers and scientists do not reason exclusively about a device in terms of the precise values and interrelationships between parameters. Rather they often reason about these values and interrelationships at a more abstract level in terms of particular features of interest, called qualitative values. Abstract interrelationships between parameters are specified as equations in a qualitative algebra — an algebra whose domain consists of a set of qualitative values and whose operators are abstractions of operators on the reals. We adopt the convention of placing a hat over qualitative values, as in $\hat{+}$ and circles around qualitative operators, as in $\oplus$.

The most common abstraction of values used in qualitative reasoning is the sign of a quantity, $\mathcal{S} \equiv \{\hat{+}, \hat{0}, \hat{-}\}$, used to describe the direction that quantities move in a continuous physical system. For example, the phrase "the ball is rising" is captured by the equation "$[V_{ball}] \approx \hat{+}$," where square brackets ("[ ]") abstract a quantity to its sign, and $\approx$ denotes qualitative equivalence (defined in a moment).

More precisely, $\Re$ denotes the reals, and $\hat{-}$, $\hat{0}$ and $\hat{+}$ denote three intervals $(-\infty, 0)$, $[0, 0]$ and $(0, +\infty)$, respectively. The sign of a real is determined by the operator $[\,]$ (i.e., given $x \in \Re$, then $[x] \in \mathcal{S}$ and $x \in [x]$). $\mathcal{S}' \equiv \{\hat{-}, \hat{0}, \hat{+}, \hat{?}\}$ extends $\mathcal{S}$ with the interval $\hat{?} \equiv (-\infty, +\infty)$, used to denote an ambiguous sign.

Pure confluences is the most commonly studied example of an algebra on signs [23, 9, 4]. It consists of sign addition ($\oplus$) and sign subtraction ($\ominus$), related by qualitative equality ($\approx$). Intuitively the expression $[x] \oplus [y]$ answers the question "what is the sign of $x + y$ given only the signs of x and y?" If x and y are positive, then their sum is also positive; on the other hand, if x and y have opposite signs then the result is ambiguous ($\hat{?}$) — the sum could be any real value: positive, negative or zero. $\ominus$ is analogous:

$\oplus$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{-}$	$\hat{-}$	$\hat{-}$	$\hat{?}$	$\hat{?}$
$\hat{0}$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{+}$	$\hat{?}$	$\hat{+}$	$\hat{+}$	$\hat{?}$
$\hat{?}$	$\hat{?}$	$\hat{?}$	$\hat{?}$	$\hat{?}$

$\ominus$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{-}$	$\hat{?}$	$\hat{-}$	$\hat{-}$	$\hat{?}$
$\hat{0}$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{+}$	$\hat{+}$	$\hat{+}$	$\hat{?}$	$\hat{?}$
$\hat{?}$	$\hat{?}$	$\hat{?}$	$\hat{?}$	$\hat{?}$

To cope with this ambiguity confluences uses the weaker relation $\approx$, rather than $=$, where $e1 \approx e2$ and is satisfied for an assignment if $e1$ and $e2$ denote the same sign, or if either is ambiguous ($\hat{?}$) — equivalently, they denote overlapping intervals. For example, $[x] \oplus [y] \approx \hat{0}$ is satisfied when $\langle x, y\rangle$ is $\langle \hat{0}, \hat{0}\rangle$, $\langle \hat{+}, \hat{-}\rangle$ or $\langle \hat{-}, \hat{+}\rangle$. Ambiguity is the predominant source of weakness in a sign algebra, and has been a major focus of research in qualitative reasoning. We explore an example of this problem in the next section.

4 Limitations of Qualitative Algebras and Other Formalisms

Sign algebras, such as confluences, are not sufficiently expressive to describe the design tasks we are considering. A simple example from the punch bowl problem highlights the basic point. The height of fluid in the punch bowl can be restored to the vat's fluid height by making the direction of bowl height change be in the same direction (i.e., have the same sign) as the vat/bowl height difference. This desired behavior is captured by the following equation, where H_b and H_v denote the height of punch in the bowl and vat, respectively:

$$[dH_b/dt] = [H_v - H_b]$$

Unfortunately this cannot be express using a sign algebra like confluences — the equation involves real subtraction ($[H_v - H_b]$), but we have only sign addition ($\oplus$) and subtraction ($\ominus$). Also the equation uses equality ($=$), but we only have qualitative equality ($\approx$). Instead the best we can do is the following equation, which replaces real subtraction with sign subtraction, and equality with qualitative equality:

$$[dH_b/dt] \approx [H_v] \ominus [H_b]$$

However, this does not express at all what we want. For example, most of the time both the height of the vat and height of the bowl will be positive (i.e., $[H_v] \ominus [H_b] \Rightarrow \hat{+} \ominus \hat{+}$). And the difference of two positives is ambiguous; thus, $[dH_b/dt]$ can take on any value in $\mathcal{S}'$ — most of the time the equation provides no constraint. This means that using a sign algebra alone we can't even express the punch bowl problem precisely, let alone solve it. The basic difficulty is that the use of a sign algebra forces us to over abstract — it forced us to replace $[H_v - H_b]$ with $[H_v] \ominus [H_b]$.[4]

To overcome this type of problem a number of attempts have been made to strengthen sign algebras by introducing a finite set of landmarks into a qualitative representation. A landmark is a value that breaks the real number line into regions of interest. For example, the sign algebra uses 0 as a landmark, breaking the number line into $\hat{+}$, $\hat{-}$ and $\hat{0}$. Forbus[10] introduced the notion of a quantity space that allows a qualitative representation to be described in terms of any finite number of landmarks, or equivalently in terms of any number of subintervals, while Williams[26] introduced a qualitative representation that breaks the space of values a set of quantities can take on (i.e., *state space*) into a set of open regions separated by boundaries. Kuipers[11] developed techniques for dynamically introducing landmarks during the simulation process.

Unfortunately, Struss[23] showed that the introduction of a finite number of landmarks cannot overcome certain fundamental weaknesses, such as the ambiguity of addition and the lack of an additive inverse. Intuitively a qualitative algebra is often viewed as an interval algebra [6, 23]. Introducing a finite number of landmarks breaks the reals into a finite number of intervals. For this case Struss [23] shows that, given only the intervals that two quantities lie within, one cannot always determine a single minimal interval that the sum lies within. The problem is analogous for subtraction, and therefore there is no additive inverse. In the design example subtraction is used; thus we cannot verify that the solution necessarily achieves the desired behavior with a finite interval algebra.

We could use a more traditional interval algebra with an infinite domain[15]. This is the basis for a wide variety of interval bounding systems.[5] But then we will have to do symbolic algebra to verify the design problem — no numbers have been specified, and the infinite domain prevents us from using the standard approach of enumerating the cases (as in envisionment and qualitative simulation).

Even symbolic manipulation won't help when the intervals have nonzero width. In [15] Moore observes that an interval algebra has an additive inverse only for intervals with zero width. One consequence is that there is no distributive law for these traditional interval algebras:

[4] One might argue that in this specific instance we can get around the problem by replacing $(H_v - H_b)$ with a variable denoting height difference $Hd_{v,b}$. In fact, as de Kleer and Brown suggest (page 22, [6]), we can arbitrarily choose to replace any expression with a variable. But this just defers the problem — how then do we express the requirement that $Hd_{v,b} = H_v - H_b$? If we don't express this constraint then we have no guarantee that the variable conveys the content of the expression.

[5] See [19] for a unification and survey of many of these earlier approaches, and [21] for its combination with "qualitative arithmetic" techniques, such as sign arithmetic, inequality arithmetic and restricted algebraic simplification.

$$x \times y + x \times z \not\Rightarrow x \times (y + z)$$

(consider $x = [1, 2]$, $y = 1$ and $z = -1$)[15]. In the design problem this law is essential to verifying the design solution given separate models for the pipe and each container (see the step from D18 to D19 in the trace of section 7).

Another problem is that signs are not represented in a standard interval algebra[15] since the intervals must have a finite width. Simply expanding the domain of the algebra won't help. If unbounded intervals are introduced into the algebra, then the additive cancellation law is lost, which is the basis of most interval analysis techniques (e.g., Gaussian elimination on interval equations[15]). Furthermore, our verification does not depend on the properties of interval equations in general, but those specific to (and peculiar to) the subdomain of signs — that is we must exploit the unique algebraic properties of the sign algebra. If we must generalize the equations to any intervals, little insight is provided.

The primary approach remaining is order of magnitude reasoning[17, 25]. Order of magnitude relations replace the crisp boundaries of a quantity space with far and close, through relations like "much bigger" and "close". In our example no order of magnitude relations were explicitly stated. Furthermore, in general regulation and feedback control depend on sensing what we might normally consider "negligible" deviations, thus crisp boundaries are required.

More important than telling us what won't work, the above analysis tells us what is needed. First, we need an algebra whose domain includes signs to state the problem and reals to provide the necessary precision. Second, since the domain of the resulting algebra is infinite, we need to perform symbolic algebra. Third to exploit the properties afforded only by precise values, we need to be able to restrict ourselves to the real (sub) algebra. Finally, to express relations between signs and to exploit their specific properties we need an abstraction operator to signs, which restricts us to the sign (sub)algebra and links us to precise values. These are exactly the four properties that characterize $SR1$ and Minima. Together they differentiate our approach from earlier work.

5 SR1 — A Hybrid of Quantitative and Qualitative Algebras

In [28] we introduced a hybrid of the real and sign algebras called $Q1$. We have since reformulated $Q1$ as a single algebra, called $SR1$, with a combined domain of signs and reals.[6] $SR1$ is defined precisely in section 8, along with

[6]The new name is also intended as a more precise indication of the type of hybrid (signs and reals), and to avoid confusion with Berleant and Kuiper's Q2 and Q3[13, 1],

several related qualitative algebras and hybrids (e.g., $S1$, $S2$, $P1$, $Z1$, $SR2$, $PI1$ and $ZI1$). The next few sections highlight the features of $SR1$ and Minima. For this a useful fiction is to think of $SR1$ as two distinct algebras linked by [ ].

Expressions in $SR1$ are composed of the standard operators on $\Re$, consisting of $+$, $-$, $\times$ and $/$, together with the analogous operators on $\mathcal{S}'$ ($\oplus$, $\ominus$, $\otimes$, $\oslash$), and [ ], which maps expressions on $\Re$ to $\mathcal{S}'$. $SR1$ expressions denote intervals and points on $\Re$. $SR1$ equations relate expressions through strict equality ($=$), subset ($\subset$) and qualitative equality ($\approx$). The following are tables for $\otimes$ and $\oslash$, where U denotes undefined values for $\oslash$ (when dividing by an interval containing zero):

$\otimes$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{-}$	$\hat{+}$	$\hat{0}$	$\hat{-}$	$\hat{?}$
$\hat{0}$	$\hat{0}$	$\hat{0}$	$\hat{0}$	$\hat{0}$
$\hat{+}$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{?}$	$\hat{?}$	$\hat{0}$	$\hat{?}$	$\hat{?}$

$\oslash$	$\hat{-}$	$\hat{0}$	$\hat{+}$	$\hat{?}$
$\hat{-}$	$\hat{+}$	U	$\hat{-}$	U
$\hat{0}$	$\hat{0}$	U	$\hat{0}$	U
$\hat{+}$	$\hat{-}$	U	$\hat{+}$	U
$\hat{?}$	$\hat{?}$	U	$\hat{?}$	U

Because $SR1$ includes strict equality and the standard real operators, as well as sign operators, it allows us to express equations like:

$$[dH_b/dt] = [H_v - H_b]$$

which accurately describes the desired behavior for the punch bowl problem. Additionally, most representations of interactions used in qualitative formalisms are expressible within $SR1$ (in some cases by embedding $SR1$ into a simple sorted, first order theory[30]). These include confluences, inequalities, quantitative equations, qualitative proportionalities, correspondences, and strict monotonicity. Three types of relations used in other qualitative reasoning systems, but not included within $SR1$ are transcendental functions[20, 19], bounded interval equations[15] and order of magnitude relations[17, 25].

$SR1$ overcomes the over-abstraction problem of qualitative algebras with finite landmarks by using the infinite resolution of the reals for selected subexpressions of equations. Of course this problem is also solved using a strictly quantitative algebra, but using only the reals presents the inverse problem of under-abstraction — the initial motivation behind qualitative algebras. Thus the resolution of the reals should only be used in those cases where it is necessary in order to avoid an ambiguity that matters with respect to the task at hand.

$SR1$ enables this by permitting statements to be made at different levels of abstraction along a qualitative/quantitative spectrum; consider how this

which are inference techniques, rather than algebras.

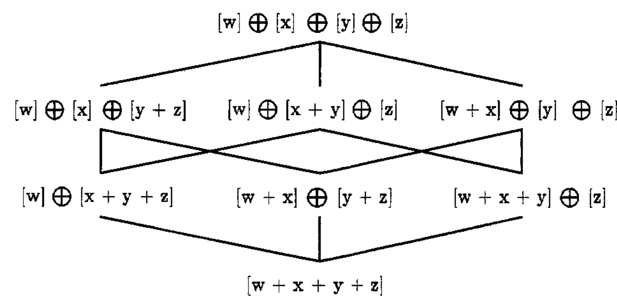

Figure 3: Expressions in $SR1$ become successively more abstract as real operators are replaced with sign operators. Above is shown just a few of the abstractions of $w + x + y + z$.

is the case. Sign operators are more abstract than real operators. Thus if we take a real expression and start replacing its operators with sign operators, the expression becomes successively more abstract (figure 3).[7]

Conversely expressions with more real operators decrease the opportunity for ambiguity to arise by delaying the point at which they move $\Re$ to S'. For example, in figure 3 suppose that w and x are positive. Then for the top expression, an ambiguity arises if either y or z is negative. On the other hand, the far left expression on the second line only introduces an ambiguity if, additionally, the sum of y and z is negative. Furthermore, for the bottom expression there is no opportunity for an ambiguity to arise. By making a judicious selection of sign and real operators we can select the exact level of abstraction appropriate for a task. Ideally we would like to select the greatest abstraction of a set of interactions that is still sufficient to determine the qualitative information required for this task. In [29] we describe a technique, called *critical abstraction*, which uses Minima to generate these greatest abstractions from more detailed or quantitative models. This process of identifying just what is relevant is crucial to qualitative reasoning tasks, such as generating qualitative representations, model construction, data interpretation, explanation, design debugging and teleological description.

[7]One should not think that there are only two abstraction levels for expressions — signs and reals. Since an expression denotes a function, raising the function's image for any particular value from reals to signs makes the function more abstract.

6 Equality and Subset in SR1

Equations in $SR1$ have a very different semantics from those in other qualitative algebras[6, 23, 9, 4]. This arises from the fact that many qualitative algebras except $SR1$ use only qualitative equality ($\approx$). The result for these other algebras is that substitution of equals for equals no longer applies (*e.g.*, $a \approx b, b \approx c \not\Rightarrow a \approx c$) — the basic operation for combining equations in traditional algebras.

Recall that an equation involving qualitative equivalence is satisfied if the two expressions denote intersecting intervals. For example, the equation $[x] \oplus [y] \approx \hat{0}$ is satisfied if $[x]$ is $\hat{+}$ and $[y]$ is $\hat{-}$ (since $\hat{+} \oplus \hat{-} = \hat{?}$ intersects $\hat{0}$). This is intuitive as an abstraction of $x + y = z$, since a positive and negative can sum to 0.

In confluences, the nonstandard treatment of equality has the serious consequence that equality is no longer transitive; for example:

$$\hat{+} \approx \hat{?},\ \ \hat{?} \approx \hat{-} \ \ \not\Rightarrow \ \ \hat{+} \approx \hat{-}$$

in other words, two intervals a and c intersecting a common interval b does not suggest that a and c intersect.[8] The serious consequence, mentioned above, is that *in confluences we can't always substitute equals for equals.* The problem again is one of over-abstraction. Interval intersection is much weaker than interval equivalence. Furthermore, it is much weaker than necessary. Our experience is that for all qualitative equations used in our design examples the much stronger interpretation of equality applies. Thus one should not be forced to introduce $\approx$ as long as stronger relations hold.

$SR1$ avoids this problem by allowing strict equality ($=$) to be used to relate expressions denoting real values, (e.g., $a + b = c$), and signs (e.g., $[a + b] = [c]$). For example, $[x] \oplus [y] = \hat{0}$ is satisfied only if x and y are $\hat{0}$. Additionally strict equality allows us to map inequalities to $SR1$ equations:[9]

$$e1 < e2 \Leftrightarrow [e2 - e1] = \hat{+}$$

allowing quantity spaces[10, 21, 12] to be expressed in $SR1$.

$SR1$ includes interval subset ($\subset$) and intersection ($\approx$) for cases where weaker relations are required. Subset is used to specify that one expression (the superset) is an abstraction of another. $\approx$ is used when both sides of an equation are abstracted, and sign addition or subtraction (not just negation) have been introduced on both sides. This paper analyzes all three relations, initially focusing on manipulations of equations involving $=$.

[8] However, in the special case that a, b and c are in $\mathcal{S}$ (i.e., are not $\hat{?}$), then $\approx$ is an equivalence relation[9], since none of the elements of $\mathcal{S}$ intersect.

[9] The inequality $e1 < e2$ implies the confluence $[e2] - [e1] \approx \hat{+}$, but the inequality is not implied by the confluence.

7 Symbolic Algebra and its Role in Design

We now turn from the issue of expressivity to inferential power. An important use of algebraic reasoning in the context of design is to verify that a proposed device achieves a specified behavior. We use verification to identify the requisite manipulations on $SR1$ equations, and argue that, while strictly qualitative reasoning is too abstract for verification, strictly quantitative reasoning pushes verification too late in the design process, forces over commitment, and results in analyses with an overly narrow range of applicability. Thus hybrid symbolic reasoning is essential even when full quantitative models are available.

To verify the first design in the punch bowl example we must show that using the bowl/vat/pipe combination (figure 2) achieves the desired effect of having the height of the bowl change in the direction of the height difference between the bowl and vat ($[H_v - H_b] = [d/dt(H_b)]$). Verification involves computing the composite behavior of the component interactions, and showing that the composition necessarily implies the desired behavior. That is, the desired behavior *subsumes* the composite.

Computing the composition depends on how the component interactions are described. If the interactions produced by the components and connections are all described quantitatively, then traditional techniques — Gaussian elimination for linear systems and Grobner basis for non-linear — can be used to solve the corresponding system of equations in terms of the variables mentioned in the desired behavior.

In our example using Gaussian elimination on the standard linear models of pipes and containers produces:

$$H_v - H_b = d \times g \times R_p \times A_b \times d/dt(H_b)$$

where d is fluid density, g is the gravitational constant,R_p is fluid resistance of the pipe, and A_b is the cross sectional area of the bowl. Next, we show that the desired behavior subsumes this equation. Given that d, g, R_p and A_b are positive constants, it follows from the properties of $SR1$ that their product will also be positive, and thus $[H_v - H_b] = [d/dt(H_b)]$. In this case the additional operation that Minima must provide is subsumption of $SR1$ equations.

Of course, if a complete quantitative description is unavailable some qualitative descriptions become necessary. But *even though quantitative models are available*, there are important reasons to describe some (although not all) component behaviors qualitatively. Most important, it allows a design system to rule out or validate large portions of the design space during the early stages of conceptual design. That is, the abstraction provided by qualitative equations allows us to construct models each of which characterizes a much larger class of components than possible with quantitative equations alone. For example, the relation $dH_b/dt = Q_b/A_b$ holds roughly for the narrow case of containers with straight, vertical sides, while the qualitative relation $[dH_b/dt] = [Q_b]$ holds for almost any container. During verification this additional abstraction allows our design system (Ibis) to validate or rule out a much larger portion of the design space in a single analysis, than possible from quantitative component descriptions alone. For example the verification shows that the punch bowl solution works independent of the non-linearities of the containers and pipe.

A secondary rationale is computational efficiency. One of the big computational costs for quantitative symbolic algebra is intermediate expression bulge — the expressions, when maintained in canonical form, grow very quickly in size[22]. Later we show that the canonical form for qualitative expressions is significantly more compact, thus reducing the problem of expression bulge.

Turning to Minima's capabilities, ideally we would like to develop a complete equation solver for $SR1$, in the spirit of Gaussian elimination and the Grobner basis techniques. But this seems infeasible, given the expressiveness of $SR1$.[10] Instead we build into Minima basic skills for pairwise composing, comparing and testing $SR1$ equations. These are used by higher level reasoning skills[30, 29] in a task specific manner to perform equation solving in a way that is sound, although likely incomplete. While we do not consider here how sequences of manipulations should be performed, we do analyze the completeness of the pairwise operations.

Next we use the design example to make these operations concrete. In the example we determined from the interactions produced by the bowl and vat that the desired interaction, $[H_v - H_b] = [d/dt(H_b)]$, could be achieved by having the flow into the bottom of the bowl be in the direction of the vat/bowl pressure difference, $[P_v - P_b] = [Q_b]$. This equation replaces height with pressure and height change with fluid flow in the earlier equation. Each replacement is the result of combining the desired interaction with one from the bowl or vat, by using substitution of equals for equals. To accomplish this, Minima carries out the following steps: 1) identify a shared variable (or subexpression) between two equations, 2) isolate a variable/subexpression on one side of one of the equations, 3) substitute the result into the other equation, and 4) simplife the combined result.

For example, to replace H_b with V_b on the right side of:

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

Minima uses $V_b = H_b \times A_b$, with shared variable H_b. Isolating H_b in the second equation produces[11] $H_b = V_b/A_b$. Substituting the result into the

[10] The problem is a superset of solving systems of non-linear inequalities.

[11] We say isolating rather than solving for a variable in an equation to avoid confusion with solving systems of equations.

first equation produces:

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

To simplify Minima uses the fact that A_b is a constant to pull it outside the differentiation. A_b is then eliminated using the fact that A_b is positive, and dividing a quantity by a positive number doesn't change its sign:

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

This result is combined with other interactions in a similar manner.

For composition, because we use strict equality $(=)$ the first[12] and third steps become straightforward. Minima needs to support the steps of variable (or expression) isolation and simplification, both of which require factorization.

The following shows the remaining sequence of compositions that leads to the first design. Dn, Fn, and Cn denote Desired relations, given Facts and Consequences, respectively:

D1)	$[H_v - H_b] = [d/dt(H_b)]$	Original Desired Relation
F2)	$H_b \times A_b = V_b$	Container Model
C3)	$H_b = V_b/A_b$	Isolate H_b in F2
D4)	$[H_v - H_b] = [d/dt(V_b/A_b)]$	Substitute for H_b in D1 using C3
D7)	$[H_v - H_b] = [d/dt(V_b)/A_b]$	Differentiate D4
D8)	$[H_v - H_b] = [d/dt(V_b)]/[A_b]$	Simplify D7
F9)	$[A_b] = \hat{+}$	Container Model
D10)	$[H_v - H_b] = [d/dt(V_b)]$	Substitute for A_b in D8 using F9
F11)	$Q_b = d/dt(V_b)$	Container Model
D12)	$[H_v - H_b] = [Q_b]$	Substitute for $d/dt(V_b)$ in D10 using F11
F13)	$P_v = d \times g \times H_v$	Container Model
C14)	$H_v = P_v/m \times g$	Isolate H_v in F13
D15)	$[P_v/d \times g - H_b] = [Q_b]$	Substitute for H_v in D12 using C14
F16)	$P_b = d \times g \times H_b$	Container Model
C17)	$H_b = P_b/d \times g$	Isolate H_b in F16
D18)	$[P_v/d \times g - P_b/d \times g] = [Q_b]$	Substitute for H_b in D15 using C17
D19)	$[(P_v - P_b)]/([d] \times [g]) = [Q_b]$	Simplify D18
F20)	$[d] = \hat{+}$	Property of fluids
F21)	$[g] = \hat{+}$	Property of gravity
D22)	$[(P_v - P_b)]/(\hat{+} \times \hat{+}) = [Q_b]$	Substitute g, d into D19 using F20,F21
D23)	$[P_v - P_b] = [Q_b]$	Simplify D22

[12] Actually, identifying shared variables is a non-trivial task in our design system, Ibis, since syntactically distinct terms are allowed to denote the same variable. In our approach equivalent variables are determined through congruence closure on functions and 1-1 functions, and unifiable variables are determined through terminological reasoning [30]. This terminological reasoning is an important element of a theory of interactions not discussed here.

Notice that lines D12 through D23 determine that height and pressure difference have the same sign. This critically depends on the fact that both containers have the same density of fluid, and that they are influenced by the same amount of gravitational acceleration. This can only be inferred through the precise quantitative relationship between pressure and height. Also note that distributivity is used to go from line D18 to D19. During our discussion of other qualitative formalisms (section 4) we observed that bounded interval algebras could not be used on this problem since they don't have a full distribution law. The final equation D23 enables the crucial insight — the close similarity between D23 and a pipe's behavior (i.e., $[P_{t1} - P_{t2}] = [Q_{t2}]$ where $t1$ and $t2$ are the ends of a pipe) suggests connecting a pipe between the two containers. The devices of figure 1 are verified in an analogous process.

In addition to simplification and composition, Minima supports tests on expressions and equations for subsumption, equality, tautology and inconsistency. We have already seen subsumption and equality used to compare the composite and desired behaviors of a design. These operations are also used to recognize redundant and equivalent results. Tests for tautology (e.g., $[a] \approx [a]$) are used to identify trivial results, and tests for inconsistent equations (e.g., $[a^2] = \hat{-}$) are used to identify inconsistent models. Each of these tests is facilitated by reducing the equations to a (pseudo) canonical form.

Having presented the intuitions and motivations behind hybrid algebras and their symbolic manipulation the remainder of this paper turns to the formalization and analysis of $SR1$, and the development of procedures for the operations discussed above.

8 Definitions of Qualitative and Hybrid Algebras

Generalizing from section 3, a qualitative algebra is an abstraction of a quantitative algebra, such as the standard algebra on the reals.[13] The qualitative algebra's domain consists of a set of qualitative values each of which specifies ranges of possible values that a quantity lies within. A qualitative value can be any subset of the quantitative domain. Each qualitative operator, qop, is an abstraction of a quantitative operator, op. Given qualitative values specifying the range of op's operands, qop determines the qualitative value that most precisely describes the range of op's result. A hybrid algebra is a type of qualitative algebra that includes the full set of quantitative values in its domain, thus allowing precise relations to be stated. The rest of this section makes these concepts precise and introduces the domains and algebras used later in our analysis.

[13]Formulations that characterize qualitative algebras as restricted interval algebras on the reals can be found in [23] and [24].

While qualitative algebras are generally thought of and used in practice as interval algebras over $\Re$ [6, 23, 25], this need not be the case. For example Minima uses two qualitative algebras on the integers ($\mathcal{I}$), which we mentioned earlier are used to represent the qualitative features of symbolic exponents of $SR1$ expressions. One algebra uses odd and even as qualitative values (called the parity of the integer), and the second uses zero and non-zero. Odd, even and non-zero are not contiguous intervals, and neither domain ranges over $\Re$. Our formulation generalizes qualitative algebras to be abstractions of any algebra, and qualitative values to be subsets, rather than subintervals, of a quantitative domain.

Definition 1 *Let D be a non-empty set of elements. Then a* qualitative value, v, on D *is a non-empty subset of D. A* qualitative domain, Q, over D *is a set of qualitative values on D. D is called the* quantitative domain *of v or Q, respectively.*

$\mathcal{S}'$, defined in section 3, is the most common example of a qualitative domain on $\Re$. For parity (odd/even) we construct a qualitative domain $\mathcal{P} = \{\hat{o}, \hat{e}\}$ on $\mathcal{I}$, where $\hat{o} = \{2i + 1 | i \in \mathcal{I}\}$ and $\hat{e} = \{2i | i \in \mathcal{I}\}$. We construct a third domain (zero/non-zero), $\mathcal{Z} = \{\hat{0}, \hat{\emptyset}, \hat{?}\}$ on $\mathcal{I}$ where $\hat{0} = \{0\}$, $\hat{\emptyset} = \{i | i \in \mathcal{I}, i \neq 0\}$ and $\hat{?} = \mathcal{I}$.

Next consider qualitative arithmetic. In the following definition "range of the operator's result" is captured using image. "Most precise qualitative value" that describes the operator's range is captured using least upper bound under subset; that is, the smallest superset containing the range.

Definition 2 *Let Q be a qualitative domain over D, and op be an n place (arithmetic) operator on D, then the* qualitative (arithmetic) operator for op on Q *is the n place operator, qop on Q, such that $qop(q_1, \ldots, q_n)$ are the least upper bounds under subset of the qualitative values in Q that contain the image $op(q_1, \ldots, q_n)$. op is called the* quantitative operator of qop.

Definition 3 *Given an algebra $A = \langle D, op_1, \ldots, op_n \rangle$ and a qualitative domain Q over D, the* qualitative algebra for A on Q *is $QA = \langle Q, qop_1, \ldots, qop_n \rangle$, where each qop_i is the qualitative arithmetic operator for op_i on Q.*

The sign algebra discussed earlier, $\langle \mathcal{S}', \oplus, \otimes \rangle$, is the qualitative algebra for $\langle \Re, +, \times \rangle$ on $\mathcal{S}'$. Our analysis includes several qualitative algebras on $\mathcal{S}'$ — the qualitative algebra $S1$, for $\langle \Re, +, \times \rangle$ on $\mathcal{S}'$; $S2$ for $\langle \Re, \times \rangle$ on $\mathcal{S}'$, $S3$ for $\langle \Re, + \rangle$, $S4$ for $\langle \Re, +, - \rangle$, and confluences for $\langle \Re, +, - \rangle$. We also use $P1$ for $\langle \mathcal{I}, +, \times \rangle$ on $\mathcal{P}$, and $Z1$ for $\langle \mathcal{I}, +, \times \rangle$ on $\mathcal{Z}$ (analyzed in [31]). We refer to the traditional algebras $\langle \Re, +, \times \rangle$, $\langle \Re, \times \rangle$ and $\langle \Re, + \rangle$ as $R1$, $R2$ and $R3$, respectively.

An important feature of the three general definitions is that a quantitative algebra is simply a limiting case of a qualitative algebra. That is, given a quantitative domain D, its singleton subsets $\hat{D} = \{\{d\}|d \in D\}$ form a qualitative domain on D. Any algebra on D is isomorphic to its qualitative algebra on $\hat{D}$, by mapping between each singleton set and its element. We often blur the distinction between singleton sets and their elements, as well as their corresponding domains and algebras. The consequence of this observation is that we can build hybrids by simply including the quantitative domain in our qualitative domains, and letting the qualitative operators range over this larger domain. To this end we make the following distinctions:

Definition 4 $\hat{D} = \{\{d\}|d \in D\}$ *is the* strictly quantitative domain on D. Q *is a* strictly qualitative domain on D *if* $\hat{D} \not\subset Q$. Q *is a* hybrid domain on D *if* $\hat{D} \subset Q$, *and* $Q \neq \hat{D}$. *If* Q *is a strictly qualitative domain on* D, *then the* hybrid domain of Q with D *is* $QS = Q \cup \hat{D}$.

Minima uses three hybrid domains — $\mathcal{S}'\Re$ is the hybrid domain of $\mathcal{S}'$ with $\Re$, $\mathcal{PI}$ is $\mathcal{P}$ with $\mathcal{I}$ and $\mathcal{ZI}$ is $\mathcal{Z}$ with $\mathcal{I}$.

The most essential element of a qualitative algebra is the abstraction operator [ ]. Which extracts the features of interest from a more detailed description. [ ] is traditionally taken to be a mapping from $\Re$ to $\mathcal{S}$. But sometimes we want to abstract from other quantitative domains, such as $\mathcal{I}$. Sometimes we want to map from a qualitative domain to a more abstract one, such as from $\mathcal{S}'$ to $\mathcal{Z}$. Sometimes we want to map from a domain to its subset, such as $\mathcal{S}'\Re$ to $\mathcal{S}'$. Or more generally, we may want to map between any two qualitative domains that have the same quantitative domain. In this more general form [ ] takes a qualitative value in one domain, and produces the most specific value in the second domain that contains it.

Definition 5 *Let* P *and* Q *be qualitative domains over some* D, *and* v *be a qualitative value in* P, *then the* abstractions of v in Q *is the set* $A(v) = \{b|b \in Q, v \subset b\}$. *The* least abstractions of v in Q *are the least upper bounds under subset of* $A(v)$. *The* abstraction operator from P onto Q, *denoted* $[\,]_{P,Q}$, *maps each element in* P *to its least abstractions in* Q.

While the explicit notation for abstraction includes the domain and range as subscripts ($[\,]_{P,Q}$), typically one or both follow from the context and are left implicit.

Finally we define a hybrid algebra as a qualitative algebra over a hybrid domain, with [ ] mapping elements of the domain to elements in the qualitative subset:[14]

[14]Although the abstraction and arithmetic operators of all the algebras discussed here

Definition 6 *Given an algebra* $A = \langle D, op_1, \ldots, op_n \rangle$ *and a hybrid domain* H *of* Q *with* D*, the* hybrid (qualitative/quantitative) algebra for A on H *is* $HA = \langle H, [\,]_{H,Q}, qop_1, \ldots, qop_n \rangle$*, where* HA *excluding* $[\,]_{H,Q}$ *is the qualitative algebra for* A *on* Q*.*

Thus rather than linking qualitative and quantitative algebras with an abstraction operator, as with $Q1$[28], a single algebra is defined over a hybrid domain, and the abstraction operator maps elements of the domain to a more qualitative subset. Minima's three hybrids are $SR1$, the hybrid for $\langle \Re, +, \times \rangle$ on $\mathcal{S}'$; $PI1$, the hybrid for $\langle \mathcal{I}, +, \times \rangle$ on $\mathcal{P}$, and $ZI1$, the hybrid for $\langle \mathcal{I}, +, \times \rangle$ on $\mathcal{Z}$. Since the ambiguity of qualitative addition makes it less than useful for many problems, we introduce three corresponding *restricted hybrid algebras* $SR2$, $PI2$ and $ZI2$ which eliminate this ambiguity by restricting $\oplus$ to range only over the quantitative subdomain.

Qualitative expressions are related by $=$, $\subset$ and $\approx$, which correspond to set equivalence, subset and nonempty intersection. In addition to their standard properties, such as transitivity or intransitivity, the following properties are central to composing qualitative equations, as is discussed in section 12.2:

Proposition 1 *Let* $rel \in \{=, \subset, \approx\}$*,* qop *be an n-ary operator on* D*, and* $q, q', q_i, q'_i \in D$ *where* $i \in \mathcal{I}$ *is from 1 to n, then*

$$\begin{aligned} q_i \ rel \ q'_i \text{ for every } i &\Rightarrow qop(q_1, \ldots, q_n) \ rel \ qop(q'_1, \ldots, q'_n), \text{ and} \\ q \ rel \ q' &\Rightarrow [q] \ rel \ [q']. \end{aligned}$$

9 Properties of SR1

In this section we demonstrate the power of $SR1$ by examining some of its most important properties with respect to the operations identified earlier. $SR1$ is $\langle \mathcal{S}' \cup \Re, \oplus, \otimes, [\,] \rangle$, where $\ominus$ and $\oslash$ are defined in terms of this structure. In the remainder of the paper s, t, u, v and w denote elements of $\mathcal{S}'$ and a, b, c, d and e denote elements of $\Re$. To simplify the presentation propositions will usually be given in line, indicated only by a number (e.g., Pn), and with proofs omitted. We begin by showing how $SR1$ maps to our earlier intuition of the algebras on signs and reals linked by $[\,]$ (i.e., $Q1$). This aides in partitioning the analysis.

are closed and well-defined, this is not guaranteed by the definitions (e.g., the sign addition of the original confluences[6] satisfies the definitions but is not defined for opposing signs). A set of sufficiency conditions is given in [31].

Proposition 2 *The operators of* $SR1 = \langle \mathcal{S}' \cup \Re, \oplus, \otimes, [\,]\rangle$ *and its two subalgebras* $S1 = \langle \mathcal{S}', \oplus, \otimes\rangle$ *and* $R1 = \langle \Re, \oplus, \otimes\rangle$ *are closed and well-defined.* $\langle \Re, \oplus, \otimes\rangle = \langle \Re, +, \times\rangle$ *and* $\langle \mathcal{S}', \oplus, \otimes\rangle = \langle \mathcal{S}', +, \times\rangle$, *but* $\langle \mathcal{S}' \cup \Re, \oplus, \otimes, [\,]\rangle \neq \langle \mathcal{S}' \cup \Re, +, \times, [\,]\rangle$. *If* $a \in \Re$, *then* $[a] \in \mathcal{S}$, $a \in [a]$.

That is, [ ] computes signs as desired. The two subalgebras are closed. Thus, by restricting each variable to range over $\mathcal{S}'$ or $\Re$, expressions on "real" variables denote reals and expressions on "sign" variables denote signs. $R1$ is isomorphic to the standard algebra on reals, and the sign operators compute images of $+$ and $-$ for elements of $\mathcal{S}'$. Using this fact we replace $\oplus$ and $\otimes$ with the more familiar $+$ and $\times$ whenever talking about $R1$ and $S1$. Later, after analyzing the properties of $S1$ we show that it satisfies the sign operator tables, and thus matches our earlier description of a sign algebra. Thus a subset of $SR1$ matches $Q1$ ($S1$ and $R1$ linked by [ ]). What remains is to show when expressions in $SR1$ reduce to this form. The exceptional cases are expressions with one sign and one real as operands (e.g., $5 \oplus \hat{+}$), and abstraction applied to signs (e.g., $[[a]]$). They reduce as follows (P3–P5):

$$\begin{aligned} a \oplus s &= [a] + s, \text{ for } s \neq \hat{0} \\ a \otimes s &= [a] \times s, \text{ and} \\ [s] &= s. \end{aligned}$$

While $SR1$ contains $Q1$, they are not isomorphic, since the first equation in the above proposition does not hold when s is zero; for example, $5 \oplus \hat{0} = 5$ while $[5] + \hat{0} = \hat{+}$. Similarly relations $\subset$ and $\approx$ can operate over a mixture of reals and signs, but can be reduced to sign operands (P6,P7):

$$\begin{aligned} a \approx s &\Leftrightarrow a \subset s, \text{ and} \\ a \subset s &\Leftrightarrow [a] \subset s. \end{aligned}$$

In the next four sections our analysis of $SR1$ first explores the properties of the real and sign subalgebras ($R1 = \langle \Re, +, \times\rangle$ and $S1 = \langle \mathcal{S}', +, \times\rangle$) and then the relationship between them. The following table summarizes the field axioms of $\langle \Re, +, \times\rangle$ and the corresponding axioms of $\langle \mathcal{S}', +, \times\rangle$:

Axioms	$\langle \Re, +, \times\rangle$	$\langle \mathcal{S}', +, \times\rangle$	
Identity :	$a + 0 = a$	$s + \hat{0} = s$	(P8)
	$a \times 1 = a$	$s \times \hat{+} = s$	(P9)
Inverse :	$a + (-a) = 0$	**none**	(P10)
	$a \times (a^{-1}) = 1$ if $0 \neq a$	$s \times (s^{-1}) = \hat{+}$ if $0 \notin s$	(P11)
Commutativity :	$a + b = b + a$	$s + t = t + s$	(P12)
	$a \times b = b \times a$	$s \times t = t \times s$	(P13)
Associative :	$(a + b) + c = a + (b + c)$	$(s + t) + u = s + (t + u)$	(P14)
	$(a \times b) \times c = a \times (b \times c)$	$(s \times t) \times u = s \times (t \times u)$	(P15)
Distributivity :	$a \times (b + c) = a \times b + a \times c$	$s \times (t + u) = s \times t + s \times u$	(P16)

where unary minus $(-)$ denotes the additive inverse for $\langle \Re, + \rangle$, and a^{-1} and s^{-1} denote the multiplicative inverses of $\langle \Re, \times \rangle$ and $\langle \mathcal{S}', \times \rangle$, respectively.

Most readers are familiar with $\langle \Re, +, \times \rangle$; thus, we present the properties of $\langle \mathcal{S}', +, \times \rangle$ and $\langle \Re, +, \times \rangle$ comparatively, breaking the discussion into properties of $\langle \Re, +, \times \rangle$ missing in $\langle \mathcal{S}', +, \times \rangle$, shared properties, and properties of $\langle \mathcal{S}', +, \times \rangle$ beyond those of $\langle \Re, +, \times \rangle$.

9.1 Weaknesses of S1

As we can see from the above table, $\langle \mathcal{S}', +, \times \rangle$ satisfies most of the field axioms. The major weakness is the lack of an additive inverse for any element of $\mathcal{S}'$ except $\hat{0}$. That is, if $s \in \{\hat{+}, \hat{-}, \hat{?}\}$, there is no $t \in \mathcal{S}'$ such that $s + t = \hat{0}$ (P10). Thus "$-$" can't be the inverse of sign addition, $\langle \mathcal{S}', + \rangle$. As a result the sign algebra does not fall into any of the standard classifications of field, ring, or group.

One major consequence of not having an additive inverse is that there is no cancellation law for $\langle \mathcal{S}', + \rangle$ (P17):

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

For example, $s = \hat{+}$, $t = \hat{0}$ and $u = \hat{+}$ satisfies the left hand side but not the right. Without additive cancellation we cannot in general solve systems of sign equations by subtracting equations and canceling terms (as is done in Gaussian elimination). Furthermore, addends cannot be moved between sides of an equation (P18):

$$s + t = u \quad \not\Leftrightarrow \quad s = u - t$$

For example, consider s, t and u being positive. Consequently we cannot always isolate a particular variable in an $SR1$ equation using standard techniques developed for equations on $\Re$.[15] This means that some shared variables cannot be used to compose equations.

Furthermore, equations on $\Re$ are canonicalized by using an additive cancellation rule to move expressions to one side of an equation:

$$a = b \Leftrightarrow a - b = 0$$

However, this approach cannot be used for sign equations, because of the missing cancellation law (P19):

$$s = t \quad \not\Rightarrow \quad s - t = \hat{0}$$

[15] This is why an algebra based only on confluences (i.e., $\langle \mathcal{S}, +, - \rangle$) appears so impoverished; its operators are by far the weakest of $SR1$'s operators on $\mathcal{S}'$ and $\Re$. Momentarily we will see that the other operators are quite powerful, and that $\langle \mathcal{S}', + \rangle$ is not as weak as it first appears.

That is, two expressions having the same sign doesn't guarantee that their difference is zero. Worse the difference of common summands is almost always ambiguous (P20):

$$s - s = \hat{?} \text{ for } s \neq \hat{0}$$

Given the above, $\langle \mathcal{S}', +, \times \rangle$ presents major hurdles to operations central to both isolating variables and canonicalization.

9.2 Commonalities Between S1 and R1

In spite of a missing inverse, $\langle \mathcal{S}', +, \times \rangle$ is still quite strong because it shares most of the remaining properties of $\langle \Re, +, \times \rangle$ that support our operations of interest. $\langle \mathcal{S}', + \rangle$ has $\hat{0}$ as an identity (P8). It is commutative and associative (P12,P14) (together these make $\langle \mathcal{S}', + \rangle$ an associative, abelian monoid).

Sign multiplication, $\langle \mathcal{S}', \times \rangle$, has essentially all the properties of $\langle \Re, \times \rangle$. It has $\hat{+}$ as an identity (P8), and $\hat{0}$ as a zero (P21). It is commutative and associative (P13,P15). It has a unique inverse for each sign not containing zero (P11):

$$s \times s^{-1} = \hat{+} \text{ when } s \in \mathcal{S}', 0 \notin s$$

Thus like $\langle \Re, \times \rangle$, $\langle \mathcal{S}', \times \rangle$ has no inverse for its zero ($\hat{0}$), but it also has no inverse for $\hat{?}$ (since $0 \in \hat{?}$). This does not present a problem in practice since a subexpression denoting $\hat{?}$ provides no information. Finally, the inverse operator is / (P22):

$$s^{-1} = \hat{+}/s$$

Since $\langle \mathcal{S}', \times \rangle$ has an inverse, it has a cancellation rule (P23), analogous to $\langle \Re, \times \rangle$, and multiplicands can be moved between sides of an equation (P24):

$$\begin{aligned} s \times u = t \times u \quad &\Leftrightarrow \quad s = t \text{ for } 0 \notin u \\ s \times t = u \quad &\Leftrightarrow \quad s = u/t \text{ for } 0 \notin t \end{aligned}$$

The second property above allows us to isolate certain variables or subexpressions of qualitative equations in many situations. Also $+$ distributes over $\times$ in $\langle \mathcal{S}', +, \times \rangle$ (P16). Thus $\times$ can be lifted to the outermost expression of a sign equation, where the cancellation rule for multiplication can be applied. For example,

$$\begin{aligned} s \times t + s \times u = \hat{+} \quad &\Rightarrow \quad s \times (t + u) = \hat{+} \\ &\Rightarrow \quad s = \hat{+}/(t + u) \text{ for } 0 \notin (t + u). \end{aligned}$$

This extends the set of variables (or subexpressions) that can be isolated. In our limited experience these properties are sufficient for composing equations in most situations.

The cancellation rule for $\langle \mathcal{S}', \times \rangle$ can be used to move the two sides of an equation to one side (P25):

$$s = t \Leftrightarrow s/t = \hat{+} \text{ for } 0 \notin t$$

This is important for canonicalization, and satisfies the void left by the missing cancellation law for $\langle \mathcal{S}', + \rangle$, highlighted in the last section.[16] That is, equations on $\Re$ are canonicalized by moving all the expressions to one side of the equation using multiplicative or additive cancellation. The cancellation rule for sign multiplication enables a similar process. Coupling this property with the commutativity, associativity and distributivity of $\langle \mathcal{S}', +, \times \rangle$ allows us to represent equations and expressions in a canonical form similar to polynomials on $\Re$.

Earlier we pointed out that $-$ is not the inverse of $\langle \mathcal{S}', + \rangle$ (there is no inverse operator by P10). However $-$ is related to $\langle \mathcal{S}', +, \times \rangle$ roughly in the same way it is related to $\langle \Re, +, \times \rangle$ (P26,P27):

$$\begin{aligned} -s &= \hat{-} \times s \\ s - t &= s + (-t) \end{aligned}$$

As a consequence of this, and $/$ being an inverse (P22), $\langle \mathcal{S}', +, \times \rangle$ and $\langle \Re, +, \times \rangle$ share the following important properties for canonicalization (P28—P31):

$$\begin{aligned} -(-s) &= s & \qquad (s^{-1})^{-1} &= s \text{ for } 0 \notin s \\ -(s+t) &= -s - t & \qquad (s \times t)^{-1} &= (s^{-1}) \times (t^{-1}) \text{ for } 0 \notin s, t \end{aligned}$$

9.3 Properties of S1 not in R1

$\langle \mathcal{S}', +, \times \rangle$ has several 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 10. First, since $\hat{+} \times \hat{+} = \hat{+}$ and $\hat{-} \times \hat{-} = \hat{+}$, an expression is its own multiplicative inverse (P32,P33):

$$\begin{aligned} s \times s &= \hat{+} \text{ for } 0 \notin s \\ s^{-1} &= s \text{ for } 0 \notin s \end{aligned}$$

A major consequence is that $\times$ is its own inverse operator (P34):

$$s/t = s \times t \text{ for } 0 \notin t$$

[16] However, this cancellation law has the less desirable property of introducing an assumption that the denominator does not contain 0.

Thus all occurrences of / in an expression can be replaced with $\times$, if we make explicit the assumption that the denominator doesn't contain 0. Finally, quotient can be replaced by multiplication in the multiplicative cancellation law (P35):

$$s \times t = u \Leftrightarrow s = u \times t \text{ for } 0 \notin t$$

A second set of properties relates to exponentiation, and is a consequence of the first. For a qualitative expression s and integer n, let s^n denote $\underbrace{s \times s \times \cdots \times s}_{n}$ if n is positive, $\hat{+}/s^{|n|}$ if n is negative, and $\hat{+}$ if $n = 0$ (D7). As a consequence of P33 we have that (P36):

$$s^i = s^{|i|} \text{ for } i \in \mathcal{I}$$

And combining it with P32 we get (P37):

$$s^{2i} \times s = s \text{ for } i \in \mathcal{I}$$

Thus all expressions raised to a positive or negative odd power are equivalent; likewise, for positive or negative even powers (i.e., non-zero). One way to view this is that an expression raised to an odd power is equivalent to that expression, while an expression raised to an even power is equivalent to its absolute value (P38), which is equivalently its square:

$$s^{2i} = |s| = s^2 \text{ for } i \in \mathcal{I}, i \neq 0$$

This allows all exponents i to be reduced to 0, 1 or 2. When the exponents are specified as numbers these properties allow all sign expressions to be immediately reduced to quadratics. This is used in section 11 to develop a compact canonical form and the corresponding canonicalization procedures. If the exponents are represented symbolically (e.g., s^{2i+4j}), they can be abstracted to qualitative expressions on values odd, zero and nonzero-even. Splitting these values into odd/even and zero/non-zero gives us the qualitative domains $\mathcal{P}$ and $\mathcal{Z}$, and the corresponding hybrid algebras over the integers, $PI1 = \langle \mathcal{P} \cup \mathcal{I}, \oplus, \otimes, [\,]_{\mathcal{P}} \rangle$ and $ZI1 = \langle \mathcal{Z} \cup \mathcal{I}, \oplus, \otimes, [\,]_{\mathcal{Z}} \rangle$. We analyze and develop symbolic manipulation procedures for these algebras in [31].

A third set of properties consists of novel cancellation rules for sign addition (P39,P40,P41):

$$\begin{aligned} s + s &= s \\ s + \hat{?} &= \hat{?} \\ s \times \hat{?} &= \hat{?} \text{ for } s \neq \hat{0} \end{aligned}$$

The first collapses common summands, while the second two show that $\hat{?}$ acts essentially like a multiplicative zero for both sign + and $\times$.

As a result of the above 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 11 during the discussion of the simplification process.

Also the sign operator tables given earlier for addition and multiplication follow directly from the above properties, the identities and zero, and commutativity. As we saw in the preceding section the remaining operators are definable in terms of these two.

The final set of properties are cancellation laws for $\subset$ and $\approx$. They both have a multiplicative cancellation law, since $=$ has one (P35), and $\approx$ and $\subset$ are strictly weaker than $=$ (P42,P43):[17]

$$\begin{aligned} s \times t \subset u &\Leftrightarrow s \subset u \times t \text{ for } 0 \not\in t \\ s \times t \approx u &\Leftrightarrow s \approx u \times t \text{ for } 0 \not\in t \end{aligned}$$

Unlike $=$ they also have additive cancellation laws (P44,P45):[18]

$$\begin{aligned} s + t \subset u &\Rightarrow s \subset u - t \\ s + t \approx u &\Leftrightarrow s \approx u - t \end{aligned}$$

But while it is a full cancellation law for $\approx$, the cancellation law for $\subset$ runs only in one direction (P46):

$$s \subset u + t \not\Rightarrow s - t \subset u$$

As we discussed earlier for $=$, each of these cancellation laws is central to variable isolation and the canonicalization of equations.

9.4 Relating S1 and R1

Recall that $+$ and $\times$ are closed over $\Re$ and $\mathcal{S}'$, and these subdomains are linked only by $[\,]$. What remains is to characterize $[\,]$ relative to the other operators. As one might expect, it follows from the properties of subset, qualitative abstraction and the qualitative operators (D5, P1) that expressions which delay qualitative abstraction are stronger (P47):

$$[a \text{ op } b] \subset [a] \text{ qop } [b]$$

and since $\subset$ implies $\approx$ (P48):

[17] More precisely, replacing $+$, $-$, $\times$ and $/$ with $\oplus$, $-$, $\times$ and $/$, each of the cancellation laws involving $\subset$ and $\approx$ (but not $=$) hold for any $s, t, u \in \mathcal{S}' \cup \Re$, where at least one of $s, t, u \in \mathcal{S}'$.

[18] The additive cancellation law for $\approx$ is referred to as "the compatibility of addition and qualitative equality" in [9].

$$[a \text{ op } b] \approx [a] \text{ qop } [b]$$

Additionally, [ ] is a homomorphism of $\Re$ onto $\mathcal{S}'$ for multiplication (P49):

$$[a \times b] \;=\; [a] \times [b]$$

That is, multiplying two real values, and then abstracting their sign, produces the same result as abstracting the signs first, and then using sign multiplication — no information is lost by abstracting earlier (i.e., within subexpressions). Since quotient, negation and exponentiation are defined in terms of multiplication, [ ] is a homomorphism for them as well (P50,P51,P52):

$$\begin{aligned} [a/b] &= [a]/[b] \\ [-a] &= -[a] \\ [a^n] &= [a]^n \end{aligned}$$

These homomorphisms play an important role during canonicalization and simplification. Using the homomorphisms, the applications of [ ] are moved as early as possible in an expression, resulting in real operators being converted to sign operators. The unique properties of the sign algebra are then used, resulting in a significant compaction of the expression, not possible with the real algebra.

Notice however that [ ] is not homomorphic for addition or subtraction (P53,P54):

$$\begin{aligned} [a+b] &\neq [a]+[b] \\ [a-b] &\neq [a]-[b] \end{aligned}$$

For example, expressing height difference as $[H_v] - [H_b]$ is strictly weaker than $[H_v - H_b]$ (e.g., consider $H_v = 8, H_b = 7$).

The homomorphisms are used during simplification since they preserve equivalence. As we see later, the properties involving $\subset$ and $\approx$ are used when determining whether one equation/expression subsumes another or are qualitatively equivalent, and when computing the transitivity of inequalities, qualitative composition and hybrid resolution.

9.5 Relation to Interval Algebras and their Properties

While qualitative algebras have been described as interval algebras, at least for traditional interval algebras[15] this is far from reality in terms of motivation, definition, properties and use. Interval algebras were first developed to deal with imprecision, for example, as the result of experimental or round off error, and are normally used to compute tightest bounds. In contrast a qualitative value denotes sets of values that are considered uniform relative to certain properties of interest, and the qualitative abstraction operator is used to throw away uninteresting information — tightest bounds are irrelevant.[19]

In terms of definitions the difference is that a traditional interval arithmetic operator computes the image of the quantitative operator while the qualitative operator finds the least upper bound of values in the qualitative domain containing that image. The domain of the interval algebra normally consists of all bounded subintervals of $\Re$ ($\{[a,b]|a,b \in \Re, a \leq b\}$)[20] and there is normally no abstraction operator, since there is little motivation to throw away precision.

The properties of a bounded interval algebra are analyzed by Moore in [15]. While Moore's analysis does not carry over directly to $SR1$ (since $\hat{+}$, $\hat{-}$ and $\hat{?}$ are unbounded), there are some interesting similarities and differences. For both algebras addition and multiplication are associative and commutative, and $\hat{0}$ is the additive and multiplicative zero. P47 of the preceding section ($[a \text{ op } b] \subset [a] \text{ qop } [b]$) corresponds roughly to what Moore calls *inclusion monotonicity.* Finally neither algebra has an additive inverse, and Moore observes that there isn't an inverse for any intervals with non-zero width.

Turning to differences, even though both are missing an additive inverse, an interval algebra *has* a cancellation law for addition (where X, Y and Z are closed bounded intervals):

$$X + Z = Y + Z \Leftrightarrow X = Y$$

Thus there is a straightforward generalization of Gaussian elimination to interval algebras, even though this is not the case for the sign algebra.

Conversely, while the sign algebra is distributive, an interval algebra is not:

$$X \times (Y + Z) \neq X \times Y + X \times Z$$

For example, $[1,2]\times(1-1) = 0$, while $[1,2]\times 1 - [1,2]\times 1 = [-1,1] \neq 0$ [15].[21] Thus an expression in an interval algebra cannot be reduced to a rational (a polynomial over a polynomial) while preserving equivalence. However, in a sign algebra expressions can be reduced to quadratics while still preserving equivalence.

[19] While tightest bounds are irrelevant here as an end in itself, they have proven useful as a means to inferring these qualitative values [21, 8, 27, 19, 13].

[20] An *extended* interval algebra permits unbounded intervals. The analysis of these algebras is not as extensive as for bounded interval algebras.

[21] The closest, related property is subdistributivity: $X \times (Y + Z) \subset X \times Y + X \times Z$. Also an expression is distributive if X is a real number or if $Y \times Z > 0$.

The multiplicative identity for an interval algebra is 1, while that for the sign algebra is $\hat{+}$. The interval algebra has no multiplicative inverse (i.e., where $y/y = 1$) except when the intervals have zero width. The sign algebra has a multiplicative inverse, which is essential to Minima's equation solving and simplification procedures. Additionally, none of the distinctive properties of the sign algebra highlighted in section 9.3 — replacement of / with $\times$, reduction of exponents to degree 1 or 2, collapsing of common summands and the treatment of $\hat{?}$ as an auxiliary zero — carry over to the interval algebra. Each of these differences is central to how Minima manipulates qualitative symbolic expressions. Thus $SR1$ is not simply an interval algebra.

Finally, in terms ofuse, the numerical community treats interval algebras much like regular algebras. Closed form solutions are constructed using Gaussian elimination, or iterative numeric techniques are used to approximate solutions of differential equations. Simplification is rarely performed, since the lack of distribution prevents reducing expressions to a canonical form. To get around this, interval algebra techniques typically focus on linear equations. In contrast the leverage of qualitative algebras is most substantial when the equations are non-linear. We will see that the extremely small canonical form for $SR1$ equations makes simplification quite desirable. But given that the equations are non-linear, developing general equation solvers becomes quite difficult.

10 Minima

Minima is a symbolic algebra system for manipulating rational expressions on the hybrid algebras $SR1$, $PI1$, $ZI1$ and their subsets, and supports the operations identified earlier. $PI1$ and $ZI1$ are discussed in [31]. Minima is a qualitative analogue of the symbolic algebra system Macsyma[16], and in fact uses Macsyma to manipulate subexpressions in $R1 = \langle \Re, +, \times \rangle$ and $I1 = \langle \mathcal{I}, +, \times \rangle$.

The foundation of Minima consists of the simplification and expression isolation techniques, which support the higher level functions of combining and comparing equations or expressions. By making the simplifier sufficiently powerful (i.e., by reducing expressions to a unique canonical form), identifying tautologies and equivalent equations is reduced to determining syntactic equivalence.[22] As we mentioned earlier, our presentation is concerned only with individual compositions and comparisons, not strategies

[22] Note that Minima provides a unique canonical form only for particular subcases, such as univariate expressions and multivariate monomials. In the general case additional non-syntactic tests are required, as described in the section on canonicalization, section 13.

for performing sequences of manipulations, such as general equation solving over $SR1$.

Like Macsyma, Minima provides two approaches to simplification and isolation, which make different trade-offs between performance and inferential power. The first approach uses the properties analyzed earlier to perform extensive simplifications and isolations, through techniques for qualitative canonicalization and factorization. Canonicalization and factorization are prohibitively expensive in traditional symbolic algebraic systems on $\Re$. However Minima exploits unique properties of the hybrid algebras that make canonicalization and factorization very efficient in practice. Nevertheless, using the first approach there is the potential, in the worst case, for a combinatorial explosion.

Using this insight, the second approach involves partial simplification and partial isolation which perform only "obvious" inferences that make local simplifications to expressions. The potentially expensive operation of distribution is avoided, which can increase the size of expressions exponentially. Completeness is traded for faster more intuitive deductions. This second approach has proven sufficient thus far for the set of examples explored in our design research. Full canonicalization is invoked only when partial simplification is insufficient (e.g., a variable can't be isolated or an equivalence can't be shown).

The next section explores simplification in more detail. The following discusses a general approach to composition — qualitative composition and hybrid resolution — and shows how it captures earlier work on inequality arithmetic and algebra, parts of transition analysis, qualitative resolution, and in [31] the composition of monotone functions (or qualitative proportionalities), The third section discusses the comparison of expressions and equations, including equivalence and subsumption.

11 Simplification and Canonicalization

The purpose of simplification is to eliminate irrelevant structure in the equations. This facilitates comparing and combining equations in two ways. First, it reduces the number of constituents of equations that must be explicitly manipulated, thus speeding up the composition and comparison operations. Second, it restricts the forms that the equations can take, and thus the number of cases that the composition and comparison operators must handle.

The partial simplifier eliminates structure through a set of "obvious" local simplifications that include cancellation (e.g., $a/a \Rightarrow 1$), function application, arithmetic (e.g., $\hat{+} \times \hat{-} \Rightarrow \hat{-}$), substitution of known constants, differentiation, and reduction of subexpressions to a standard, n-ary, lexically ordered form (e.g., $(b \times a) \times c \quad \Rightarrow \quad a \times b \times c$). The properties of $SR1$, described previously in section 9, provide the tools to perform this simplification.

Canonicalization is a superset of partial simplification that pushes the elimination of irrelevant structure to the point where syntactic differences reflect semantic differences in certain important respects. Specifically, two expressions or equations are syntactically different if and only if they are semantically distinct.[23] This for example reduces tests for expression and equation equivalence to determining syntactic equivalence. Partial simplification is a quick operation and is always performed first to eliminate superfluous structure before the more expensive operation of canonicalization is applied.

The following demonstrates the three stages of simplification and canonicalization, and demonstrates the use of $SR1$'s properties in this context. Minima is supplied the expression:

$$([a^3/a] + [c_1]) \ \times \ [-a/(b-4) \times (-c_2)]$$

and values for two constants $c_1 = 5$ and $[c_2] = \hat{+}$, and it automatically starts performing partial simplification. First the real subexpressions are simplified, producing:

$$([a^2] + [5]) \ \times \ [(c_2 \times a)/(b-4)]$$

where the common numerator and denominator, a, in the left subexpression were cancelled, 5 was substituted for c_1 in the middle expression, and in the right subexpression the double negation was cancelled, and c_2 was brought inside the quotient. In the second stage the homomorphisms (P49 – P52) are applied and real numbers are abstracted to signs (by D5), producing:

$$([a]^2 + \hat{+}) \times (([c_2] \times [a])/[b-4])$$

where [5] on the left has been replaced with $\hat{+}$, and / and $\times$ on the right have been brought outside the abstraction operator.

Mapping from the reals to signs 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:

$$[P] = [d \times g \times H] \Rightarrow [d] \times [g] \times [H] \Rightarrow [P] = [H]$$

Second, the properties of section 9.3 allow significant simplifications in S' not possible in $\Re$. As a result Minima often eliminates a substantial amount of an expression's superfluous structure.

In the last stage the sign expressions are partially simplified. In our example:

[23]Two expressions are semantically equivalent if they denote the same function, and two equations if they denote the same relation.

$$([a]^2 + \hat{+}) \times ((\hat{+} \times [a]) \times [b-4]) \text{ for } 0 \notin [b-4]$$

is produced by substituting $\hat{+}$ for c_2, by replacing / with $\times$ (by P34), and by making explicit the condition that the denominator doesn't contain 0.

Next the multiplicative identity, $\hat{+}$, is cancelled (by P9), and parentheses are eliminated according to associativity (P15):

$$([a]^2 + \hat{+}) \times [a] \times [b-4] \text{ for } 0 \notin [b-4]$$

all local simplifications have been performed, thus partial simplification terminates.

Next Minima is asked to canonicalize this result. Note that typically, as a result of partial simplification, the real subexpressions are quite small, and thus can be canonicalized quickly. In this example the real subexpressions are already canonical; thus, the first two stages of canonicalization have no effect. Walking through the final stage — canonicalizing sign expressions — the first multiplicand is distributed (by P16):

$$([a]^2 \times [a] \times [b-4]) + (\hat{+} \times [a] \times [b-4]) \text{ for } 0 \notin [b-4]$$

The property $s^2 \times s = s$ (P37) is used to reduce the cubic term, $([a]^2 \times [a])$, to degree 1, and the multiplicative identity, $\hat{+}$, is cancelled (by P9):

$$([a] \times [b-4]) + ([a] \times [b-4]) \text{ for } 0 \notin [b-4]$$

Finally the property $s + s = s$ (P39) is used to collapse the common summands:

$$([a] \times [b-4]) \text{ for } 0 \notin [b-4]$$

Next we consider the specific properties of the two simplification approaches, and their differences, starting with canonicalization.

11.1 Canonicalization and Pseudo-Canonicalization

The first approach reduces an expression or equation to a form analogous to a multivariate (quadratic) polynomial. For certain subcases, such as univariates and monomials, the form is canonical. For the remaining cases we refer to the form as pseudo-canonical, due to its similarity to the canonical form for $R1$. We drop the prefix "pseudo" for most of this discussion, but indicate when a form is and isn't unique.

The canonical form for $SR1$ expressions is developed in terms of the forms for real and sign subexpressions, and their composition. In traditional symbolic algebra systems real expressions are canonicalized by reducing them to one of two unique forms. First, expressions can be reduced to a unique rational form — a fraction consisting of two polynomials with common factors removed. Second, the numerator and denominator of the fraction can be represented as a prime factorization,[24] rather than two polynomials.

We select the second form — prime factorization — on the following basis. The canonicalization process should use the homomorphisms to map as many real operators to sign operators as possible. This allows the canonicalizer to use the unique properties of the sign algebra to further reduce the expression. Recall that [ ] is homomorphic for multiplication, division, exponentiation and negation (P49 – P52), but not for addition or subtraction (P53,P54). Furthermore, to apply the homomorphism for multiplication through negation, they must appear at the top level of a real expression contained within [ ]. For example, the homomorphism can be applied to $[a \times (b+c)]$, but not to $[a+(b \times c)]$. A prime factorization moves $\times$, $/$, negation and exponentiation to the top level, while polynomial form moves $\times$, $/$ and exponentiation to the inner-most expression; thus, prime factorization is preferred for the real subexpressions.

Next consider sign expressions. The unique properties of $\langle S', +, \times \rangle$ allow all sign expressions to be reduced to *quadratic* polynomials (P55):

$$c_0 \ + \ (c_1 \times s) \ + \ (c_2 \times s^2) \ \text{ where } c_i \in S'$$

by replacing $/$ with $\times$ (P34), by constructing a polynomial using associativity, commutativity, and distributivity (P12 – P16), and by reducing each exponent to degree 1 or 2 (P37,P38). This is a significant compaction over that possible with real expressions.

Quadratics are created and manipulated in a manner analogous to the polynomial arithmetic of Macsyma[16] and other symbolic algebras systems. To perform quadratic arithmetic on $S1$, negation, subtraction, exponentiation and quotient are reduced to addition and multiplication, according to P26, P27, D7 and P34. Addition and multiplication are then performed according to the following proposition:

Proposition 56 (Quadratic Arithmetic) *Let*

$$\begin{aligned} A(s) &= a_0 \ + \ a_1 \times s \ + \ a_2 \times s^2 \\ B(s) &= b_0 \ + \ b_1 \times s \ + \ b_2 \times s^2 \end{aligned}$$

where s is a variable over S', and each a_i and b_i is an expression over S' not involving s, then,

$$\begin{aligned} A(s)+B(s) &= (a_0+b_0) \ + \ (a_1+b_1) \times s \ + \ (a_2+b_2) \times s^2 \\ A(s) \times B(s) &= (a_0 \times b_0) + \end{aligned}$$

[24] A factorization is prime if none of the factors can be factored.

$$(a_0 \times b_1 + a_1 \times b_0 + a_1 \times b_2 + a_2 \times b_1) \times s +$$
$$(a_0 \times b_2 + a_1 \times b_1 + a_2 \times b_0 + a_2 \times b_2) \times s^2.$$

An expression is converted to a quadratic by evaluating it using the quadratic arithmetic operators. In the univariate case the coefficients are constants in $\mathcal{S}'$, and are added and multiplied according to sign arithmetic. In the multivariate case the variables of the expression $s_1, \ldots, s_n$ are ordered ($\prec$), for example lexicographically, such that $s_i \prec s_{i+1}$. Coefficients of a quadratic in s_n are quadratics in $s_1, \ldots, s_{n-1}$. To add or multiply two quadratics their coefficients are recursively added and multiplied using the quadratic operators (terminating in sign addition and multiplication).[25]

All expressions reduce to quadratics by this process (P57). This reduction, however, does not result in a canonical form, — several quadratics denote the same function; for example, by P40 $s + \hat{?} = \hat{?}$. Additionally, the following set of equivalences hold between sign quadratics in v (P58 – P63):

$$\begin{array}{rcl}
s + (s \times v^2) & = & s \text{ for } v \in \mathcal{S} \\
s + (s \times v) + (s \times v^2) & = & s + (s \times v) \\
-s + (s \times v) + (s \times v^2) & = & -s + (s \times v^2) \\
s - (s \times v) + (s \times v^2) & = & s - (s \times v) \\
-s - (s \times v) + (s \times v^2) & = & -s + (s \times v^2)
\end{array}$$

where s and v are expressions in $\mathcal{S}1$. The final stage of canonicalization for sign expressions eliminates irrelevant terms in a quadratic using these equivalences as simplification rules (where $=$ becomes $\Rightarrow$). For example, $\hat{+} + u + u^2 \ \Rightarrow \ \hat{+} + u$ using P59. Note that the first equivalence is used only when v ranges over $\mathcal{S}$, (as opposed to $\mathcal{S}'$), for example, when $v = [a]$.

In the univariate case the set of equivalences is complete (P64); thus, the final stage reduces a univariate expression to a canonical form. In this case the coefficients of a quadratic are elements of $\mathcal{S}'$, and $\mathcal{S}'$ contains 4 distinct values; thus, there are 64 distinct univariate quadratics. After canonicalization 14 semantically distinct expressions can be produced.[26]

Multivariate quadratics are not always guaranteed to be canonic, even after applying the above rewrites. In this case, where the expressions are

[25] The following additional simplifications are performed a) the multiplicative and additive identities ($\hat{+}$ and $\hat{0}$) are cancelled, b) terms of the form $\hat{0} \times s^i$ are removed from a quadratic, and c) a quadratic of the one term $a \times s^0$ reduces to a.

[26] This seems quite small, considering that there are $4^4 = 256$ possible functions from $\mathcal{S}'$ to $\mathcal{S}'$. Of course what happens to $\hat{?}$ isn't particularly interesting. There are 64 functions from $\mathcal{S}$ to $\mathcal{S}'$, and only 9 from $\mathcal{S}$ to $\mathcal{S}$.

pseudo-canonical, determining equivalence involves more than a syntactic comparison, as is discussed in section 13.

Multivariate quadratics are canonic when restricted to expressions in $S2$ (P65). Recall that in $S2$ binary addition and subtraction are disallowed, but multiplication, negation, quotient, and exponentiation are permitted. In this case the quadratics are multivariate monomials. This is an important subcase, since all the design examples we have considered can be expressed in $SR2$ — $S2$ combined with $R1$, or equivalently $SR1$ without sign addition and subtraction. Likewise a qualitative representation can be described by a set of nonlinear inequalities, which are all expressible in $SR2$. Multivariate quadratics are also canonic for $S1$ if we disallow expressions that can take on the value of $\hat{?}$ (P66). That is, there must be no opportunity for ambiguity to arise.

To canonicalize $SR1$ expressions we combine the procedures, discussed above, for canonicalizing $R1$ and $S1$ expressions. An $SR1$ expression can be viewed as an expression in $S1$ whose "variables" are the signs of expressions in $R1$. Roughly speaking we canonicalize an $SR1$ expression by first computing the prime factorizations of the real subexpressions, next taking their signs, and finally reducing the sign expression to a quadratic, while treating the signs of real subexpressions as distinct variables.

Elevating the real subexpressions to signs in part involves applying the homomorphisms, and reducing the exponents to degree 1 or 2. Thus, an expression, $E(x)$, reduces as follows (P67):

$$[E(x)] = [\prod_i f(x)_i^{n_i}] \Rightarrow \prod_i [f(x)_i]^{p(n_i)}$$

where $f(x)_i$ are the prime factors of $E(x)$, and $p(n_i) = 1$ if n_i is odd, and $p(n_i) = 2$ if n_i is even.

A caveat is that, although a prime factor, $f(x)$, is canonical over the reals, it is not if we are only interested in its sign, $[f(x)]$. For example, the expression $a^2 + 2$, is prime, but never crosses zero; thus $[a^2 + 2] = \hat{+}$.

More generally we know from standard algebra that the prime factorization of an expression $E(x)$ over the reals consists of a set of linear and quadratic terms. That is, for each expression $E(x)$ in $\langle \Re, +, \times \rangle$ there exists a C, $L(x)$ and $Q(x)$ such that:

$$E(x) = C \times L(x) \times Q(x)$$

where C is a constant, and:

$$\begin{aligned} L(x) &= \prod_i (x - d_i)^{n_i} \\ Q(x) &= \prod_j (a_j \times x^2 + b_j \times x + c_j)^{m_j} \end{aligned}$$

where a_j, b_j, c_j and d_i are constants, n_i and m_j are non-zero integers, and the terms of $L(x)$ and $Q(x)$ are prime for the reals (that is each quadratic in $Q(x)$ has complex roots).[27]

The key feature is that $Q(x)$ is non-zero for all $x \in \Re$. Since $Q(x)$ is continuous, it will always have the same sign, and thus can be combined with C, leaving us a constant, C', times a set of linear terms (P68):

$$\begin{aligned} [E(x)] &= [C'] \times \prod_i [x - d_i]^{p(n_i)} \text{ for } x \neq d_i \text{ when } n_i < 0 \\ C' &= C \times Q(0) = C \times \prod_j c_j^{m_j} \end{aligned}$$

This is a canonical form for $[E(x)]$. Also recall that $SR2$ combines $\langle \Re, +, \times \rangle$, $\langle S', \times \rangle$ and $[\,]$. Since $[\,]$ is homomorphic to multiplication, $SR2$ is equivalent to $\langle \Re, +, \times \rangle$ plus $[\,]$ (P69). Thus, P68 also provides a canonical form for univariate expressions in $SR2$.

Consider the physical interpretation of this canonical form. Intuitively, $[C']$ is the sign of $E(x)$ for sufficiently large values of x. $E(x)$ hits zero for each d_i where x is defined. $E(x)$ approaches positive or negative infinity when x is not defined for d_i (i.e., when $n_i < 0$ and thus d_i is a pole). If $p(n_i) = 1$ then $E(x)$ crosses between positive and negative at d_i, either through zero or infinity. If $p(n_i) = 2$ then it touches zero or approaches infinity at d_i, but doesn't cross between positive and negative. From the starting sign, the zeros and the poles we can determine the sign of $E(x)$ between adjacent d_i. $[dE(x)/dt]$ and $[d^2E(x)/dt^2]$ describe the critical points and convexity of $E(x)$ in a similar manner. These are most of the qualitative features represented for example within QMR[18] for the case of rational expressions. Thus the canonical form for $[E(x)]$ is an explicit and succinct representation of $E(x)$'s qualitative features (figure 4).

Thus far we have discussed the canonicalization of hybrid expressions. To canonicalize a hybrid *equation*, Minima first uses the cancellation law for multiplication (P24)[28] to bring both expressions to one side of the equation (P25):[29]

$$s = t \Leftrightarrow s \times t = \hat{+} \;\text{ for } 0 \notin t$$

[27] Procedures exist for computing C, $L(x)$ and $Q(x)$ for the univariate case, but not for arbitrary multivariates.

[28] For real equations the canonical form is a rational equated to 0; one side of the equation is brought over to the other side using the cancellation law for addition. Sign expressions do not have an additive cancellation law; thus, the law for multiplication must be used. This has the unfortunate consequence of introducing an assumption about the denominator being non-zero. Thus we make explicit this assumption, along with the original equation simplified for the cases where the assumption is false.

[29] Two sides of an equation are not combined if one expression simplifies to 0.

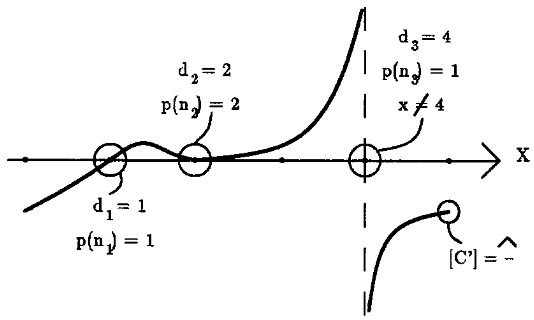

Figure 4: Example of the qualitative features of an expression $E(x)$ highlighted by the canonicalization of $[E(x)]$, where $[E(x)] = \hat{-} \times [x-1] \times [x-2]^2 \times [x-4]$ for $x \neq 4$.

and then canonicalizes the combined expression.[30] Cancellation is not performed when one side of the equation is zero. Canonicalization is analogous for $\subset$. For $\approx$ the additive rather than the multiplicative cancellation laws are used since they do not introduce the assumption $0 \notin t$.

11.2 Performance of (Pseudo) Canonicalization

We evaluate the performance of canonicalization in terms of the three main stages. The major cost when canonicalizing real subexpressions is factoring, which relies heavily on computing greatest common denominator (GCD). This cost is incurred for both rational form and prime factorization. Rational form requires determining and then canceling common factors between the numerator and denominator. In the worst case reduction to polynomial form can produce an exponential number of summands. The complexity is exponential both in the size of the expression and the number of variables.

In practice AI circuit analysis systems using Macsyma have found this cost unacceptable for practical problems such as power supply analysis and circuit parameter selection[22, 7]. The symptom is that during analysis the real equations start small, but grow in size quickly as equations are combined. This growth is reflected in a substantial slow down due to factorization.

$S\Re 1$ is a superset of the real algebra, and thus inherits its worst case complexity. However, in practice canonicalization of $S\Re 1$ equations is much faster than that for real equations, and thus far appears acceptable for our

[30]The partial simplifier is run on the equation before and after canonicalization.

design approach — interaction-based design [30]. First, real subexpressions within $SR1$ equations start out smaller before simplification than corresponding real equations, since some real operators have been replaced with sign operators. Second, while real equations explode in size as they are combined, as $SR1$ equations are combined *the real expressions tend to stay small.* The reason is that most of the operators in the real subexpressions are mapped to sign operators using the homomorphisms. The only operators not mapped are addition and subtraction; thus, after canonicalization the remaining real subexpressions are the prime factors, which are quite simple — linear or quadratic.

The second stage of canonicalization — applying the homomorphism — is a linear time operation. The third stage — canonicalizing sign expressions — is much less costly in practice than that for real expressions. First, sign expressions reduce to polynomial rather than rational form; thus, the costly operations of factoring and GCD are unnecessary. Furthermore, all polynomials reduce to quadratics, thus expressions tend to stay smaller. While reduction to rational form is exponential both in the initial expression size and the number of distinct variables, reduction to quadratic form is exponential only in the variables, and linear in the expression size. The overall result is that the homomorphisms reduce the size of real expressions by mapping real operators to sign operators, and then this additional structure is collapsed because of the compact form for sign expressions.

11.3 Partial Simplification

The partial simplification approach avoids performing simplifications that radically change the structure of an equation, or that can cause it to grow exponentially. One motivation, of course, is computational complexity. The second is that radical changes to structure make equations difficult for engineers to interpret. The culprit is distributivity, which can produce an exponential growth in expression size during canonicalization. For example, using distributivity the expression $(a+b)(c+d)(e+f)$ reduces to $ace + bce + ade + bde + acf + bcf + adf + bdf$. By eliminating distributivity for $\Re$ and S', but performing all local simplifications, the complexity is reduced to O(n log(n)) in the expression size — the cost of recursively sorting operands. Simplification of expressions and equations roughly involves using the properties of section 9 (except distribution), and analogous properties for real expressions and equations as rewrite rules. Expression simplification includes arithmetic and combining common denominators. Equation simplification includes the use of cancellation, but avoids combining subexpressions on the right and left sides of the equality. The precise set of simplification rules is given in [31]. An example of partial simplification has already been presented.

12 Composition

In our design verification example the interactions being composed were described as equations involving equivalences, and composition was performed based on substitution of equals on a shared variable (or expression). This section presents a general approach to composition where the interactions are described by any $SR1$ equations, including $\subset$ and $\approx$, as well as $=$.

The discussion begins with substitution of equals (section 12.1). We then expand to *qualitative composition* on subset and qualitative equality (section 12.2), which allows arbitrary composition involving shared sign variables and subexpressions. We conclude with *hybrid resolution*, which allows arbitrary composition involving shared real variables and subexpressions (section 12.3). With respect to developing a unified theory of qualitative algebraic reasoning we show how these composition rules capture qualitative resolution, inequality arithmetic and algebra, parts of transition analysis, and in appendix A the composition of monotonicity functions.

12.1 Composition by Substitution of Equals: Isolation and Factorization

Equations are composed by isolating a shared variable or subexpression, substituting for it, and simplifying the result. Simplification was presented earlier. Variable isolation and substitution depend on the type of equation involved: $=$, $\subset$ or $\approx$. This section concentrates on variable isolation for $=$, expanding to isolation and substitution for $\subset$ and $\approx$ in section 12.2. The $S1$ factorization technique which is the primary focus of this section applies equally to all forms of isolation.

Given an equation in $\langle \Re, +, \times \rangle$ variables are isolated using standard techniques. However, this is not the case for $\langle S', +, \times \rangle$. Since there is no cancellation law for addition, addends cannot be moved between the left and right sides of an equation (P18):

$$s + t = u \quad \not\Leftrightarrow \quad s = u - t$$

However, it is often possible to isolate certain variables and subexpressions. Cancellation can be performed for multiplication (section 9.2); thus, we can isolate any variable or subexpression that is an argument of a top-level multiplication (or division) by moving the remaining arguments to the opposite side of the equation (P24):

$$s \times t = u \quad \Leftrightarrow \quad s = u \times t \text{ for } 0 \notin t$$

This is sufficient to isolate any variable in an equation on $S2 = \langle S', \times \rangle$, by rewriting the equation as (P70):

$$v^i = e$$

where $v, e \in \mathcal{S}'$, v is the variable being isolated, e is a product not containing v, and i is of degree 1 or 2. v is real-valued only when i is 1 or e is positive. If i is 2 and e positive we further reduce the equation as follows (P71):

$$v^2 = e,\ \hat{-} \not\subset e \Rightarrow v = \pm e$$

For equations in $SR2$ this approach is used to isolate any of the (real-valued) prime factors. This is the most common case in our experience.

A problem occurs when we move to $S1$. For example, in the equation $(s+t) \times u = v$ we can isolate $s+t$:

$$(s+t) \times u = v \quad \Rightarrow \quad (s+t) = v \times u \text{ for } 0 \notin u$$

as well as u and v; however, we cannot isolate s or t. To identify "obvious" isolations partial simplification is performed on an equation, and then any of the multiplicands of the top-level expressions may be isolated.

A second, more powerful approach maximizes the resolution of the subexpressions that can be isolated. This in turn increases the set of $S1$ equations that can be composed, but at a computational cost. The approach involves factoring the expression of a canonicalized equation, and then isolating 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 exploit the fact that canonical sign expressions are at most quadratic to produce a much simpler approach. Consider factorization for the different cases. In the following unless otherwise indicated k, k_0, k_1, k_2 are non-zero sign expressions not containing s, and a, b, c, and d are sign expressions not containing s. A monomial:

$$M(s) = k \times s^i$$

is factored by factoring the coefficient k (P72). Given a linear expression:

$$L(s) = k_0 \ + \ k_1 \times s$$

its factorization $L'(s)$ is of the form (P73):

$$L(s)' = a \times (b + c \times s)$$

where $k_0 = a \times b$, $k_1 = a \times c$, and a contains the common factors of k_0 and k_1. Finally given a quadratic:

$$Q(s) = k_0 \ + \ k_1 \times s \ + \ k_2 \times s^2$$

where one of k_1 or k_0 may be zero, its factorization is of the form (P74):

$$Q'(s) = (a \times s \ + \ c) \ \times \ (b \times s \ + \ d)$$

where a, b, and c are factored, and:

$$\begin{aligned} k_2 &= a \times b \\ k_1 &= (a \times c) + (b \times d) \text{ and} \\ k_0 &= c \times d. \end{aligned}$$

Suppose k_0 is $\hat{0}$, then the factorization reduces to (P75):

$$Q'(s) = a \times s \times (b \times s \ + \ d)$$

where, analogous to the linear case, a is the common factors of k_1 and k_2. Suppose k_1 is $\hat{0}$, then $Q(s)$ cannot be factored (P76), since:

$$\begin{aligned} &k_1 = (a \times c) + (b \times d) = \hat{0} \\ &\quad \Rightarrow \quad a = \hat{0} \lor c = \hat{0}, \ b = \hat{0} \lor d = \hat{0} \end{aligned}$$

but:

$$\begin{aligned} k_2 &= a \times b \neq \hat{0} \Rightarrow a \neq \hat{0}, b \neq \hat{0}, \text{ and} \\ k_0 &= c \times d \neq \hat{0} \Rightarrow c \neq \hat{0}, d \neq \hat{0}. \end{aligned}$$

Finally, suppose all the coefficients are non-zero. In this rare case it is relatively inexpensive to determine the factors by generate and test — guessing values for a, b, c and d and testing the guess with the equations relating them to k_0, k_1 and k_2. First, we recursively factor the coefficients k_2 and k_0. Each coefficient is itself at most quadratic, and thus will have at most two factors each. Next we guess the values for a, b, c and d, by distributing k_2's factors between a and b, and k_0's factors between c and d. Exploiting symmetry either a and b each get one of k_2's factors, or a gets both factors and $b = \hat{+}$. k_0's factors are distributed to c and d in a similar manner. The result is four sets of assignments. Each assignment is checked by testing if k_1 and $(a \times c) + (b \times d)$ are equivalent.

In each case discussed — monomial, linear or quadratic — factorization is linear in the size of the canonical expression. Composition based on substitution of equals is straightforward given Minima's simplifier and the variable isolation technique just discussed. Appendix A demonstrates how partial simplification plus composition of equivalence subsumes Kuiper's composition of monotonicity functions[11]. We show how the algebraic relations within qualitative process theory[10] map to $SR1$ in [31].

12.2 Qualitative Composition of $=$, $\subset$ and $\approx$

Thus far we have focused on composing equations involving equality by using substitution of equals (P77):

$$E(s),\ s = t \Rightarrow E[s|t]$$

where $E(s)$ is an equation in s, and $E[s|t]$ indicates that one or more occurrences of s in $E(s)$ has been replaced with t. This is sufficient for equations in $SR2$ (i.e., $SR1$ without sign addition), where abstraction preserves equivalence (P78):

$$a = b \times c \Rightarrow [a] = [b] \times [c]$$

But in $SR1$ abstraction introduces $\subset$ and $\approx$(P79,P80):

$$\begin{aligned} a = b + c &\ \Rightarrow\ [a] \subset [b] + [c] \\ a + b = c + d &\ \Rightarrow\ [a] + [b] \approx [c] + [d] \end{aligned}$$

Thus composition must generalize to equations involving $=$, $\subset$ and $\approx$; we refer to this as *qualitative composition.* To generalize we use the fact that $\subset$ is transitive, although asymmetric, to substitute supersets.[31] $\approx$ is intransitive, and thus does not have a corresponding substitution rule; however, as we demonstrate below, $\approx$ can often be reduced to $\subset$, thus allowing substitution.

To develop qualitative composition we first observe that $E(s)$ in the substitution of equals rule (P77) is not restricted to $=$, but may involve $\subset$ or $\approx$. The rule only requires an equivalence to appear in the equation used during isolation. The composition process is analogous in the remaining cases where isolation is performed on equations involving $\subset$ or $\approx$. However, there are important differences. One is that for $\subset$ one side of the relation may denote a sign, while the other side denotes a real, for example,[32] $a \subset [d]$. Thus composition may substitute a sign for a real, or vice-versa. For example, composing the above equation with $b = a + c$, Minima substitutes $[d]$ for a producing:

$$b \subset [d] \oplus c\ .$$

Note that while $+$ has been replaced with $\oplus$, this is only a notational change. Although we are able to blur the distinction between real and qualitative addition when the operands denote all signs or all reals, they are not equivalent when the operands are mixed (P2). If we know further that $d \neq 0$, then exploiting the reduction (P3–P5):[33]

[31] Substitution of superset, as well as equals, is sometimes used by traditional interval algebra techniques [15].

[32] During this process we exploit the equivalence $a \subset S \Leftrightarrow [a] \subset S$ (P7).

[33] We can also have Minima elevate d to $[d]$ without the condition $d \neq 0$, but at the cost of losing information.

$$a \oplus s = [a] + s \text{ for } s \neq 0$$

simplification elevates c to its sign $[c]$, allowing $\oplus$ to be denoted once again as sign addition $+$:

$$b \subset [d] + [c]$$

Another difference occurs during isolation; recall that, unlike $=$, $\subset$ and $\approx$ have both multiplicative and additive cancellation laws (P42 – P45). Although the additive law for $\subset$ only allows addends to be moved from the subset (left) to the superset expression (right) (P46):

$$s \subset u + t \not\Rightarrow s - t \subset u$$

To isolate a variable or expression in $\approx$ the relation is first canonicalized by factoring the real subexpressions and reducing the sign expressions to quadratics (the sign expression is not factored). Since $\approx$ has the full complement of cancellations a variable is isolated the same way as for equations on the reals.

Because of the asymmetry in the additive cancellation law for $\subset$, a variable or subexpression on the left side of $\subset$ is isolated using the approach for $\approx$, while on the right side it is isolated using the approach for $=$.

If a variable or subexpression cannot be isolated in an expression involving $=$ or $\subset$ the relation can be weakened to one that has the desired cancellation (P81,P82):

$$\begin{aligned} s + t = u \quad &\Rightarrow \quad s \subset u - t \\ s \subset t + u \quad &\Rightarrow \quad s - u \approx t \end{aligned}$$

After a variable has been isolated it may be possible to "strengthen" the relation as follows (P83,P84,P85):

$$\begin{aligned} [x] \approx s \quad &\Leftrightarrow \quad [x] \subset s \\ s \subset [x] \quad &\Leftrightarrow \quad s = [x] \\ \hat{?} \subset s \quad &\Leftrightarrow \quad \hat{?} = s \end{aligned}$$

Finally, substitution is performed using the following proposition:

Proposition 86 (Qualitative Composition Rules) *Let $s, t, u \in S' \cup \Re$, let equation $E(s)$, and expressions $e(s)$ and $e(t)$ be in $SR1$, and let at least one of s, t and u be in S' for all but the first rule, then*

$$\begin{aligned} E(s), s = t \quad &\Rightarrow \quad E[s|t] & s \subset t, e(s) \subset u \quad &\Rightarrow \quad e[s|t] \approx u \\ s = e(t), t \subset u \quad &\Rightarrow \quad s \subset e[t|u] & s \approx e(t), t \subset u \quad &\Rightarrow \quad s \approx e[t|u] \\ s \subset e(t), t \subset u \quad &\Rightarrow \quad s \subset e[t|u] & s \approx t, e(t) \subset u \quad &\Rightarrow \quad e[t|s] \approx u \\ s \subset t, e(t) \subset u \quad &\Rightarrow \quad e[t|s] \subset u. & & \end{aligned}$$

What is absent from these rules is substitution of $\approx$, as $\approx$ is intransitive. The composition rules apply to any qualitative or hybrid algebra, since they depend only on the set properties of $=$, $\subset$ and $\approx$, and the inclusion monotonicity[15] of the qualitative operators (P1). What depends on $SR1$'s particular properties are the cancellation laws used during isolation (P35,P42–P45) and the reduction of $\approx$ and $\subset$ to stronger relations ($\subset$ and $=$, respectively)(P83–P85). Dormoy and Raiman's qualitative resolution rule for composing confluences[9]:

$$s + [a] \approx u,\ t - [a] \approx v \Rightarrow s + t \approx u + v.$$

follows as an instance of qualitative composition, on equations restricted to $\approx$ and sign $+$. That is, we compose the two left hand equations by isolating $[a]$ in one of them (using P45), simplify $\approx$ to $\subset$ using P84, and substitute the superset using P86. The result is equivalent to that of qualitative resolution (see [31] for further details).

12.3 Hybrid Resolution, Inequality Reasoning and Transition Analysis

The most common qualitative reasoning inferences not already captured involve inequality reasoning and its coupling to transition analysis. If we express inequalities as signs of differences (e.g., $a > b \Rightarrow [a - b] = \hat{+}$) and introduce the following schema (called the *inequality composition law*):

$$[q_1 - q_3] \subset [q_1 - q_2] + [q_2 - q_3]$$

then inequality reasoning falls out of substitution of supersets. For example, we infer $a > b, b > c \Rightarrow a > c$ by instantiating the law to get $[a - c] \subset [a - b] + [b - c]$, substituting supersets using $[a - b] \subset \hat{+}$ and $[b - c] \subset \hat{+}$, and simplifying to get $[a - c] \subset \hat{+} + \hat{+} = \hat{+}$. Furthermore, we can couple to transition analysis using the qualitative variant of the mean-value theorem (similar to [6]):

$$[q(t + \epsilon)] \subset [q(t)] + [dq(t + \epsilon)/dt] \text{ for sufficiently small } \epsilon$$

While these equations are traditionally used only while performing arithmetic, Minima can also use them during symbolic algebra.

The inequality composition law, is reminiscent of qualitative (or traditional) resolution. The two expressions on the right side of the equation are being combined by resolving q_2 and its negation. The difference from qualitative resolution is that the resolvants are real subexpressions within a hybrid expression. By generalizing the law it can be used, in conjunction with qualitative composition, to "resolve" arbitrary variables and subexpressions in SR1 hybrid equations:

Proposition 87 (SR1 Hybrid Resolution Law) *Given $a, b, q_1, q_2, q_3 \in \Re$, then*

$$[(b \times q_1) - (a \times q_3)] \subset [q_1 - (a \times q_2)] + [(b \times q_2) - q_3]$$

Formulating hybrid resolution as an equation schema instead of a rule of inference allows us to exploit the existing techniques for factoring, variable isolation and qualitative composition.

To perform hybrid resolution, first two equations are identified that share a common real variable (or subexpression). For example the following two equations share x:

$$\begin{aligned} [(x + w) \times z + w] + s &\subset t \\ [-x + y] \times t &= u \end{aligned}$$

Second, the sign expression ($[e(v)]$) containing the variable of interest (v) in each equation is isolated, using the simplification, factoring and isolation techniques for $S1$ already presented:

$$\begin{aligned} [(x + w) \times z + w] &\subset t - s \\ [-x + y] &= u \times t \end{aligned}$$

Third, the two $e(v)$'s are rewritten in the form $a \times v - b$ and $-a \times v + b$,[34] by recursively applying the following rewrites:

$$\begin{aligned} v &\Rightarrow 1 \times v + 0 \text{ applied only initially} \\ (a \times v + b) + c &\Rightarrow a \times v + (b + c) \\ (a \times v + b) \times c &\Rightarrow (c \times a) \times v + (c \times b) \\ -(a \times v + b) &\Rightarrow (-a) \times v + (-b) \\ (a \times v + b)/c &\Rightarrow (a/c) \times v + (b/c) \end{aligned}$$

For our example:

$$\begin{aligned} [z \times x + (z \times w + w)] &\subset t - s \\ [-x + y] &= u \times t \end{aligned}$$

Finally, the right side of these equations are substituted into the hybrid resolution law by applying the appropriate rule of qualitative composition for each of the two equations. For our example the result before simplification is:

$$[1 \times (z \times w + w) - z \times (-y)] \subset (u \times t) + (t - s)$$

[34] where v is the variable or expression being isolated, and a, b, c and d may include other instances of v.

13 Comparison

Desired predicates for comparing expressions are equivalence, subsumption and qualitative equivalence, which test $=$, $\subset$ and $\approx$, respectively. Predicates on single equations are tautology and inconsistency, and for comparing equations, logical equivalence ($\Leftrightarrow$) and subsumption ($\Leftarrow$). Before evaluating any predicate the expressions and equations are (pseudo) canonicalized to limit the cases to be considered.

Starting with $=$, recall that two canonical expressions are equivalent, that is they denote the same function, exactly when they are syntactically equivalent. Also, given an ordering on variables, multivariate expressions on $R1$ canonicalize as rational expressions or factors, and $S2$ expressions canonicalize as monomials (P65). Univariate expressions on $SR2$ are canonical, by computing a prime factorization and removing the quadratic factors (since they never cross zero) (P68). Multivariate quadratics on $S1$ are canonical if they cannot take on the values $\hat{?}$ (P66), and univariate quadratics are canonical after performing an additional set of reductions (P64). These reductions, however, do not reduce an $S1$ expression to a canonical form in the multivariate case. For example the following two reduced expressions are equivalent (P88):

$$u \times v + \hat{?} \times u = -u \times v + \hat{?} \times u$$

The following proposition provides a (somewhat unwieldy) condition for two $S1$ quadratics being equivalent in the general case. Note that $\overline{v}_i$ denotes a vector of variables $v_1, \ldots, v_i$:

Proposition 89 (S1 Equivalence Condition) *Let S, T be quadratics in $\overline{v}_n$, with quadratic coefficients in $\overline{v}_{n-1}$. Given*

$$\begin{aligned}
S &= s_2 \times v_n^2 + s_1 \times v_n + s_0,\\
T &= t_2 \times v_n^2 + t_1 \times v_n + t_0, \ \textit{then}\\
\forall \overline{v}_n . S = T \quad \Leftrightarrow \quad & s_0 = t_0,\\
& (s_2 \neq t_2 \vee s_1 \neq t_1) \Rightarrow\\
& (s_0 = \hat{?}, \vee\\
& ((s_1 \vee s_2 = \hat{?}, \vee (s_2 = -s_0, (s_0 = \hat{+} \vee s_0 = \hat{-}))),\\
& (t_1 \vee t_2 = \hat{?}, \vee (t_2 = -s_0, (s_0 = \hat{+} \vee s_0 = \hat{-})))), \vee\\
& ((s_0 = \hat{+} \vee s_0 = \hat{-}), s_1 = t_1, (s_1 = \hat{+} \vee s_1 = \hat{-}),\\
& (s_2 = 0 \vee s_2 = s_0), (t_2 = 0 \vee t_2 = t_0))).
\end{aligned}$$

Analogous conditions for subsumption ($\supset$) and qualitative equivalence ($\approx$), although straightforward to derive from this one, are equally opaque. However, the subcases where at least one coefficient is zero are much simpler; for example:

Proposition 90 (S1 Equivalence of Linear Polynomials) *Let S, T be linear in $\overline{v}_n$, with quadratic coefficients in $\overline{v}_{n-1}$, where*

$$\begin{aligned} S &= s_1 \times v_n + s_0, \\ T &= t_1 \times v_n + t_0, \ \textit{then} \\ \forall \overline{v}_n . S &= T \Leftrightarrow (s_0 = t_0, s_1 \neq t_1 \Rightarrow s_0 = \hat{?}). \end{aligned}$$

The complete set of subcases for $=$, $\subset$ and $\approx$ are given in [31]. Each of these conditions is used to test pairs of quadratics by recursively testing their coefficients. A subcondition like, $s_1 \neq t_1 \Rightarrow s_0 = \hat{?}$, involves generating (on demand) the assignments where the test $s_1 = t_1$ fails, and testing $s_0 = \hat{?}$ for each assignment.

For expressions on $SR1$, equivalence is tested by testing the sign expressions as above, and testing the real subexpressions for syntactic equivalence. This approach may be incomplete for arbitrary sign quadratics since although it treats the real subexpressions as if they where independent sign variables, the real expressions are mutually constrained through shared variables on $\Re$.

As was discussed in the section on design (section 7), subsumption is essential to verifying that a device achieves a desired qualitative behavior, even when the behaviors of the device's components are described quantitatively. An expression $e1(\overline{x})$ subsumes $e2(\overline{x})$ if $e1(\overline{x}) \supset e2(\overline{x})$ for all assignments of variables in $\overline{x}$ (P91). Presuming that the variables in $\overline{x}$ are real valued, then if $e1(\overline{x})$ and $e2(\overline{x})$ denote reals, subsumption reduces to equivalence (P92). Otherwise, $e1$ must denote a sign to subsume $e2$, and if $e2(\overline{x})$ denotes a real, subsumption reduces to $e1(\overline{x}) \supset [e2(\overline{x})]$ (P93).

What remains is to determine subsumption between sign expressions. We consider the two most common cases. If $e1(\overline{x})$ and $e2(\overline{x})$ are expressions in $SR2$, then subsumption reduces to equivalence (P94). For example, $[H]$ subsumes $d \times g \times H$ since $[d \times g \times H] \Rightarrow [H]$. The next most common case is a sum of products, such as $-[a] \times [b] + [c] \supset -[a \times b - c]$. To compute subsumption we first use the homomorphisms (P49–P52) to turn the sign products into real products, for example:

$$\begin{aligned} -[a] \times [b] + [c] &\Rightarrow [-a \times b] + [c] \\ -[a \times b - c] &\Rightarrow [-a \times b + c] \end{aligned}$$

Next using the property (P95):

$$[a] \subset [b] \ + \ [c] \Rightarrow [a - c] \subset [b]$$

then $e1(\overline{x})$ subsumes $e2(\overline{x})$ if $e2(\overline{x})$ can be reduced to $\hat{0}$ by (non-deterministically) subtracting the real expression of each summand in $e1(\overline{x})$ from one of the real expressions of the summands in $e2(\overline{x})$ (P96). In the above example, to show that:

$$[-a \times b] + [c] \supset [-a \times b + c]$$

we subtract $-a \times b$, from the right side, producing:

$$[(-a \times b + c) - (-a \times b)] \Rightarrow [c]$$

Subtracting the second summand, $[c]$, then reduces this to $\hat{0}$. Note that Minima uses a somewhat more sophisticated approach which avoids useless subtractions and accounts for the fact that $s^{2i+1} \Rightarrow s$.

Given the predicates on expressions described thus far, the predicates on equations are straightforward. An equation is a tautology if it is satisfied for all assignments, and inconsistent if it is satisfied for no assignments. Testing tautology simply involves applying the relation of an equation as a predicate on the two expressions. We have not developed general predicates for inconsistency, but this would involve a set of predicates on expressions that are analogous to those above, with the exception that they test possibility rather than necessity (i.e., possibly $=$, $\subset$ and $\approx$). Inconsistency would then be tested in an analogous manner to tautology.

Two equations are equivalent ($\Leftrightarrow$) if they denote the same relation. To test a pair first the two sides of each equation are brought together using the appropriate cancellation laws for $=$, $\subset$ and $\approx$, as is discussed in the section on qualitative composition (section 12.2). Next each resulting equation is canonicalized, and then the relation symbol and expressions are tested for equivalence. Finally applicability conditions are also tested for equivalence, as in "for $0 \notin u$" in $s = t \times u$ for $0 \notin u$.

An equation subsumes another if the first is satisfied by all assignments satisfying the second; that is, the second equation implies the first ($e1 \Leftarrow e2$). A pair is tested by canonicalizing the equations as we did for equivalence, and using combinations of the following propositions to compare the expressions and relation symbols:

Proposition 97 (Qualitative Comparison Laws) *Given $w, x, y, z \in S' \cup \Re$, then:*

$$\begin{aligned} x = y &\Rightarrow x \subset y, \\ x \subset y, w \subset x, y \subset z &\Rightarrow w \subset z, \text{ and} \\ x \subset y &\Rightarrow x \approx y. \end{aligned}$$

An additional condition for qualitative equivalence holds, but is not easily used (P98):

$$x \approx y,\ (x \subset y \Rightarrow w \subset z),\ (y \subset x \Rightarrow z \subset w) \Rightarrow w \approx z$$

This completes our presentation of the simplification, composition and comparison operations that comprise Minima's skills for manipulating interactions. There remain several open questions about completeness for the full algebra in the multivariate case. We believe that the most important directions involve integrating Minima with order of magnitude, traditional interval algebra and differential equation techniques, and the use of tasks to identify strategies for guiding algebraic manipulation for tractable subalgebras of $SR1$.

14 Conclusion

The apparently weak properties of a qualitative algebra have led some to conclude that we must turn instead to extra-mathematical properties of physical systems. Instead we construct a powerful qualitative algebra, $SR1$, with three distinctive features: First, $SR1$ is a hybrid algebra that merges the signs and reals into a single domain. This allows a problem solver to select the right level of abstraction intermediate between traditional qualitative and quantitative algebras. Second, $SR1$ uses a complete sign algebra, which contains a number of important and unique properties, unexplored by previous researchers. Third, $SR1$ provides true equality and subset as well as the much weaker relation of qualitative equality, used in other qualitative algebras. This permits substitution of equals, qualitative composition and hybrid resolution, three rules of inference that capture a wide class of inferences needed for design and verification, as well as embodying many of the inferences of qualitative arithmetic, composition of monotonicity, inequality algebra, transition analysis, qualitative resolution, and traditional algebra.

Through $SR1$ and Minima we have demonstrated progress towards a theory of interactions that allows us to represent algebraically just the interesting features of interactions, and that captures skills for composing and comparing descriptions of interactions. This theory is sufficiently expressive to determine for a variety of fluid regulation devices what behaviors they *will achieve* — not just what is impossible — and embodies in a simple manner both our earlier work on qualitative/quantitative symbolic algebra, and many existing formalisms for algebraically describing and individually manipulating interactions.

15 Acknowledgments

I would like to thank Richard Zippel for teaching me some of the ins and outs of symbolic algebra, and Elisha Sacks for showing me how Minima could talk to

Macsyma. I would like to thank Randy Davis, Olivier Raiman, Brian Falkenhainer, Todd Cass, Vijay Saraswat and Leah Williams for many enlightening discussions. I would also like to thank the following people for comments on this work and its presentation: Dan Weld, Jeff Van Baalen, Paul Resnick, Leo Joskowicz, Walter Hamscher, Johan de Kleer, Randy Davis, Mike Brady and Danny Bobrow.

References

[1] D. Berleant and B. Kuipers. Combined Qualitative and Numerical Simulation with Q3. In *Proceedings of the Workshop on Qualitative Physics*, pages 140–152, Lugano, Switzerland, July 1990.

[2] R. Bhaskar and A. Nigam. Qualitative Physics using Dimensional Analysis. Technical report, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1988.

[3] D. Bobrow, editor. *Qualitative Reasoning About Physical Systems.* MIT Press, Cambridge, 1985.

[4] T. Bylander. The Qualitative Difference Resolution Rule. In *Proceedings of AAAI*, August 1991.

[5] C. Chiu and B. Kuipers. Comparative Analysis and Qualitative Integral Representations. In *Proceedings of the Workshop on Qualitative Physics*, Palo Alto, August 1989.

[6] J. de Kleer and J. Brown. A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24:7–83, Dec. 1984.

[7] J. de Kleer and G. J. Sussman. Propagation of Constraints Applied to Circuit Synthesis. *Circuit Theory and Applications*, 8:127–140, 1980.

[8] T. Dean. Temporal Imagery: an Approach to Reasoning about Time for Planning and Problem Solving,. Research Report 433, Yale University, October 1985.

[9] J. Dormoy and O. Raiman. Assembling a Device. In *Proceedings of AAAI*, pages 330–336, August 1988.

[10] K. Forbus. Qualitative Process Theory. *Artificial Intelligence*, 24:85–168, Dec. 1984.

[11] B. Kuipers. Commonsense Reasoning about Causality: Deriving Behavior from Structure. *Artificial Intelligence*, pages 169–203, Dec. 1984.

[12] B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29:289–338, Sep. 1986.

[13] B. Kuipers and D. Berleant. Using Incomplete Quantitative Knowledge in Qualitative Reasoning. In *Proceedings of AAAI*, pages 324–329, August 1988.

[14] O. Mayr. *The Origins of Feedback Control.* MIT Press, Cambridge, 1970.

[15] R. Moore. *Methods and Applications of Interval Analysis.* SIAM Studies in Applied Mathematics. SIAM, Philadelphia, 1979.

[16] J. Moses. Algebraic simplification: A guide for the perplexed. *Communications of the ACM*, 14(8):527–537, Aug. 1971.

[17] O. Raiman. 'Order of Magnitude Reasoning. In *Proceedings of AAAI*, pages 100–104, August 1986.

[18] E. Sacks. Qualitative Mathematical Reasoning. In *Proceedings of IJCAI*, pages 137–139, August 1985.

[19] E. Sacks. 'Hierarchical Reasoning about Inequalities. In *Proceedings of AAAI*, pages 649–654, August 1987.

[20] R. Simmons. Representing and Reasoning about Change in Geological Interpretation. TR 749, MIT Artifical Intelligence Lab, Cambridge, December 1983.

[21] R. Simmons. 'Commonsense' Arithmetic Reasoning. In *Proceedings of AAAI*, pages 118–124, August 1986.

[22] R. M. Stallman and G. J. Sussman. Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-aided Circuit Analysis. *Artificial Intelligence*, 9:135–196, 1977.

[23] P. Struss. Problems of Interval-based Qualitative Reasoning. In D. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning About Physical Systems.* Morgan Kauffman, San Mateo, 1990.

[24] L. Trave-Massuyes and N. Piera. The Orders of Magnitude Models as Qualitative Algebras. In *Proceedings of IJCAI*, pages 1261–1265, August 1989.

[25] D. Weld and J. de Kleer, editors. *Readings in Qualitative Reasoning About Physical Systems.* Morgan Kauffman, San Mateo, 1990.

[26] B. C. Williams. Qualitative Analysis of MOS Circuits. *Artificial Intelligence*, pages 281–346, Dec. 1984.

[27] B. C. Williams. Doing Time: Putting Qualitative Reasoning on Firmer Ground. In *Proceedings of AAAI*, pages 105–112, Philadelphia, August 1986.

[28] B. C. Williams. MINIMA: a Symbolic Approach to Qualitative Algebraic Reasoning. In *Proceedings of AAAI*, pages 264–269, St. Paul, August 1988.

[29] B. C. Williams. Capturing How Things Work: Constructing Critical Abstractions of Local Interactions. In *AAAI Workshop on the Automatic Generation of Abstractions and Approximations*, Boston, July 1990.

[30] B. C. Williams. Interaction-based Invention: Designing Novel Devices from First Principles. In *Proceedings of AAAI*, July 1990.

[31] B. C. Williams. A Theory of Interactions: Unifying Qualitative and Quantitative Algebraic Reasoning — Extended Report. Technical report, Xerox Palo Alto Research Center, Palo Alto, CA, 1991.

[32] K. Yip. Understanding Complex Dynamics by Visual and Symbolic Reasoning. *Artificial Intelligence*, 1991. this volume.

A Monotonicity and its Composition

This section demonstrates that, by mapping monotonicity functions into signs of differences, Kuiper's composition of monotonicity functions[11] falls out naturally as a byproduct of Minima's simplifier. What is gained is the ability to tap into the much greater expressivity of $SR1$, and the power of Minima/Macsyma's capabilities for symbolic manipulation of hybrid equations and partial differentiation.

Kuiper's formalism consists of four monotonicity functions — M^+, M^-, M_z^+ and M_z^- — together with the addition, subtraction and composition of these functions. Additionally it permits $+$ and $\times$ on $\Re$, $=$, and the predicates "constant" and "> 0". By introducing ∂v as a change in a variable v,[35] the four monotonicity functions[36] map to $SR1$ expressions as follows:

$$\begin{array}{lcl} y = M^+ & \Leftrightarrow & [\partial y] = [\partial x] \\ y = M^- & \Leftrightarrow & [\partial y] = -[\partial x] \\ y = M_z^+ & \Leftrightarrow & [\partial y] = [\partial x], [y] = [x] \\ y = M_z^- & \Leftrightarrow & [\partial y] = -[\partial x], [y] = -[x] \end{array}$$

Since the M functions map to sign ([ ]), adding and subtracting these functions correspond to sign addition and subtraction; for example:

$$M^+(x) + M^+(y) = [\partial x] + [\partial y]$$

Using the fact that [ ] on signs behaves as an identity function (i.e., $[[x]] = [x]$), the composition of M functions maps nicely to the composition of [ ]. For example:

$$M^+(M^+(x)) = [[\partial x]]$$

Finally, the predicate constant(x) maps to $\partial x = 0$, and $x > 0$ to $[x] = \hat{+}$.

Kuiper's formalism provides three sets of rules: a set for mapping arithmetic constraints to monotonicity relations, a set for composing monotonicity functions, and a set for adding/subtracting monotonicity functions. Figure 5 shows each of these composition rules on the left, and on the right the corresponding $SR1$ expression before and after simplification. For each line the unsimplified expression corresponds to the left side of the composition rule, and the simplified expression corresponds to the right side.

Minima performs a wide variety of operations on monotone functions not expressed in Kuiper's rules. For example, monotone functions may be

[35] We abuse the standard use of ∂v as "the differential of v," but only in that a relation between ∂v's does not presume the existence of a differentiable function relating those variables. This is important in that a monotone function need not be differentiable.

[36] In both Kuipers' and Forbus' formalisms the functions are not only monotone, but strictly increasing or decreasing.