An Algebraic Foundation for Truth Maintenance*

Allen L. Brown, Jr., Dale E. Gaucas, and Dan Benanav
GE Corporate Research and Development Center
P.O. Box 8
Schenectady, New York 12301

Abstract

We have recast the problem of truth maintenance in a setting of algebraic equations over Boolean lattices. If a method of labeling propositions to justify them according to some reasoning agent's constraints of belief happens to conform to the postulates of Boolean lattices, the labeling system can be reformulated as an algebraic equation solving system. All truth maintenance systems known to us can be so reformulated. This note summarizes our investigations into the existence and structure of solutions of these algebraic systems. Our central result is a unique factorization theorem for lattice equational systems and their solutions. Our theoretical results are interpreted to compare various styles of truth maintenance and to reveal certain computational difficulties implicit in the algebraic structure of truth maintenance.

I. Introduction

Lattice-theoretic truth maintenance is a single theoretical framework that subsumes various notions of truth maintenance, including the assumption-based justifications reported by de Kleer [de Kleer, 1984, de Kleer, 1986a, de Kleer, 1986b, de Kleer, 1986c] and the nonmonotonic justifications reported by Doyle [Doyle, 1979a, Doyle, 1979b, Doyle, 1978].and Goodwin [Goodwin, 1982, Goodwin, 1985, Goodwin, 1984, Goodwin, 1987]. Our complete body of work on lattice-theoretic truth maintenance includes

- An analysis of the algebraic structure of truth maintenance
- An investigation of the abstract and concrete computational complexity of truth maintenance
- A formal account of the embedding of other forms of truth maintenance in the lattice-theoretic paradigm

In this note we focus on the first aspect, because of its intrinsic interest, and because this aspect is a precursor to the others. Our express aim here is to present the lattice-theoretic account of truth maintenance, cite the more important *algebraic* results *vis a vis* this account, and interpret these results so as to cast a qualitative light on various computational considerations of truth maintenance. Readers interested in other aspects of our theoretical work or our practical experience with an implementation embodying this theory are referred to [Benanav *tt a*/., 1986].

The initial motivation for this work was the desire to unify in a single abstraction the truth maintenance paradigm of Doyle and Goodwin, and that of de Kleer. The systems of these investigators can be viewed as constraint propagation mechanisms. Given a disjunctive set of sets of premises and a set of (monotonic) deductive constraints, de Kleer's ATMS tells a client problem solving system what things it is currently obliged to believe, assuming one or another of the sets of premises. Doyle's and Goodwin's TMS's, on the other hand, tell the client problem solving system what things it is currently obliged to believe, given a single set of premises under deductive constraints, some of which may be nonmonotonic in nature.* Our original intuition was that it should be possible to account *simultaneously* for multiple sets of premises *and* nonmonotonic deductive constraints.**'***

This intuition arose from the striking similarity observed between the computations of truth maintenance systems and the computations of global flow analysis

*This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under USAF/Rome Air Development Center contract F30602-85-C-0033. Views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official opinion or policy of DARPA or of the U.S. government.

*A monotonic deductive constraint obliges a rational agent to believe its consequent, given that it currently believes all of its antecedents. A *nonmonotonic* deductive constraint obliges a rational agent to believe its consequent given that it believes all of its monotonic antecedents and none of its nonmonotonic antecedents.

"The intellectual challenge of unifying these two approaches to truth maintenance is sufficient motivation for proceeding. Nonetheless, we note that de Kleer [de Kleer, 1086b], and Morris and Nado [Morris and Nado, 1986] are practically motivated to augment their assumption-based truth maintenance systems to support some form of nonmonotonic justification. In our approach nonmonotonicity will be "built-in" rather than "added-on". Although we will not do so here, it can be shown that our conceptually parsimonious approach is at a computational advantage relative to the attempts of de Kleer, and Morris and Nado.

***We have recently been made aware of the work of McDermott [McDermott, 1983] whose perspective on truth maintenance has much in common with our own. Indeed, his concrete solution to what we will eventually define as *even equational systems* appears to be identical to ours, though arrived at from a quite different point of departure. Our investigation is broader in both the scope of equational systems investigated, and in the characterization of those systems' structures and solution spaces.

that underly modern optimizing compilers [Aho and Ullman, 1977, Hecht, 1977, Schaeffer, 1973, Waite and Goos, 1984]. Global flow analysis can be couched in the following terms: Given the constraints imposed by individual program statements and their interconnecting topology, what facts is a reasoning agent (in this case concerned with programs) obliged to believe about the state of computation at various points in the program's control flow? In a sense the information propagation problem solved by global flow analysis can be viewed as the dual of the truth maintenance problem. The former assigns propositions to contexts established by various paths through a program. The latter assigns contexts of belief to propositions under various deductive constraints. There are two principal methods of solving information propagation problems. Both hinge on solving systems of equations whose unknowns range over the domain of an algebraic lattice. The work that we will describe presently retains the idea of equations over a lattice, but for various technical reasons (principally nonmonotonic constraints) the solution methods used in global flow analysis are inappropriate. A rather different solution method has been developed.

II. Lattice Equational Systems

Let B be a Boolean lattice equipped with the usual meet, join, and complementation operators; a partial order, $<$; and maximum and minimum elements, $\top$ and $\bot$, respectively.* A complete account of such structures can be found in any of [Balbes and Dwinger, 1974, Birkhoff, 1967, Skornjakov, 1977]. Elements of B will be called *situations*, and will be denoted by A and B. A and B (possibly subscripted) are *lattice expressions* in B. Moreover, if A and B are expressions in B then so are $A \vee B$, $A \wedge B$, $\overline{A}$ and $\overline{B}$. Especially important to us will be the existence of the partial order, the complement, maximum and minimum elements, and the mutual distributivity of meet and join.

A *lattice unknown* is a super- and/or subscripted s. Each lattice expression in B and unknown is a *lattice form* in B. Moreover, if X and Y are forms in B then so are $X \vee Y$, $X \wedge Y$, $\overline{X}$ and $\overline{Y}$. Individual (fixed) lattice forms in B will be denoted by X and Y, possibly subscripted. Every fact or proposition has an associated unknown. Note that a proposition and its negation have distinct associated unknowns. Indeed, an unknown corresponds exactly to a *node* as that term is used by Doyle, Goodwin, and de Kleer. A *lattice equation over* B is a relation of the form $X = Y$ where X is a lattice unknown and Y is a lattice form. A *lattice equational system over* B, Σ, is any collection of lattice equations over B such that the total number of lattice unknowns occurring on the right-hand sides of the equations is finite and any lattice unknown occurs at most once on the left-hand side of an equation. The equation on whose left-hand side s appears will be called the s equation,

Σ will be sub- or superscripted when it is useful to distinguish among various equational systems. Unless the context is ambiguous, we will freely say 'system' without modifiers. A lattice equational system should be interpreted as encoding the way a reasoning agent's belief (or disbelief) in a collection of propositions entails belief in others. If Σ is a lattice equational system such that the right-hand side of each equality is of the form $\bigvee_i \bigwedge_j X_{ij}$ where each X_{XJ} is an element of B or an unknown (possibly complemented), then Σ is said to be in *disjunctive normal form.** Since we can transform any form into disjunctive normal form, we will usually treat forms over B and lattice equational systems as if they were in disjunctive normal form.

A *solution* to a lattice equational system, Σ, is a function, T, from the lattice unknowns into B such that if for each equation in the system, each unknown s in the equation is replaced by $T(s)$ the equation holds in B. Moreover, T takes any unknown, s, not on the left-hand-side of some equation in Σ into 1, and in that regard the system Σ implicitly has the equation s = .1. We will interpret lattice equations as constraints. A solution, then, is a *labeling* of propositions with situation *. In particular, the situations are those in which a reasoning agent is obliged to believe the correspondingly labeled proposition given acceptance of the constraints imposed by the system. We will often subscript T' with the name of the system of which it is a solution. A *justification* of a disjunctive normal form lattice equational system, Σ, is an ordered pair, $d = \langle s, X \rangle$, where s appears on the left-hand side of some equation in Σ and X is a d'sjunct on the right-hand side of that same equation. Also, s is called the *consequent* of the justification d and each conjunct of the disjunct X is called a *nonmonotonic* or *monotonic antecedent* of d depending on whether or not it is complemented. The sets of monotonic and nonmonotonic antecedents of d are respectively denoted $a(d)$ and $a(d)$. A justification, d, is *valid* with respect to a situation, A, and a solution, T, of an equational system Σ if and only if,

$$A \leq \bigwedge_{s \in \alpha(d)} \Gamma(s) \wedge \bigwedge_{s \in \delta(d)} \overline{\Gamma(s)}$$

We will write Valid(A,d,Γ) to indicate that d is valid with respect to A and solution Γ. A solution, Γ, is *well-founded with respect to a lattice equational system*, Σ, *at lattice unknown*, s, if and only if either $\Gamma(s) = \bot$, or $\Gamma(s) = \bigvee_i A_i$, and for each A_i, there is a partially ordered set, $\langle \mathcal{P}_{A_i}, \prec_{A_i} \rangle$, such that $\mathcal{P}_{A_i}$ is a set of justifications from Σ and

1. There is a justification, d in $\mathcal{P}_{A_i}$, whose consequent is s

*In this report we assume B to be a recursive set, its operators to be total recursive functions, and its partial order to be a recursive relation.

*We use disjunctive normal form for notational convenience. While its existence is required in establishing some of the formal results that we cite, it plays no essential role in lattice-theoretic truth maintenance computations.

2. For every justification d, in $\mathcal{P}_{A_i}$, Valid(j4,-,d,r)
3. Every unknown, *s'*, that is a monotonic antecedent of some *d* in $\mathcal{P}_{A_i}$ is also the consequent of some justification *d'* in $\mathcal{P}_{A_i}$ and $d' \prec_{A_i} d$

A solution to a lattice equations] system is *well-founded* if and only if it is well-founded with respect to the system at every lattice unknown mentioned in the system.

We interpret justifications, validity and well-foundedness in the following way: Validity describes the circumstances under which the consequents of a justification are to be believed given the belief status of the antecedents. A justification therefore constitutes an independent source of support justifying belief in a consequent. Chaining justifications together constitutes a supporting argument. Since we wish our arguments to be noncircular, we impose an additional condition, well-foundedness, to guarantee that state of affairs.

Let us first consider some uninterpreted equational systems, all taken to be over the Boolean lattice, B, having at least two distinct elements: The system Σ_1

$$s = s$$

has one well-founded solution, $\{s = \bot\}$. On the other hand, any of $\{\{s = A\} | A \neq \bot \text{ and } A \in \mathcal{B}\}$ are also solutions, though not well-founded. The system Σ_2

$$s = \overline{s}$$

the classical "odd loop" of Doyle's TMS, has no solutions, well-founded or otherwise. The system E3

$$\begin{aligned} s_1 &= s_2 \\ s_2 &= s_1 \end{aligned}$$

has one well-founded s o l $\{s_1 = \bot, s_2 = \bot\}$. The system Σ_4

$$\begin{aligned} s_1 &= \overline{s}_2 \\ s_2 &= \overline{s}_1 \end{aligned}$$

has well-founded solutions $\{\{s_1 = A, s_2 = \overline{A}\} | A \in \mathcal{B}\}$ The system Σ_5

$$\begin{aligned} s_1 &= \overline{s}_1 \vee s_2 \\ s_2 &= \overline{s}_2 \vee s_1 \end{aligned}$$

has a single solution $\{s_1 = \top, s_2 = \top\}$, and it is not well-founded. Finally, Σ_6

$$\begin{aligned} s_1 &= \overline{s}_2 \wedge \overline{s}_3 \\ s_2 &= \overline{s}_1 \wedge \overline{s}_3 \\ s_3 &= \overline{s}_1 \wedge \overline{s}_2 \end{aligned}$$

has well-founded solutions $\{\{s_1 = \bot, s_2 = A, s_3 = \overline{A}\} | A \in \mathcal{B}\} \cup \{\{s_1 = A, s_2 = \bot, s_3 = \overline{A}\} | A \in \mathcal{B}\} \cup \{\{s_1 = A, s_2 = \overline{A}, s_3 = \bot\} | A \in \mathcal{B}\}$. **If** $\mathcal{B} = \{\top, \bot\}$ **and interpreting** $\top$ **as** "IN" and ± as "OUT", it should be apparent to readers familiar with the TMS's of Doyle and Goodwin how lattice equational systems correspond to their TMS nodes and justifications.

The correspondence with de Kleer's ATMS is a little harder to convey, and we shall attempt only an approximation here.* We shall do this by actually interpreting a lattice equational system with respect to a toy application. Imagine a simple series-connected circuit consisting of a voltage source, V, of 5 volts connected to resistor *R1* at node n_1, which in turn is connected to resistor R_2 at node n_2, which is connected to ground. The application is a program that diagnoses ground faults in electrical circuits. In its truth maintenance database it has the following system of equations, Σ_7:

$$\begin{aligned} s_1 &= A \\ s_2 &= B_1 \\ s_3 &= B_2 \\ s_4 &= s_1 \wedge s_2 \wedge s_3 \end{aligned}$$

The situations A, Bi, and B_2 respectively correspond to the assumption that the voltage source, V, and resistors, *R* and R_2, are working. *s-[* corresponds to the proposition that voltage at node n_1 is held at 5 volts. s2 corresponds to the conjunctive proposition that the current into the resistor and node n_1 is the same as the current out of the resistor R_1 at node n_2 and that the voltage drop across the resistor is the product of its resistance and the current through. 63 corresponds to the conjunctive proposi- R_2 tion that the current into the resistor R_2 at node n_2 is the same as the current out of the resistor at ground and that the voltage drop across the resistor is the product of its resistance and the current through. Finally, s_4 corresponds to the proposition that the voltage at node n_2 is the product of 5 volts and the resistance of R_2 divided by the sum of the resistances of R_1 and R_2. The equations can now be interpreted as saying that the propositions associated with s_1, s_2, and s_3 hold whenever the corresponding assumptions can be believed. The proposition associated with 34 is believed whenever the propositions associated with s_1, s_2, and s_3 are believed. A solution to this system will tell us the circumstances under which the various propositions are to be believed. Since the well-founded solution is $\{s_1 = A, s_2 = B_1, s_3 = B_2, s_4 = A \wedge B_1 \wedge B_2\}$, a reasoning agent believes the propositions associated with s_1, s_2, s_3, and s_4 in situations whose meets with (respectively) A,B1,B2 and $A \wedge B_1 \wedge B_2$ are not $\bot$.

In the foregoing examples we have made implicit use of the fact that any set of TMS or ATMS justifications has equivalent renderings in the lattice-based formalization. For our last example we consider the classical problem of adding facts to or deleting facts from worlds or

•Readers interested in the precise details of encoding these other truth maintenance systems in the lattice-theoretic paradigm should consult [Ben anav *et al.*, 1986].

states. To begin with, we interpret situations as worlds or states. We have already asserted that every fact or proposition, p, has an associated unknown, say 3_p. We will also posit additional unknowns, s_a and sd, corresponding to the beliefs (respectively) that *p* has been added and that *p* has been deleted. Consider now the system of equations

$$\begin{aligned} s_a &= A \\ s_d &= B \\ s_p &= s_a \wedge \bar{s}_d. \end{aligned}$$

The well-founded solution of this system is $\{s_a = A, s_d = B, s_p = A \wedge \overline{B}\}$. Our interpretation of this solution is that a reasoning agent believes *p* just in case he believes himself to be in a world or state whose meet with A -B_is not ±. In addition, we can use the lattice partial order to encode inheritance among worlds. Notice that the_fact *p* will be added to *any* world, *A'* such that $A' \leq A \wedge \overline{B}$, and deleted from any world, *B'* such that $B' \leq B$

III. The Existence of Solutions

We have seen how lattice-theoretic truth maintenance is connected to some well known models of truth maintenance; we now turn to the challenge of actually solving truth maintenance problems is this new paradigm. We have already seen in Σ_2 that solutions need not exist, but even if they do (as in Σ_7), there may not be well-founded ones. It is well known that general polynomial equations in rational coefficients cannot be solved by applying the operations of addition, multiplication, and rational root extraction to their coefficients [Birkhoff and MacLane, 1965, van der Waerden, 1953]. By analogy we might ask about the solvability of lattice-equational systems by taking meets, joins, and complements of lattice expressions appearing in the equations. Put another way, could it be the case that the equations are not solvable by applying the obvious operations to the available data? To answer this question we must first formalize our notion of the 'available data'.

A *surface element* of a lattice equational system E is an element of *B* that actually appears in E. The Boolean lattice generated by meets, joins, and complements over the surface elements is called the *surface lattice.* An *atom* of the Boolean lattice, 5, is any element, $A \in \mathcal{B}$, $A \neq$ _L, such that there is no $B \in \mathcal{B}$ satisfying $A > B > \perp$. A lattice is *atomic* if each of its elements is the join of atoms. If E contains only a finite number of equations, the number of surface elements is finite and thus the surface lattice is atomic. The atoms of this lattice will be called *surface atoms.* A *surface solution* is one such that for every lattice unknown, s, that appears in E, *T(s)* is in the surface lattice. Consider again the system, E4. Note that it has many possible well-founded solutions (depending on the Boolean lattice with respect to which the system is being interpreted) of which only two, $\{s_1 = \top, s_2 = \perp\}$ and $\{s_1 = \perp, s_2 = \top\}$ are surface. Our question posed in the last paragraph is answered by the following:

Theorem III.I *If a finite lattice equational system* E *over B has a well-founded solution, then, it has a well-founded surface solution.*

Thus we see that if there are any well-founded solutions at all, we are guaranteed that some of them can be computed by taking meets, joins, and complements over the available data.

Thus far we have established a framework within which we can formally describe truth maintenance problems *and* within which solutions can be connected with the available data in the equations. For this framework to be truly useful we must provide a way of finding solutions other than by blindly enumerating candidates and testing them. Suppose we could obtain a solution. How do we know that this is the *only* solution? Or even the only *surface* solution? To convey some idea of the challenge of this problem consider the system Σ_8

$$\begin{aligned} s &= s \wedge \bigwedge_{1 \leq k \leq n} \bar{s}_{2k} \\ s_1 &= s \vee \bar{s}_2 \\ s_2 &= \bar{s}_1 \\ \vdots &\quad \vdots \quad \vdots \\ s_{2n-1} &= s \vee \bar{s}_{2n} \\ s_{2n} &= \bar{s}_{2n-1}. \end{aligned}$$

This system has 2^n well-founded surface solutions, *s* is always ± and we are free to choose ± *or* T as the value of *each* of the odd-indexed unknowns. The usual method of solving algebraic equational systems is by using substitution together with other "legal" (with respect to the algebraic system in question) transformations to produce a new lattice equational system whose solutions are also solutions of the original. Because of the algebraic nature of the meet and join operators there is no obvious way of effecting such a transformation. Before introducing the more novel transformation that we will need, let us first formalize the notion of substitution that we will be using.

Let E_, be E less its *s* equation. Systems will be presented always according to some fixed lexical order. This is possible since each system is obviously a recursive set. Hence it is reasonable to speak of the n^{th} occurrence of the unknown, a, on the right-hand side of an equation in E. We define a *local substitution, a.n* as follows: If the *s'* equation is the locus of the n^{th} occurrence, then $\sigma_{s,n}(\Sigma)$ is E_v together with a new *s'* equation wherein the n^{th} occurrence of *s* is replaced by the right-hand side of the *s* equation in Σ. If there is no n^{th} then $asn_{fl}(\Sigma) = \Sigma$. Suppose there are *k* *s*'s in E before the *s* equation, m in the *s* equation, and n after the *s* equation. The (global) *substitution transformation* of Σ under s, (o,(E), is

$$\sigma_{s,1}(\cdots\sigma_{s,k-1}(\sigma_{s,k}(\sigma_{s,k+m+1} (\cdots\sigma_{s,k+m+n-1}(\sigma_{s,k+m+n}(\Sigma))\cdots)))\cdots)$$

This transformation has the effect of replacing every right-hand side occurrence of *s* (except those in the *s* equation) with the right-hand side of the s equation. The following lemma suggests the other transformation necessary for computing solutions.

Lemma III.1 *Let X_1, X_2 and X_3 be expressions having no occurrence of s, then*

$$\begin{aligned} s &= X_1 \vee (X_2 \wedge s) \vee (X_3 \wedge \bar{s}) \\ &\Leftrightarrow \quad (X_1 \vee X_3) \leq s \leq (X_1 \vee X_2) \\ &\Leftrightarrow \quad (\Gamma(X_1) \vee \Gamma(X_3)) \leq \Gamma(s) \leq (\Gamma(X_1) \vee \Gamma(X_2)) \end{aligned}$$

where Γ is a solution of a system including the s equation.

Let the s equation be rearranged to have the form: $s = X_1 \vee (X_2 \wedge s) \vee (X_3 \vee \bar{s})$. The distributive nature of the Boolean lattice guarantees that we can always do this. The *minimization transformation* of Σ under s, $\mu_s(\Sigma)$, is Σ_{-s} together with the equation $s = X_1 \vee X_3$. This transformation is semantically equivalent to having substituted $\perp$ for every occurrence of s on the right-hand side of the s equation. Or put another way, we are taking the lower bound of the solution interval defined in the previous lemma.

We ask then whether or not minimization and substitution can be used to produce a solution. The answer will always be in the affirmative for an important class* of equational systems. We may apply these transformations to the original system of equations in such a way as to yield a new system free of unknowns on the right-hand side. The resulting system constitutes a solution for the original system of equations. Before discussing the exact method of applying these transformations, however, we offer the following apparently technical but actually qualitatively important result about minimization and substitution.

Lemma III.2 *A well-founded solution, Γ, of a lattice equational system, Σ, is a well-founded solution of $\sigma_s(\Sigma)$ unless σ_s involves a local substitution for a complemented occurrence of s. If $\mu_s(\Sigma) = \Sigma'$, then a well-founded solution Γ' of Σ' is a well-founded solution of Σ.*

A close examination of the proof would reveal that the substitution operation has the property that it preserves "solution-ness" but may lose well-foundedness. On the other hand, minimization preserves well-foundedness, in the sense that any well-founded solution to the original system that *persists* in being a solution to the transformed system is still well-founded. Consider first the system Σ_4. Applying the transformation σ_{s_2} yields the system Σ_4'

$$\begin{aligned} s_1 &= s_1 \\ s_2 &= \bar{s}_1 \end{aligned}$$

which has the same surface solutions as Σ_4, but only the second of them is well-founded. Applying μ_{s_1} to Σ_4' yields

$$\begin{aligned} s_1 &= \perp \\ s_2 &= \bar{s}_1 \end{aligned}$$

*The well-known truth maintenance systems in the literature only guarantee well-founded solutions for even equational systems.

which has only one solution, $\{s_1 = \perp, s_2 = \top\}$.

A *process* for Σ is any functional composition of minimizations and substitutions. μ_s, $\sigma_{s,n}$ and σ_s are all processes for any lattice unknown, s. If π is a process, so are $\mu_s \circ \pi$, $\sigma_{s,n} \circ \pi$ and $\sigma_s \circ \pi$. A *terminal* process for Σ, denoted τ, is a process such that for every process π, $\tau \circ \pi(\Sigma) = \tau(\Sigma)$.

A *path of length* n from s_0 to s_n is a sequence of triples of the form

$$(X_1, Y_1, s_1), (X_2, Y_2, s_2), \ldots, (X_n, Y_n, s_n)$$

where $X_i \in \{s_{i-1}, \bar{s}_{i-1}\}$, X_i is an antecedent of the Y_i disjunct of the s_i equation in Σ, and $1 \leq i \leq n$. X_i is a complemented (uncomplemented) unknown if it is a complemented (uncomplemented) conjunct of Y_i. Unknown s is connected to unknown s' if there is a path of any length from s to s'. A path is *odd* if it has an odd number of complemented unknowns and *even* otherwise. A system is odd (and even otherwise) if it has an unknown, s, and an odd path from s to s.

Theorem III.2 *Every even lattice equational system, Σ, has a terminal process, τ, such that $\Gamma_{\tau(\Sigma)}$ is a well-founded solution of Σ.*

An immediate consequence of the previous theorem is an algorithm that is analogous to Gaussian elimination [Birkhoff and MacLane, 1965, Gantmacher, 1959] that will always produce a well-founded solution for an even lattice equational system:

1. Let x be a LIFO queue of the equations in Σ; let y be an empty LIFO queue of equations in Σ; let z be an equation in Σ
2. Until x is empty, x becomes $\sigma_s(\mu_s(x))$ (the order of equations remaining invariant with respect to their left-hand sides) where s is the unknown on the left-hand side of the first equation in the queue, dequeue x to z, enqueue z to y
3. Until y is empty, y becomes $\sigma_s(y)$ (the order of equations remaining invariant with respect to their left-hand sides) where s is the unknown on the left-hand side of the first equation in the queue, dequeue y to z, enqueue z to x
4. End

This algorithm terminates with x being a queue of equations with constant right-hand sides (a solution). Step 2 is the analogue of forward elimination; step 3 corresponds to back substitution. Each (unspecified) order in which unknowns are removed from the queue, x, is an *elimination sequence*. Every such sequence produces a well-founded solution (for an even system) and it may be the *only* sequence that produces that particular solution. In re-examining Σ_4 above we implicitly applied this algorithm, first eliminating s_1 and thereby generating the first of the two well-founded surface solutions. Had we done the other elimination first, we would have obtained the second well-founded surface solution. Can we generate all of the surface solutions by varying the order of elimination? Unfortunately the answer is no as can be seen by examining the following system, Σ_9:

$$\begin{aligned} s_1 &= A \wedge \bar{s}_2 \\ s_2 &= (A \wedge \bar{s}_1) \vee \bar{s}_3 \\ s_3 &= \bar{s}_2 \end{aligned}$$

only three of whose four well-founded solutions can be produced by varying the order of elimination. As suggested by the statement of the theorem, there are odd lattice equational systerns some of whose elimination sequences do not produce solutions. Such a system is Σ_{10}:

$$\begin{aligned} s_1 &= \bar{s}_2 \\ s_2 &= \bar{s}_1 \\ s_3 &= s_1 \wedge \bar{s}_3. \end{aligned}$$

Addressing either of the aforementioned deficiencies requires the separability results of the next section.

IV. The Structure of Systems and Solutions

In this section we discuss some separability results for lattice equational systems. These results are of two classes: topological and algebraic. They are important because

- They provide the machinery from which all surface solutions to all equational systems may be generated
- They provide the basic perspective for analyzing the abstract complexity of the truth maintenance problem
- They suggest concrete "divide and conquer" algorithms for solving the truth maintenance problem by
  - supporting "lazy evaluation" of the solution *vis a vis* any given unknown
  - supporting incremental update of solutions as justifications are added or removed
  - enabling parallel solution methods

Let the equations of Σ be partitioned into equivalence classes in such a way that the s_1 and s_2 equations of Σ will be in the same equivalence class Σ', a subsystem of Σ, just in case s_1 is connected to s_2 in Σ and s_2 is connected to s_1 in Σ. Each equivalence class Σ' is a *strongly connected subsystem* of Σ. A partial order, $\leq_\Sigma$, can be defined on the strongly connected subsystems of an equational system such that for subsystems Σ' and Σ'', $\Sigma' \leq_\Sigma \Sigma''$ if and only if there is an s' equation of Σ' and an s'' equation of Σ'' such that s' is connected to s''. Σ'' is *minimal*, if and only if there exists no Σ' such that $\Sigma' \leq_\Sigma \Sigma''$. Given a partitioning of a lattice equational system Σ into strongly connected subsystems, there always exists at least one minimal strongly connected subsystem of Σ.

Proposition IV.1 *Every lattice equational system can be partitioned into a partial order of strongly connected subsystems. Moreover, each of these subsystems can be treated as if the unknowns whose corresponding equations are in other strongly connected subsystems were lattice elements (i.e., constants).*

Since we can solve each of the partitions separately, treating the unknowns whose corresponding equations are *not* in the strongly connected subsystem being solved as if they were expressions from the surface lattice (*i.e.*, constants), we can pursue a strategy of lazy evaluation and incremental update. Not only can the system be partitioned very efficiently [Aho *et al.*, 1974], but there are also efficient methods of updating this partially ordered partition as justifications are added (and deleted). As things change in the truth maintenance database, we compute new partitions and only *re*-solve for unknowns whose equations are in the same strongly connected subsystem as the changed equation, and (optionally) for unknowns in greater subsystems. For example, consider Σ_{10}. This system partitions into two strongly connected subsystems. The s_1 and s_2 equations form the first and lesser (in $\leq_{\Sigma_{10}}$) subsystem; the second strongly connected subsystem consists of the s_3 equation. We can solve the second subsystem, treating it as if s_1 were a constant. Note that the s_3 equation has a solution only in the case that $s_1 = s_3 = \bot$. This "forces" the solution of the first subsystem to be $\{s_1 = \bot, s_2 = \top\}$.

A lattice equational system, Σ, is *reduced* if and only if for every pair of unknowns s and s' whose equations are in the same strongly connected subsystem of Σ such that s is an antecedent of s' in Σ, s is a nonmonotonic antecedent of s'.

Theorem IV.1 *For every lattice equational system, Σ, there is a process, π, such that $\Sigma' = \pi(\Sigma)$ is reduced, and Γ is a well-founded solution of Σ if and only if it is a well-founded solution of Σ'.*

The utility of the previous theorem becomes clearer when we combine the computation of strongly connected subsystems of a system with reduction and minimization. Since we need to do only *substitutions* through uncomplemented occurrences of unknowns in the original system in order to reduce it, the resulting system has exactly the same well-founded solutions as the original. Reconsider now E_8. If we minimize with respect to s and reduce again we get the reduced system:

$$\begin{aligned} s &= \bot \\ s_1 &= \bar{s}_2 \\ s_2 &= \bar{s}_1 \\ \vdots &\;\;\vdots\;\; \vdots \\ s_{2n-1} &= \bar{s}_{2n} \\ s_{2n} &= \bar{s}_{2n-1}. \end{aligned}$$

The system has now been separated into n + 1 strongly connected subsystems, each of which is disconnected from the others, hence independently solvable. We see now how the 2^n solutions arise, since we have n repetitions of the system Σ_4.

If we combine the previous two results, and manipulate the proof of theorem III.2 we obtain

Theorem IV.2 *Let Σ be a lattice equational system whose surface lattice is $\{\top, \bot\}$. Every well-founded surface solution of Σ is produced by some sequence of minimizations and global substitutions.*

Thus if we restrict the lattice over which the equations are taken to correspond exactly to the TMS's of Doyle and Goodwin, each well-founded solution will be produced by some elimination sequence. Unfortunately, among the large number of elimination sequences, many (in the case of odd systems) do not produce solutions. Since we and others [McAllester] have independently shown the NP-completeness (in the size of the equational system) of solving this restricted class of lattice equational systems, there can be no "easy" characterization of the circumstances under which a particular elimination sequence will produce a well-founded solution.

We turn now to the question of how to find all the well-founded surface solutions for arbitrary systems. Let $\Gamma_1 \vee \Gamma_2$, $\Gamma_1 \wedge \Gamma_2$, and $A \wedge \Gamma$ denote functions such that $(\Gamma_1 \vee \Gamma_2)(s) = \Gamma_1(s) \vee \Gamma_2(s)$, $(\Gamma_1 \wedge \Gamma_2)(s) = \Gamma_1(s) \wedge \Gamma_2(s)$, and $(A \wedge \Gamma)(s) = A \wedge \Gamma(s)$. A *Goodwin projection* of a lattice element B with respect to an atom A, denoted $\gamma_A(B)$, is defined by

$$\gamma_A(B) = \begin{cases} \top & \text{if } A \leq B \\ \bot & \text{otherwise.} \end{cases}$$

We extend the notion of a Goodwin projection to expressions over lattice elements by

$$\begin{aligned} \gamma_A(B_1 \wedge B_2) &= \gamma_A(B_1) \wedge \gamma_A(B_2) \\ \gamma_A(B_1 \vee B_2) &= \gamma_A(B_1) \vee \gamma_A(B_2) \\ \gamma_A(\overline{B}) &= \overline{\gamma_A(B)} \end{aligned}$$

and to lattice equations by applying the projection to each constant term in each equation.

Theorem IV.3 *Γ_Σ is a well-founded surface solution of Σ if and only if there exists a subset, $\{A_i | 1 \leq i \leq N\}$, of the atoms of the surface lattice of Σ, and corresponding Goodwin projections, $\{\gamma_{A_i} | 1 \leq i \leq N\}$, such that*

$$\Gamma_\Sigma = \bigvee_{i=1}^{N} A_i \wedge \Gamma_{\gamma_{A_i}(\Sigma)}$$

where the $\Gamma_{\gamma_{A_i}(\Sigma)}$ are well-founded surface solutions.

This theorem guarantees a unique prime (where the primes are the surface atoms) factorization of lattice equational systems and their solutions. Notice that each Goodwin projection of a system Σ results in a system whose surface lattice is $\{\top, \bot\}$. We "know" how to solve these by theorem IV.2. Hence factoring followed by finding all of the (successful) elimination sequences produces all of the surface solutions. In particular, all four of the well-founded surface solutions of £9 can be produced by taking the Goodwin projections with respect to the surface atoms, *A* and *A,* solving the two resulting systems, "multiplying" the resulting solutions by *A* and *A* respectively, and "adding" the results. Since each projected system has two solutions, the overall system has four. Elsewhere [Benanav *et al.*\ we have shown the problem of solving general lattice equational systems to be *NP-hard* in the size of the system. We see now how this might arise: Using the algebraic results that we have cited can produce solutions at the cost of composing two potentially exponential processes, the projection by surface atoms and the finding of minimization and substitution sequences that actually produce a well-founded solution.

V. Conclusions

In the foregoing we have introduced a general model of truth maintenance couched in a lattice-theoretic framework. All of the truth maintenance systems familiar to us in the literature can be construed as solving systems of lattice equations. Indeed, those systems can be properly embedded in our lattice-theoretic formalism. We introduced the fundamental transformations of substitution and minimization and showed how they could be used to produce solutions of even lattice equational systems. We have cited a number of theoretical results about the algebraic structure of truth maintenance systems and interpreted these results in terms of concrete examples. We have used these examples to illustrate how a given formal algebraic result either reveals some intrinsic difficulty, or how it can be used to computational advantage. Finally, we sketched how our separation results can be used to generate all of the surface solutions of an arbitrary lattice equational system. The principal technical contributions of those aspects of our work on lattice-theoretic truth maintenance that we have presented in this paper are:

- The formalization of truth maintenance in a way that properly includes nonmonotonic justifications and assumption-based justifications
- The presentation of a point of view from which one can algebraically analyze the structure of truth maintenance problems and the construction of solutions to those problems
- The motivation of the algebraic results with computational and phenomenological interpretations

Though not the topic of this paper, it is also from this same lattice-theoretic point of view that we have carried out the analysis of the abstract and computational complexity of truth maintenance.

References

[Aho and Ullman, 1977] Alfred V. Aho and Jeffrey D. Ullman. *Principles of Compiler Design.* Addison-Wesley, Reading, Massachusetts, 1977.

[Aho *et al,* 1974] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, Massachusetts, 1974.

[Balbes and Dwinger, 1974] Raymond Balbes and Philip Dwinger. *Distributive Lattices.* University of Missouri Press, Columbia, Missouri, 1974.

[Benanav *et al*., 1986] Dan Benanav, Allen L. Brown, Jr., and Dale E. Gaucas. Reason maintenance from a lattice-theoretic point of view. In Lee S. Baumann, editor, *Proceedings of the Expert Systems Workshop,* pages 83-87, Defense Advanced Research Projects Agency, Science Applications International Corporation, Pacific Grove, California, April 1986.

[Benanav *et* al.] Dan Benanav, Allen L. Brown, Jr., and Dale E. Gaucas. *A Lattice-Theoretic Framework for Reason Maintenance.* Technical Report, General Electric Corporate Research and Development Center, Schenectady, New York, forthcoming.

[Birkhoff, 1967] Garrett Birkhoff. *Lattice Theory.* Volume 25 of *American Mathematical Society Colloquium Publications,* American Mathematical Society, Providence, Rhode Island, third edition, 1967.

[Birkhoff and MacLane, 1965] Garrett Birkhoff and Saunders MacLane. *A Survey of Modern Algebra.* MacMillan, New York, third edition, 1965.

[de Kleer, 1984] Johan de Kleer. Choices without backtracking. In *Proc. 4th Nat. Conf. on Artificial Intelligence,* pages 79-85, Austin, 1984.

[de Kleer, 1986a] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence,* 28:127-162, 1986.

[de Kleer, 1986b] Johan de Kleer. Extending the ATMS. *Artificial Intelligence,* 28:163-196, 1986.

[de Kleer, 1986c] Johan de Kleer. Problem solving with the ATMS. *Artificial Intelligence,* 28:197-224, 1986.

[Doyle, 1978] Jon Doyle. *Truth Maintenance Systems for Problem Solving.* Technical Report AI-TR-419, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, January 1978.

[Doyle, 1979a] Jon Doyle. A glimpse of truth maintenance. In *Proc. 6th Int. Joint Conf. on Artificial Intelligence,* pages 232-237, 1979

[Doyle, 1979b] Jon Doyle. A truth maintenance system. *Artificial Intelligence,* 12:231-272, 1979.

[Gantmacher, 1959] F R. Gantmacher. *Matrix Theory.* Volume 1, Chelsea, New York, 1959.

[Gaucas and Brown] Dale E. Gaucas and Allen L. Brown, Jr. *A Role for Assumption-based and Non-monotonic Justifications in Automating Strategic Threat Analysis.* Technical Report, General Electric Corporate Research and Development Center, Schenectady, New York, forthcoming.

[Goodwin, 1982] James W. Goodwin. *An Improved Algorithm for Non-Monotonic Dependency Update.* Technical Report LITH-MAT-R-82-23, Linkoping University, Linkoping, Sweden, August 1982.

[Goodwin, 1984] James W. Goodwin. Watson: a dependency directed inference system. In Proc. *of the Workshop on Non-Monotonic Reasoning,* pages 103-104, New Paltz, October 1984.

[Goodwin, 1985] James W. Goodwin. A process theory of non-monotonic inference. In Proc. *9th Int. Joint Conf. on Artificial Intelligence,* pages 185-187, Los Angeles, August 1985.

[Goodwin, 1987] James W. Goodwin. *A Theory and System for Non-Monotonic Reasoning.* PhD thesis, Linkoping University, Linkoping, Sweden, 1987.

[Hecht, 1977] Matthew S. Hecht. *Data Flow Analysis of Computer Programs.* American Elsevier, New York, 1977.

[McAllester] David A. McAllester. Personal communication.

[McDermott, 1983] Drew McDermott. Contexts and data dependencies: a synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 5(3):237-246, May 1983.

[Morris and Nado, 1986] Paul H. Morris and Robert A. Nado. Representing actions with an assumption-based truth maintenance system. In *Proc. 5th Nat. Conf on Artificial Intelligence,* pages 13-17, Philadelphia, 1986.

[Schaeffer, 1973] Marvin Schaeffer. *A Mathematical Theory of Global Program Optimization.* Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[Skornjakov, 1977] L.A. Skornjakov. *Elements of Lattice Theory.* Hindustan Publishing Corporation, Dehli, India, 1977. Translated from the Russian by V. Kumar.

[van der Waerden, 1953] B.L. van der Waerden. *Moden Algebra.* Ungar, New York, 1953.

[Waite and Goos, 1984] William M. Waite and Gerhard Goos. *Compiler Construction.* Springer-Verlag, New York, 1984.