A Calculus for Set-Based Program Development

Georg Struth

Institut für Informatik, Universität Augsburg
D-86135 Augsburg, Germany
struth@informatik.uni-augsburg.de

Abstract We propose an algebraic calculus for set-based program development. First, we reconstruct a fragment of set theory via atomic distributive lattices (ADL). Semantically, ADL extends boolean reasoning about sets by element-wise reasoning; it avoids presupposing a universal set. Operationally, ADL yields abstract, concise, elegant proofs from few elementary principles. Second, we develop a focused automated proof-search procedure for ADL with simple deduction and effective reduction and simplification rules. Proof-search is guided by rewriting techniques. The procedure decides several subclasses. Main application is the proof-support for formal methods like B or Z.

1 Introduction

Intuitive or naive set theory is both an official mathematical ontology and a univeral mathematical tool. In computer science, it is the basis of popular and successful formal methods like Z [17] and B [1]. Therefore, the integration of intuitive set-theoretic reasoning into efficient focused automated deduction systems is an important question. There are a few systems that implement axiomatic set-theory (e.g. [7,16,14,15]), but these are either interactive or designed for foundations rather than applications as automated formal methods. Hines [10] proposes a resolution calculus for restricted reasoning with some set-theoretic operations, but the problems of characterizing the underlying fragment of set theory and proving completeness of his calculus are now open for more than a decade. So the apparent lack of answer to the above question indicates an interesting gap both in the field of formal methods and in automated deduction.

We propose a calculus to close this gap. Besides, it solves the problems in [10]. Our first main contribution is a core calculus for set-theory as used in formal methods. The second one is the integration of this calculus into a focused resolution-based automated deduction system.

In opposition to the usual logical approaches, our core calculus is algebraic. It is the calculus of atomic distributive lattices (ADL). Appropriateness of ADL for reasoning with sets follows from the representation theorem for this class. Accordingly, every atomic distributive lattice can be isomorphically embedded into a field of sets; the zero of the lattice represented by the empty set, join and meet by union and intersection, the lattice order by set inclusion. But reasoning with ADL differs from boolean reasoning about sets: The ontological committment to a universal set is avoided, since there need not be a maximal element. Set-difference can nevertheless be expressed, since ADL has unique sectional complements. More important, ADL supports element-wise reasoning: elements of sets are in one-to-one correspondence with singleton sets, which represent atoms. Using techniques from the representation theorem we also show that atomicity of the lattice expresses precisely extensionality of the set theory. Our approach shares the usual benefits of other algebraic calculi for reasoning about programs (c.f. [6,11,13]): Economy of axioms, support of abstract, concise, elegant calculations from few elementary principle and relation to standard algebraic decision procedures. It is therefore particularly suited for automation.

We also propose a focused ordered resolution calculus (ADC) for ADL. Focusing means integrating mathematical and procedural knowledge, here via specific inference rules, rewriting techniques, ordering constraints and simplification techniques. The inference rules are specific ordered chaining rules (c.f. [3]) for ADL that extend a Knuth-Bendix completion procedure for distributive lattices [18]. They are restricted to manipulations with maximal terms in maximal atoms. Focusing seems indispensible for structures like ADL. Axiomatic reasoning would lead to an explosion of the search space. Our term-oriented ordering constraints, for instance, restrict the prolificy of theory unifications that would otherwise arise for joins and meets. We develop ADC with the derivation method from [19]. Its main idea is to internalize axioms into inference rules by establishing a separation property on axioms in refutations by ordered resolution and then internalizing the axioms into derived inference rules by inspecting their proof-patterns with non-theory clauses. Completenss of the resulting focused calculus then follows from faithfulness of the construction. Fortunately, since the derivation method is modular with respect to extensions, the development of ADC need not be done from scratch: we can base it on a focused calculus for distributive lattices [20]. By carefully chosing an appropriate representation of ADL, all inference rules of ADC are restrictions of those for distributive lattices. Using a variant of extensionality for eliminating all negative literals, the most prolific rules of the previous calculus even disappear. Moreover, the ADL-axioms dealing with atoms are entirely casted into simplification and reduction rules. Part of the efficiency of ADC is also due to the fact that atoms are only lazily introduced instead of boiling down the whole structure. Intuitively, new atoms are only used to witness that two sets are different. This feature also causes the specialization of ADC to decision procedures for certain subclasses, in particular finite structures.

Forgetting the ordering constraints, the inference rules of ADC are essentially those of [10]. We can also transfer the two main simplification techniques of [10] to the ordered resolution framework and to ADL. However, the ordering constraints give restrictions on certain inferences on variables for free that would otherwise be difficult to verify; the simplification rules of ADL can be justified using the standard generic notion of redundancy [2].

Here, we can only outline the main ideas of our approach. Formal details can be found in [21].

2 Lattices

A *lattice* [4,9] is a partially ordered set $(L, \leq)$ closed under least upper bounds or joins (denoted by $\sqcup$) and under greatest lower bound or meets (denoted by $\sqcap$) for all pairs of elements. Formally, for all $a, b, c \in L$,

$$a \leq c \wedge b \leq c \Leftrightarrow a \sqcup b \leq c, \tag{1}$$

$$c \leq a \wedge c \leq b \Leftrightarrow c \leq a \sqcap b. \tag{2}$$

The *dual* of a statement about lattices is obtained by interchanging joins and meets and converting the ordering. Thus (1) and (2) are dual statements. L is *distributive*, if

$$a \sqcap (b \sqcup c) \leq (a \sqcap b) \sqcup (a \sqcap c)$$

holds for all $a, b, c \in L$ or its dual and therewith both.

We denote the minimal and the maximal elements with respect to $\leq$ of L, if they exist, by 0 and 1. A lattice with 0 and 1 is called *bounded.* Formally, for all $a \in L$,

$$0 \leq a. \tag{3}$$

The class of lattices is denoted by L, the class of distributive lattices by DL. If K is a class of lattices, then K_0 denotes the subclass that has a zero, K_1 the subclass that has a one and K_{01} the subclass that is bounded.

We consider lattices as orderings. Alternatively, the class can also be axiomatized equationally. The translation between the two classes is given by

$$a \leq b \Leftrightarrow a \sqcup b = b \Leftrightarrow a \sqcap b = a. \tag{4}$$

In the equational definition, joins and meets are associative, commutative, idempotent ($a \sqcap a = a = a \sqcup a$) and absorptive ($a \sqcup (a \sqcap b) = a = a \sqcap (a \sqcup b)$ operations. Experience shows that order-based reasoning with lattices is more natural than equational reasoning.

Let $L_1, L_2 \in \mathsf{L}$. A mapping $h : L_1 \to L_2$ *preserves joins*, if $h(a \sqcup b) = h(a) \sqcup h(b)$ for all $a, b \in L_1$. It *preserves meets*, if $h(a \sqcap b) = h(a) \sqcap h(b)$ for all $a, b \in L_1$. A *lattice-homomorphism* (or *homomorphism*) preserves joins and meets. We also require $h(0) = 0$ and $h(1) = 1$, if present. An injective lattice homomorphism is called a *(lattice-)embedding*, a surjective lattice embedding a *(lattice-)isomorphism.* Every join or meet preserving mapping is *monotone*, that is $a \leq b$ implies $h(a) \leq h(b)$ for all $a, b \in L_1$.

Example 1.

(i) A family of subsets of some set is called *ring of sets*, if it closed under (set-theoretic) union and intersection. Every ring of sets is a distributive lattice. A finite lattice is distributive iff it is isomorphic to a ring of sets.

(ii) Every chain (for example the chain of natural numbers) is a distributive lattice.

3 Complements

Let $L \in \mathsf{L}_{01}$. A *complement* of an element $a \in L$ is an element $b \in L$ such that $a \sqcup b = 1$ and $a \sqcap b = 0$. L is *complemented*, if every element has a complement. A *boolean lattice* is a complemented distributive lattice. The class of boolean lattices is denoted by BL.

Our main interest are lattices with a weaker notion of complementation. Let $L \in \mathsf{L}_0$ and consider the sublattice $L|a = \{b \in L : b \leq a\}$. L is *sectionally complemented*, if $L|a$ is complemented for every $a \in L$.

In DL, sectional complements and complements are uniquely determined (if they exist). In DL_0, we write $b - a$ for the sectional complement of a in $L|(a \sqcup b)$. In DL_{01}, we write a' for the complement of a. Every complemented distributive lattice is sectionally complemented with $b - a = a' \sqcap (a \sqcup b)$ and $a' = 1 - a$. Every sectionally complemented distributive lattice with 1 is boolean. In Section 5 we will see that sectional complementation is very natural for atomic distributive lattices. Also in the context of intuitive set theory, sectional complements are natural concepts. There, the empty set is the zero, but we would like to avoid assuming the existence of a one, that is a universal set.

Example 2.

(i) In a ring of sets, $s_1 - s_2$ denotes set-difference, that is $s_1 - s_2$ is the set off all elements of s_1 that are not elements of s_2.
(ii) A ring of set is called *field of sets*, if it closed under set-difference. Every field of sets is a sectionally complemented distributive lattice.

We now present some laws for computing with sectional complements. First, we introduce some identities for simplifying expressions with sectional complements. Then we generalize some standard laws of complements to sectional complements: de Morgan's laws, monotonicity laws and shunting laws. Some of them can be found in the literature [9,5]. Most of the laws are based on the following property of sectional complements, which is immediate from the definition.

$$a = b - c \Leftrightarrow a \sqcup c = b \sqcup c \wedge a \sqcap c = 0.$$

– Simplification of sectional complement expressions.

$$(a - b) \sqcup b = a \sqcup b, \tag{5}$$
$$(a - b) \sqcap b = 0, \tag{6}$$
$$a \sqcap (a - b) = a - b, \tag{7}$$
$$a \sqcup (a - b) = a, \tag{8}$$
$$a - a = 0, \tag{9}$$
$$a - (b - c) = (a - b) \sqcup (a \sqcap c), \tag{10}$$
$$a - (a - b) = a \sqcap b. \tag{11}$$

– Generalized de Morgan's laws.

$$a - (b \sqcap c) = (a - b) \sqcup (a - c), \tag{12}$$
$$a - (b \sqcup c) = (a - b) \sqcap (a - c), \tag{13}$$
$$(a \sqcap b) - c = (a - c) \sqcap (b - c), \tag{14}$$
$$(a \sqcup b) - c = (a - c) \sqcup (b - c). \tag{15}$$

– Generalized monotonicity laws.

$$a \leq b \Rightarrow a - c \leq b - c, \tag{16}$$
$$a \leq b \Rightarrow c - b \leq c - a. \tag{17}$$

– Generalized shunting laws.

$$(a - b) \leq c \Leftrightarrow a \leq b \sqcup c, \tag{18}$$
$$a \sqcap (c - b) \leq d \Leftrightarrow a \sqcap c \leq b \sqcup d, \tag{19}$$
$$a \leq (c - b) \sqcup d \Leftrightarrow a \leq c \sqcup d \wedge a \sqcap b \leq d, \tag{20}$$
$$a \leq c - b \Leftrightarrow a \leq c \wedge a \sqcap b \leq 0. \tag{21}$$

(7) is very useful in the form $a - b \leq a$. Each of these laws can be proven in a few lines of lattice calculus. The standard laws for complements are recovered by setting $a' = 1 - a$.

The computations are interesting for two reasons. First, they are used in the normal form computations of our focused calculus in Section 9. There they allow us to completely eliminate sectional complements. Second, they support abstract algebraic reasoning with sets, for instance in set-based program development methods like Z and B.

4 Atoms

Intuitively, an atom of a lattice with zero is an element that lies immediately above (hence *covers*) the zero. Formally, let $L \in \mathsf{L}_0$. Then $\alpha \in L$ is an *atom* of L, if for all $b \in L$,

$$\alpha \not\leq 0, \tag{22}$$
$$b \leq \alpha \Rightarrow \alpha \leq b \vee b \leq 0. \tag{23}$$

Simple lattice calculus shows that (23) is equivalent with

$$\alpha \leq a \sqcup b \Leftrightarrow \alpha \leq a \vee \alpha \leq b, \tag{24}$$

if L is also distributive and sectionally complemented. $A(L)$ denotes the set of atoms of L.

(24) relates atoms with join-irreducible elements. An element a in a lattice L is *join-irreducible*, if for all $b, c \in L$, $a = b \sqcup c$ implies $a = b$ or $a = c$. All atoms

of a lattice are join-irreducible and all join-irreducible elements of a distributive sectionally complemented lattice with zero are atoms. Example 3 (ii) presents a finite distributive lattice with join-irreducible elements that are not atoms.

The following properties are helpful as rewrite rules for eliminating certain negative inequalities.

Lemma 1. *Let $L \in \mathsf{L}_0$. For all $\alpha, \beta \in A(L)$ and $a, b \in L$,*

$$\alpha \not\leq b \Leftrightarrow \alpha \sqcap b \leq 0, \tag{25}$$

$$\alpha \sqcap \beta \not\leq 0 \Leftrightarrow \alpha = \beta, \tag{26}$$

$$\alpha \sqcap a \leq b \Leftrightarrow \alpha \sqcap a \leq 0 \vee \alpha \leq b, \tag{27}$$

$$\alpha \sqcap a \not\leq b \Leftrightarrow \alpha \sqcap b \leq 0. \tag{28}$$

Example 3.

(i) In a field of sets, the atoms are precisely the singleton sets.

(ii) In the interval $[0, n]$ of natural numbers, all elements except 0 are join-irreducible. 1 is the only atom.

(iii) Consider the finite boolean lattice L_n generated by $a_1, \ldots, a_n$. The atoms of L_n are the elements $c_1 \sqcap \ldots \sqcap c_n$, where c_i is one of a_i and a_i'. By distributivity, every element $s \in L_n$ is equivalent to some $t \in L_n$ which is a join of meets of a_i and a_i'. If the join contains at least two elements, then t has at least two lower covers and is not join-irreducible. If the join has only one element, then $t = c_1 \sqcap \ldots \sqcap c_k$, where c_i is one of a_i and a_i' and $k \leq n$. If $k = n$, then t is an atom, hence join-irreducible. If $k < n$, then $t \sqcap a_{k+1}$ and $t \sqcap a'_{k+1}$ are lower covers of t. Thus t is not join-irreducible. Consequently, the join-irreducible elements of L_n are precisely the atoms.

5 Atomicity

A lattice $L \in \mathsf{L}_0$ is *atomic*, if for each non-zero $a \in L$ there is a nonempty $T \subseteq A(L)$ such that $a = \bigsqcup T$. For a class K of lattices, the subclass of atomic lattices is denoted by AK. We also define a mapping $\eta : L \to 2^{A(l)}$ that associates with each element $a \in L$ the set of atoms below it.

$$\eta(a) = \{\alpha \in A(L) : \alpha \leq a\}. \tag{29}$$

L is *η-stable*, iff $a = \bigsqcup \eta(a)$ holds for all $a \in L$.

It is easy to show that a lattice with zero is atomic iff it is η-stable. Using (24) is also easy to show that η is a homomorphism, if $L \in \mathsf{DL}$. If L is is non-distributive, η preserves zero and joins, but not necessarily meets.

Lemma 2. *Let L be a lattice with at least two elements. If η is injective, then $\eta(a) \neq \emptyset$ for all $a \in L$, $a \neq 0$.*

Proof. If L has at least two elements, then $A(L) \neq \emptyset$, since these elements have different images under η by injectivity. Moreover, $0 \in L$, since the definition of atoms presupposes a zero. Since $\eta(0) = \emptyset$, $\eta(a)$ must, for all $a \neq 0$ contain at least one atom. □

We now present an alternative characterization of atomicity. $L \in \mathsf{L}_0$ is *extensional*, if for all $a, b \in L$,

$$a \leq b \Leftrightarrow \forall \alpha \in A(L).(\alpha \leq a \Rightarrow \alpha \leq b). \tag{30}$$

By (25) and first-order logic, (30) is equivalent to

$$a \not\leq b \Leftrightarrow \exists \alpha \in A(L).(\alpha \leq a \wedge \alpha \sqcap b \leq 0). \tag{atomic}$$

Theorem 1. *A lattice with zero (and at least two elements) is atomic iff it is extensional.*

Proof. (Sketch) The key is to show the following claim: Lattice L is atomic iff

$$a \leq b \Leftrightarrow \eta(a) \subseteq \eta(b) \tag{31}$$

for all $a, b \in L$, which is equivalent to

$$a = b \Leftrightarrow \eta(a) = \eta(b), \tag{32}$$

since η preserves meets. To establish the claim, we first remember that atomicity is equivalent to η-stability. By η-stability,

$$\eta(a) \subseteq \eta(b) \Rightarrow \bigsqcup \eta(a) \leq \bigsqcup \eta(b) \Leftrightarrow a \leq b.$$

The other direction of (31) is just monotonicity of η.

We now show the converse direction of the claim, that (32) implies η-stability. First, we know that $\eta(a)$ and $\eta(b)$ are non-empty, by Lemma 2.

We verify that a is a least upper bound of $\eta(a)$. Since η is an embedding of L into some subsemilattice of $2^{A(L)}$ (η preserves meets), we can carry out the proof entirely on the set-side. Obviously, a is an upper bound of $\eta(a)$. To show that it is a least upper bound, assume, by reductio ad absurdum, another upper bound b of $\eta(a)$ such that $a \not\leq b$. Thus $\eta(a) \not\subseteq \eta(b)$ by (31) and by boolean reasoning $\eta(a) \cap (A(L) - \eta(b)) \neq \emptyset$. So there is some atom $\alpha \in \eta(a) \cap (A(L) - \eta(b))$. Consequently, $\alpha \in \eta(a)$ and $\alpha \in A(L) - \eta(b)$, hence on the one hand $\alpha \notin \eta(b)$. On the other hand, $\alpha \in \eta(a)$ implies $\alpha \in \eta(b)$, a contradiction. This proves our claim.

It now remains to show that (32) is equivalent to extensionality. But

$$\begin{aligned} a \leq b &\Leftrightarrow \eta(a) \subseteq \eta(b) \\ &\Leftrightarrow \forall \alpha \in A(L).(\alpha \leq a \Rightarrow \alpha \leq b) \end{aligned}$$

$\square$

Note that there is a finite $L \in \mathsf{DL}$ with $\eta(a) \neq \emptyset$ for all $a \neq 0$ that is not extensional and therefore not atomic.

We can restate Theorem 1 as follows. Define the relation $\sim$ for all $a, b \in L$ by

$$a \sim b \Leftrightarrow \forall \alpha \in A(L).(\alpha \leq a \Leftrightarrow \alpha \leq b). \tag{33}$$

$\sim$ is a congruence on sectionally complemented distributive lattices and $L \in \mathsf{L}$ is atomic iff $a = b \Leftrightarrow a \sim b$ holds for all $a, b \in L$.

Algebraically, (30) expresses an *extensionality principle*: two elements of an atomic lattice are equal, iff they are built from the same atoms. Similarly, (atomic) expresses a *separability principle*: two elements of an atomic lattice are different, iff they can be distinguished by an atom.

Operationally, (30) allows the transition between atom-free and atom-wise reasoning. Moreover, in Section 9, (30) and (atomic) are important for normal form computations for our focused calculi. (atomic) allows us to replace all negative inequalities by positive ones.

The following statement shows that in distributive lattices, atoms induce sectional complements.

Lemma 3. *Every atomic distributive lattice is sectionally complemented. For $L \in \mathsf{ADL}$ and $a, b \in L$, $a \leq b$, $\bigsqcup(\eta(b) - \eta(a))$ is the sectional complement of a in $L|b$.*

Proof. First, we show that $b \sqcap c = 0$. Since $\eta(b)$ and $\eta(a) - \eta(b)$ are disjoint, $(\bigsqcup \eta(b)) \sqcap (\bigsqcup(\eta(a) - \eta(b))) = 0$, using Lemma 1 (ii). Thus $b \sqcap c = 0$.

Now we show that $b \sqcup c = a$.

$$\begin{aligned}
b \sqcup c &= (\bigsqcup \eta(b)) \sqcup (\bigsqcup(\eta(a) - \eta(b))) \\
&= \bigsqcup(\eta(b) \cup (\eta(a) - \eta(b))) \\
&= \bigsqcup \eta(a) \\
&= a.
\end{aligned}$$

Thus c is a sectional complement of b with respect to a. By distributivity of the lattice, the complement is unique. $\square$

Consequently, we can use (24) instead of (23) in ADL.

Example 4.

(i) The set of all subsets of some set is an atomic boolean lattice.

(ii) In every field of sets, the singleton sets are precisely the atoms. Hence instead of the set-theoretic expression $a \in s$ we can write $\{a\} \subseteq s$ according to set theory and more abstractly $\alpha_a \leq s$ in AL. Existence of this atom is guaranteed by atomicity. Conversely, in a field of sets, we can write $a \in s \Rightarrow a \in t$ instead of $\alpha_a \leq s \Rightarrow \alpha_a \leq t$. Then (30) is equivalent to the standard axiom of extensionality of set theory,

$$a = b \Leftrightarrow \forall x.(x \in a \Leftrightarrow x \in b).$$

Conversely again, we can introduce the $\in$-relation as syntactic sugar for $L \in \mathsf{AL}$, defining $\alpha \in s \Leftrightarrow \alpha \leq s$ for all $\alpha \in A(L)$ and $a \in L$.

(iii) In Example 2 (i) we have stated, that in a field of sets, $s_1 - s_2$ denotes the set of all elements of s_1 that are not in s_2. This can be easily verified in ADL. First, we replace every statement of the form $a \in s$ by $\alpha_a \leq s$, using atomicity. Then, it remains to show that $\alpha \leq s_1 - s_2$ iff $\alpha \leq s_1 \wedge \alpha \not\leq s_2$. This follows immediately from (21) and (25).

6 Representation

The techniques of Theorem 1 usually serve for proving the well-known representation theorems for atomic lattices. The following facts are proven in [21] for $L \in \mathsf{L}$.

- If L is finite, then $\eta(a) \neq \emptyset$ for every non-zero $a \in L$.
- If $\eta(a) \neq \emptyset$ for every non-zero $a \in L$ and L is sectionally complemented, then L is atomic.
- L is distributive iff all sectional complements are unique.

It follows that every finite sectionally complemented lattice is atomic and distributive and therefore boolean. Moreover, every finite boolean lattice is atomic.

Theorem 2.

(i) Every atomic distributive lattice and every atomic boolean lattice L can be embedded into the field of sets $2^{A(L)}$.
(ii) Every finite atomic distributive lattice and every finite boolean lattice is isomorphic with the field of sets $2^{A(L)}$.

Our previous examples show that ADL has at least sets as models. The representation theorem shows that it has at most these models. Thus first-order reasoning about fields of sets is precisely first-order reasoning about ADL. But this is more than boolean reasoning. It is stronger, since via atoms, we are able to reason element-wise and it is weaker, since we avoid the ontological commitment to a universal set.

Given the representation theorems and the standard translation between objects of of set-theory and those of ADL, we can prove all statements of Section 3 to Section 5 entirely at the set-side.

Finally, well-known size bounds for finite lattices follow immediately from the representation theorems. The free boolean lattice on n generators, for instance, has 2^n atoms (c.f Example 3 (iii)) and therefore 2^{2^n} elements.

7 Ordered Resolution and Redundancy

We now come to the second part of our abstract, the discussion of the focused calculus for ADL. Ordered resolution is not only one of its main ingredients, it is also used as a metaprocedure for its development. We first recall some basic facts about ordered resolution and redundancy elimination (c.f. [2]).

Let $T_\Sigma(X)$ be a set of terms with signature Σ and variables in X. A term is *ground*, if it contains no variables. An *atomic formula* is an expression $p(t_1, \dots, t_n)$, where p is an n-ary predicate symbol and $t_1, \dots, t_m \in T_\Sigma(X)$. A *literal* is an atomic formula ϕ (*positive literal*) or its negation $\neg\phi$ (*negative literal*). A *clause* is a finite multiset of literals. A *Horn clause* contains at most one positive literal. A *clause set* is a set of clauses. If Γ is a clause and ϕ a literal, we write Γ, ϕ instead of $\Gamma \cup \{\phi\}$.

We consider calculi constrained by syntactic orderings. A *term* and a *literal ordering* $\prec$ is a well-founded total ordering on the respective ground expressions. $\prec$ is lifted to non-ground expressions by stipulating $e_1 \prec e_2$ iff $e_1\sigma \prec e_2\sigma$ for all ground substitutions σ. A literal l is *maximal* with respect to a multiset Γ of literals, if $l \nprec l'$ for all $l' \in \Gamma$. It is *strictly maximal* with respect to Γ, if $l \npreceq l'$ for all $l' \in \Gamma$. The non-ground orderings are still well-founded, but need no longer be total.

Literal orderings are extended to clauses, measuring clauses as multisets of literals and comparing them via the multiset extension of the literal ordering. A literal is assigned greater weight when it is negative than when it is positive. See Section 10 for more details. A clause ordering inherits totality and well-foundedness from the literal ordering. Again, the non-ground extension need not be total. We usually denote all syntactic orderings by $\prec$.

Definition 1. *Let $\prec$ be a literal ordering. The* ordered resolution calculus OR *consists of the deduction inference rules*

$$\frac{\Gamma, \phi \qquad \Delta, \neg\psi}{\Gamma\sigma, \Delta\sigma}, \quad \text{(Res)} \qquad\qquad \frac{\Gamma, \phi, \psi}{\Gamma\sigma, \phi\sigma}. \quad \text{(Fact)}$$

- *In the* ordered resolution rule *(Res), σ is a most general unifier of ϕ and ψ, $\phi\sigma$ is strictly maximal with respect to $\Gamma\sigma$ and maximal with respect to $\Delta\sigma$.*
- *In the* ordered factoring rule *(Fact), σ is a most general unifier of ϕ and ψ and $\phi\sigma$ is strictly maximal with respect to the set of positive literals and maximal with respect to the set of negative literals in $\Gamma\sigma$.*

In all inference rules, *side formulas* are the parts of clauses denoted by capital Greek letters. Literals occurring explicitly in the premises are called *minor formulas*, those in the conclusion *principal formulas*.

Let S be a clause set and $\prec$ a clause ordering. A clause Γ is *$\prec$-redundant* or simply redundant in S, if it is a semantic consequence of instances from S which are all smaller than Γ with respect to $\prec$. A ground inference is *redundant* in S, if either the maximal premise is redundant or else its conclusion is a semantic consequence of instances from S which are all smaller than the maximal premise with respect to $\prec$. An inference is *redundant*, if all its ground instances are. Closing S under OR up to redundant inferences and eliminating redundant clauses on the fly transforms S in the limit into an *ordered resolution basis* (an *orb*).

As usually, an OR-proof is defined inductively as a finite tree whose nodes are labeled by clauses and whose edges are determined by OR-inferences. An OR-*refutation* from a clause set S is an OR-proof with all leaves in S and with the empty clause as root.

Proposition 1.

(i) Orbs of inconsistent clause sets contain the empty clause.
(ii) Fair OR*-implementations refute inconsistent clause sets in finite time.*
(iii) For every inconsistent clauses set containing an orb there is a refutation in which no OR*-inference has both premises from the orb.*

8 The Derivation Method

We now briefly sketch the main ideas of the derivation method for focused calculi. At the clause level, our intension is the internalization of axioms into inference rules. At the term level, we would like to integrate theory-specific simplification rules, rewriting techniques and decision procedures. The method has a syntactic and a semantic side.

At the syntactic side, consider a set T of *theory clauses* and a set S of *non-theory* clauses that is disjoint from T such that $S \cup T$ is inconsistent. We intend to internalize T into a set of derived inference rules in refutations. The (ground) chaining rule

$$\frac{\Gamma, a \leq b \qquad \Delta, b \leq c}{\Gamma, \Delta, a \leq c}, \tag{34}$$

for instance, internalizes the instance of the transitivity law $a \not\leq b, b \not\leq c, a \leq c$ in the resolution proof

$$\cfrac{\cfrac{\Gamma, a \leq b \quad a \not\leq b, b \not\leq c, a \leq c}{\Gamma, b \not\leq c, a \leq c} \qquad \Delta, b \leq c}{\Gamma, \Delta, a \leq c}$$

With appropriate ordering constraints, (34) is an extension of a critical pair computation of a Knuth-Bendix completion procedure for transitive relations to the clausal level (c.f. [19]).

In general, this internalization is possible, if there exists an OR-refutation of $S \cup T$ with the following properties.

- The refutation is T-*separable*: For every premise $\Gamma \in T$ with k literals, the following $k-1$ inferences do not have another premise from T.
- The refutation is T-*serial*: For every premise $\Gamma \in T$ with k literals, the following $k-1$ inferences have one minor formula that is an instance of a literal of Γ.

Intuitively, T-separability guarantees that there is enough distance between premises from T to partition a refutation into subproofs that consume all but one literal from each premise from T separately. T-seriality guarantees that this consumption is not interrupted by inferences with minor formulas from non-theory clauses.

In our application, theory clauses have at most three literals. Thus only subproofs of size at most 2 must be inspected with respect to separability and seriality and, by Proposition 1 (iii), only subproofs of size at least 2, if T is an orb. This suggests the following three-step scenario. Let T be an input theory.

1. Construct an orb of T.
2. Establish separability and seriality for the ground case; extract inference rules.
3. Lift the inference rules to the non-ground case.

The derivation method is modular. When a set T of axioms is added to an orb B and the changes of the syntactic ordering do not affect B, then only inferences between B and T must be inspected for extending B. Also separability and seriality need only be checked with respect to T. In particular, if B is a subset of the orb of B and T and the second step succeeds, then the inference rules for B will at most be restricted in the extension.

At the semantic side, we use three ways to integrate declarative and procedural background knowledge. The first way is the selection of an appropriate theory specification. Our axioms for sectional complements and atoms in Section 3 to Section 5 are equivalences that act as clausal simplification rules. The second way is the extension of related procedures. In Section 11 we will extend a chaining calculus DC for DL [20]. The third way is the adaption of syntactic orderings $\prec$. We will reuse that of DC.

9 Reduction of Lattice Inequalities

Let $L = \{\sqcup, \sqcap, -, \alpha, 0\}$ be a signature for ADL. In particular, the unary function α denotes that some element is an atom. For the sake of simplicity, we flatten terms, that is we consider join and meets as operation symbols of polyadic arity. We also consider terms modulo associativity and commutativity (AC). Let Σ be a signature of free functions disjoint from L. As usually, we identify terms with trees. A term $t \in T_{\Sigma \cup L}(X)$ is *pure*, if for all subterms t' of t, if the root of t' has a label from Σ, then $t' \in T_{\Sigma}(x)$. A literal, clause or clause set is *pure*, if all terms that occur in it are pure. A *lattice term* is a pure term whose root is labeled by a lattice operation symbol. A term is *elementary*, if it is pure and the label of its root is neither $\sqcup$, $\sqcap$ or $-$.

An inequality $s \leq t$ is *reduced* if s is a (polyadic) join and t a (polyadic) meet of elementary terms. We write

$$s_1 \dots s_m \leq t_1 \dots t_m$$

instead of $s_1 \sqcap \dots \sqcap s_m \leq t_1 \sqcup \dots \sqcup t_m$. A clause or clause set is *reduced* if all the inequalities it contains are reduced. A formula is in *reduced clause normal form* (in $RCNF$), if it is in clause normal form (CNF) and every clause is reduced. A clause set is in *positive reduced clause normal form* (in $RCNF^+$), if it is in $RCNF$ and all literals are positive.

We now present an equivalence transformation ν from CNF to $RCNF^+$ for ADL. Since the transformation has more than 20 rules, we only discuss its main features. Details can be found in [21]. The rules of ν purify terms, eliminate negative lattice inequalities, split certain terms containing atoms, eliminate sectional complements, split with respect to joins and meets, discard literals and clauses and simplify lattice terms.

1. Purification. If in clause Γ, l the literal l contains an impure term t, then its subterm t' whose root is labeled by L is replaced by a fresh variable x in t and the clause Γ, l is replaced by $\Gamma, x \not\leq u, u \not\leq x, l[x]$. This renaming is justified, since in first-order logic
$$\phi(s) \Leftrightarrow \forall x.(x = s \Rightarrow \phi(x)),$$
$$\phi(s) \Leftrightarrow \exists x.(x = s \wedge \phi(x)).$$
2. Elimination of negative inequalities. If in $\Gamma, s \not\leq t$ the term s does not contain an atom symbol, then the clause is replaced by the clauses $\Gamma, \alpha(f(\overline{x})) \leq s$ and $\Gamma, \alpha(f(\overline{x})) \sqcap t \leq 0$, where f is a fresh Skolem function and $\overline{x}$ denotes the free variables in s and t. This is justified by (atomic). If s contains an atom term α, the replacement is based on (28) and (25); α is reused.
3. Atom-based splitting. A clause $\Gamma, \alpha \leq t_1 \sqcup t_2$ is replaced by $\Gamma, \alpha \leq t_1, \alpha \leq t_2$. This is justified by (24).
4. Elimination of sectional complements. These rules, for positive and negative literals, are justified by the generalized shunting rules (18)–(20). $\Gamma, s \not\leq r \sqcup (t_1 - t_2)$, for instance, is replaced by $\Gamma, s \not\leq r \sqcup t_1, s \sqcap t_2 \not\leq r$.
5. Join- and meet-based splitting. These rules, for positive and negative literals, are justified by (1), (2) and distributivity. $\Gamma, r \sqcap (s_1 \sqcup s_2) \not\leq t$, for instance, is replaced by $\Gamma, r \sqcap s_1 \not\leq t, r \sqcap s_2 \not\leq t$.
6. Clause simplification. These rules, for positive and negative literals, are justified by reflexivity of $\leq$, (3) and (22). A clause $\Gamma, 0 \leq s$, for instance, is discarded; a clause $\Gamma, \alpha \leq 0$ is replaced by Γ.
7. Term simplification. Subterms $s \sqcap s$, $s \sqcup s$, $s - 0$ and $s \sqcup 0$ are replaced by s; $0 - s$ and $s \sqcap 0$ by 0.

Note that also positive inequalities can be further reduced by (30). This is discussed in [21]. One can also use the generalized de Morgan laws and the simplification rules for sectional complements for further reduction.

All concrete rules of ν are so designed that they produce clauses. All axioms of ADL that deal with atoms are used in ν. In the elimination of negative inequalities, the introduction of new atoms is restricted as far as possible. The respective rule can be applied only finitely many times. It can be shown that ν terminates. By induction on the structure of lattice terms it is easy to see that some rule of ν is applicable, whenever a clause set is not in $RCNF^+$.

Finally, ν is not optimized. In several rules, lattice terms and clauses are copied. This could be avoided by renaming subterms, using an extension of the purification rule together with a Tseitin transformation at the clause level [23].

10 A Focused Calculus for DL

In this section we sketch the ordered chaining calculus DC for distributive lattices from [20] and further motivate the semantic side of the derivation method.

We first consider the orb D for DL. First, using the rules in ν for purification, join- and meet-based splitting and the appropriate clause and term simplifiction rules, we may assume that all clauses are reduced. Therefore, D can be restricted to axiomatize the reduced clausal theory of DL. Here, we may use D as a black-box. There are only two important points. First, D supposes normalization with respect to idempotence. This has been integrated into ν. Second, D uses the rule

$$x_1 \not\leq y_1 z, x_2 z \not\leq y_2, x_1 x_2 \leq y_1 y_2 \tag{cut}$$

for characterizing distributivity. Appropriately written, this is a lattice-theoretic variant of (Res). With appropriate ordering constraints, (cut) is a critical pair computation of a Knuth-Bendix completion procedure for DL (c.f. [18]). Extension to the clause level yields the focused ground inference rules

$$\frac{\Gamma, s_1 \leq a \qquad \Gamma', [s_2]a \leq t_2}{(\Gamma, \Gamma', s_1[s_2] \leq t_2)\nu} \tag{Cut}$$

$$\frac{\Gamma, s \leq t_1 a, s \leq t_1 t_2}{(\Gamma, [s]a \not\leq t_2, s \leq t_1 t_2)\nu} \tag{DF}$$

- In (Cut), the terms containing a are strictly maximal in the minor formulas. The minor formulas are strictly maximal wrt. the side formulas in the premises.
- In (DF), a is an elementary term, either t_1 is strictly maximal in the minor formulas or s is strictly maximal in the minor formulas and s is removed in the antecedent of the conclusion. The leftmost minor formula is strictly maximal with respect to the side formulas and the rightmost minor formula.

The brackets in the rules denote that an expression $[r]s \leq t$ can be read either as $s \leq t$ or as $rs \leq t$. Brackets in premises and conclusions are synchronized.

In addition, there is a dual rule to (DF). There are also several rules that resolve with negative literals. We do not mention these rules, since for ADL, we may assume $RCNF^+$, such that these rules are not applicable.

The construction of the syntactic ordering $\prec$ on terms and literals again follows our intention to model (cut) as a lattice-theoretic variant of (Res) and (Cut) as an extension of (34). To this end we compare joins and meets of elementary terms as multisets in the term ordering. Elementary terms are compared with respect to some standard term ordering. The literal ordering compares inequalities lexicographically with respect to the tuple (t_ν, p, s, t_μ). Here, t_ν denotes the maximal term in the inequality: t_μ the minimal term. p assigns to negative inequalities greater weight than to positive ones. s assigns greater weight to inequalities in which the maximal term occurs at the left-hand side. See [20] for a formal definition and a discussion. All these orderings are well-founded by construction; terms which are equal modulo associativity and commutativity are assigned the same measure. All these orderings are extended to the non-ground level and the clause level according to section 7.

11 A Focused Calculus for ADL

We now present an ordered chaining calculus for finite atomic distributive lattices. In the finite case, all non-theory clauses are ground, since existential and universal quantification can be replaced by joins and meets. The extension to the non-ground case is discussed in Section 12. We use the bracket notation introduced in Section 10.

We reuse the syntactic ordering for DC with one moderate local modification that has no impact on completeness of DC. In the term ordering, we also force atom terms to be smaller than all other terms, for example using a pair (a, g), such that $a = 1$ if the respective term denotes an atom and otherwise $a = 0$. g is the usual measure for a term.

Definition 2. *Let $\succ$ be the atom and clause ordering defined above. Let all clauses be in $RCNF^+$. Let ν be the transformation defined in Section 9. The* ordered chaining calculus for finite atomic distributive lattices ADC *consists of the deductive inference rules and the redundancy elimination rules of* OR[1] *and the focused inference rules (Cut), (DF) and the rule dual to (DF). The calculus is meant modulo associative and commutativity at the lattice level.*

Comparing ADC with DC, it turns out that addition of mathematical structure has lead to a simpler calculus. The elimination of negative inequalities by (atomic) is very beneficial, since the negative chaining rules are the most prolific rules of DC (c.f. [20] for a discussion). Moreover, the entire impact of sectional complements, atoms and atomicity could be integrated into the simplification rules of ν. This justifies the mathematical efforts from Section 3 and 5.

We now sketch the soundness and completeness proofs of ADC. Soundness is obvious relative to soundness of DC (c.f. [20]) and correctness of ν (c.f 9). Completeness is shown using the derivation method relative to previous results for DC.

Lemma 4. *Every ordered resolution inference with a clausal variant of (3), (22), (24) and (atomic) is redundant.*

Proof. Inspection with respect to $\prec$ show that every such inference yields a clause set that is smaller than and equivalent to the premise clause. □

[1] Section 7 only defines a semantic *notion* of redundancy. Every set of inference rules implementing this notion is admitted. Many such rules have already been encoded into ν'.

Lemma 4 immediately implies the following result, which yields the first step of our derivation method, the construction of an orb.

Proposition 2. *Let AD be the set D extended with clausal variants of (3), (22), (24) and (atomic). AD is an orb for the reduced clausal theory of* ADL.

We now proceed to the second step of the derivation method, the derivation of the focused inference rules.

Lemma 5. *For every set S in $RCNF^+$ such that $S \cup AD$ is inconsistent, there exists an AD-separable and AD-serial* OR*-refutation.*

Proof. Completeness of DC provides the focused inference rules for DL. Lemma 4 prohibits all interferences of (3), (22), (24) and (atomic) inside of the macros corresponding to these inference rules and among these axioms. This establishes AD-separability. In [18], T-seriality has been shown for arbitrary T (In fact, T-seriality is shown only relative to local violations of the ordering constraints, which are, however, all inside of macros and therefore disappear in the derived rules.). □

We are now prepared for our main theorem.

Theorem 3. *. Let all clauses be in $RCNF^+$. The ground ordered chaining calculus* ADC *is refutationally complete for finite atomic distributive lattices: For every ground reduced clause set that is inconsistent in the first-order theory of finite atomic distributive lattices, there exists a refutation in* ADC*.*

Proof. DC is refutationally complete for DL, but only the rules (Cut) and (DF) apply to clauses in $RCNF^+$. By Lemma 4 and Lemma 5, the axioms (3), (22), (24) and (atomic) de not interfere with these derived rules; they can be completely internalized into ν.

Conclusions of (Cut) must possibly be normalized by ν with respect to idempotence and negative inequalities must be eliminated by ν from conclusions of (DF). □

12 Discussion

In this section we briefly discuss further results that are formally proven in [21].

- We have developed an alternative to ADC in which, using (30), reduced terms can be further *atomized* to the form either $\alpha \leq s$ for s elementary or $\alpha s \leq 0$.
- The ordering constraints on (Cut) can be strengthened such that a is strictly maximal in the respective inequalities. This is shown by a proof transformation by induction on the structure of refutations. This restriction is very important for pruning the search space. We do not know a similar restriction for DC.
- ADC can be easily extended to a calculus for finite BL, thus improving a result from [20]. It can be immediately used as a calculus for finite sectionally complemented lattices (c.f Section 6).

– ADC (and its extension) decide the reduced clausal theories of finite ADL, finite BL, the universal theories of ADL and ABL and solve the uniform word problems for ADL and ABL. This follows from inspection of the rules in ν that eliminate negative inequalities. In fact, ν has been designed especially for this purpose. Since for the above problems, the input specification is ground, Skolemization does not introduce non-ground terms. Since Skolemization is needed only for negative inequalities with atom-free left-hand sides, the number of Skolemizations is bounded by the number of subterms that appear in the specification. Moreover, the ADC-inferences neither add any new symbols, nor do they shuffle atoms from left-hand to right-hand sides. Therefore only finitely many inferences lead to irredundant conclusions. The resulting orb contains the empty clause if and only if the initial clause set was inconsistent. It seems very interesting to extend these decision procedures to further classes, for instance by integrating more simplification rules into ADC. The elementary theory of distributive lattices, for instance, is undecidable [8], while the elementary theory of boolean lattices and atomic boolean lattices is decidable [22,12].
– We have extended ADC to the non-ground case, using standard lifting techniques. Now, some rules from ν are no longer simplifications. Idempotence of join and meets, for instance, must be taken out of ν, new deduction rules must be added to ADC. These are ordered factoring rules at the level of lattice terms. Moreover, also (Cut) inferences that cut out variables must be considered. Although such inferences can to a certain extent be restricted by the ordering constraints and avoided by further simplification and redundancy elimination techniques, it is open whether they can be completely avoided.
– Hines [10], in his related calculus, has presented two simplification techniques. One is based on *chainless sets*. In ADL, these are atoms that appear only at left-hand sides of inequalities. Since ν and the inference rules of ADL are designed such that atoms are never shuffled from left-hand sides of to right-hand sides, the atoms introduced by elimination of negative inequalities are chainless. It is easy to show that the chainless set technique integrates in the redundancy elimination framework. The other technique depends on an axiom that excludes a universal set. It allows the elimination of certain inequalities with variables that could otherwise lead to prolific (cut) inferences. We show that this technique also works in ADC.

13 Conclusion

We have proposed atomic distributive lattices as an algebraic core calculus for reasoning about sets in program development methods like B or Z. We have developed an axiomatization that supports the effective reduction and simplification of terms, inequalities and clauses and yields a modular extension of a focused calculus for distributive lattices. In particular, this extension simplifies the previous calculus. This nicely mirrors the fact that atomic lattices are mathematically simpler than non-atomic ones.

We envision the following further work. First the calculus should be implemented and integrated into a formal method. Second, more structure should be added, for instance, types for sets, pairs, comprehension, infinite sets, a choice function (c.f. [1]). Third, the transformation ν should be optimized, further simplification techniques should be developed. The theoretical results in this paper then open the way for practical reasoning about sets.

References

1. J.-R. Abrial. *The B-Book.* Cambridge University Press, 1996.
2. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. Logic and Computation*, 4(3):217–247, 1994.
3. L. Bachmair and H. Ganzinger. Rewrite techniques for transitive relations. In *Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 384–393. IEEE Computer Society Press, 1994.
4. G. Birkhoff. *Lattice Theory*, volume 25 of *Colloquium Publications.* American Mathematical Society, 1984. Reprint.
5. R. P. Dilworth. Lattices with unique complements. *Trans. Amer. Math. Soc.*, 57:123–154, 1945.
6. H. Doornbos, R. C. Backhouse, and J. van der Woude. A calculation approach to mathematical induction. *Theoretical Computer Science*, 179:103–135, 1997.
7. M. Gordon. Set theory, higher-order logic or both? In J. Grundy and J. Harrison, editors, *Theorem Proving in Higher-Order Logic: 9th International Conference*, volume 1125 of *LNCS*, pages 191–202. Springer-Verlag, 1996.
8. A. Grzegorczyk. Undecidability of some topological theories. *Fund. Math.*, 38:137–152, 1951.
9. H. Hermes. *Einführung in die Verbandstheorie.* Springer-Verlag, 1967.
10. L. Hines. Str$\dot{+}$ve$\subseteq$: The Str$\dot{+}$ve-based Subset Prover. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 193–206. Springer-Verlag, 1990.
11. C. A. R. Hoare and B. von Karger. Sequential calculus. *Information Processing Letters*, 53(3):123–130, 1995.
12. D. Kozen. Complexity of Boolean algebras. *Theoretical Computer Science*, 10:221–247, 1980.
13. D. Kozen. Kleene algebra with tests. *Transation on Programming Languages and Systems*, 19(3):427–443, 1997.
14. L. C. Paulson. Set theory for verification: I. From foundations to functions. *J. Automated Reasoning*, 11:353–389, 1993.
15. A. Quaife. Automated deduction in von-Neumann-Bernays-Gödel set theory. *J. Automated Deduction*, 8:91–147, 1993.
16. P. Rudnicki. An overwiev of the MIZAR project. Technical report, Department of Computing Science, University of Alberta, 1992.
17. J. M. Spivey. *Understanding Z.* Cambrigde University Press, 1988.
18. G. Struth. An algebra of resolution. In L. Bachmair, editor, *Rewriting Techniques and Applications, 11th International Conference*, volume 1833 of *LNCS*, pages 214–228. Springer-Verlag, 2000.
19. G. Struth. Deriving focused calculi for transitive relations. In A. Middeldorp, editor, *Rewriting Techniques and Applications, 12th International Conference*, volume 2051 of *LNCS*, pages 291–305. Springer-Verlag, 2001.

20. G. Struth. Deriving focused lattice calculi. In S. Tison, editor, *Rewriting Techniques and Applicaions, 13th International Conference*, volume 2378 of *LNCS*, pages 83–97. Springer-Verlag, 2002.
21. G. Struth. A calculus for set-based program development (manuscript). http://www.informatik.uni-augsburg.de/∼struth/papers/atomic.html, 2003.
22. A. Tarski. Arithmetical classes and types of Boolean algebras. *Bull. Am. Math. Soc.*, 55(64):1192, 1949.
23. G. S. Tseitin. On the complexity of derivations in propositional calculus. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning: Classical Papers on Computational Logic*, pages 466–483. Springer-Verlag, 1983. reprint.