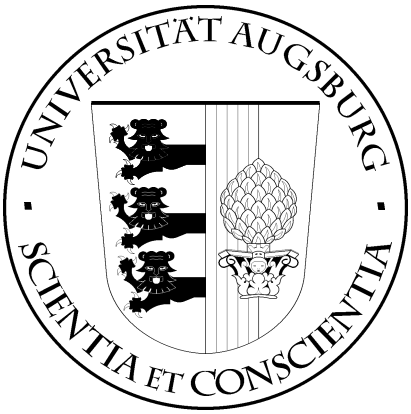
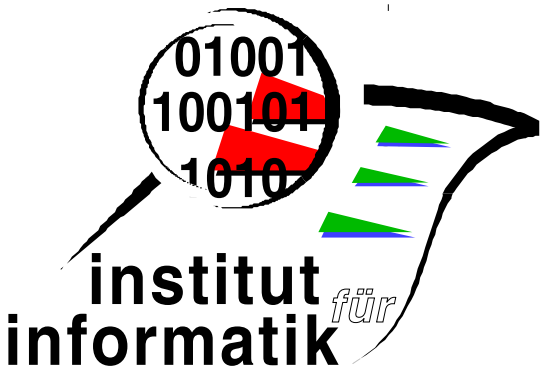
Universität Augsburg



A Calculus for Set-Based Program Development
Part II: Proof Search

Georg Struth

Report 2003-16 Oktober 2003



Institut für Informatik
D-86135 Augsburg

A Calculus for Set-Based Program Development Part II: Proof Search

Georg Struth

Institut für Informatik, Universität Augsburg
Universitätsstr. 14, D-86135 Augsburg, Germany
Tel:+49-821-598-3109, Fax:+49-821-598-2274,
struth@informatik.uni-augsburg.de

Abstract The first part of this work introduced a calculus for atomic distributive lattices. It is tailored for operational reasoning in naive or intuitive set theory and in set-based program development methods like Z or B. Here, we use this calculus for developing several focused automated proof-search procedures for atomic distributive and atomic boolean lattices. The procedures are based on ordered resolution; proof-search is guided by rewriting techniques. We derive simple deduction and powerful reduction and simplification rules, in particular decision procedures for several subclasses. Our results solve a longstanding open problem in automatic deduction and close an interesting gap in the proof support for formal methods.
Keywords Naive set theory, set-based program development, lattice theory, automated deduction, ordered resolution, term rewriting, decision procedures.

1 Introduction

In the first part of this work [22], we have developed a core calculus for operational reasoning in naive set theory and for set-based program development methods like B or Z. The calculus is based on atomic distributive lattices (ADL). The representation theorem for this class implies that the calculus satisfies some basic requirements for the intended applications. It models the empty set and the operations of set-union, set-intersection and set-difference. It avoids the ontological commitment to a universal set. It supports element-wise reasoning, since elements of sets are in one-to-one correspondence with atoms of lattices. Finally, atomicity of the lattice captures precisely extensionality of the set theory.

This algebraic turn to set theory is interesting on its own. It supports human reasoning about sets that is entirely calculatory and based on a few elementary principles. It provides abstraction, composition and extension facilities that are much less explicit in logic. Moreover, our algebraic core calculus opens the way to automated operational reasoning with sets. Such support of the needs of a working mathematicians and software engineer is one of the main desiderata of set-based formal methods and automated deduction. See [22] for further discussion. Among various attempts, an approach by Hines [11] has received considerable attention more than a decade ago. It is based on a series of investigations on

automated proof-search procedures for arithmetics [6,16,5,12] during the Eighties. Adapting these results to sets, Hines provides quite intuitive transformation, inference and simplification rules. He reports on interesting applications, for instance in analysis. However, while already completeness proofs of the arithmetics systems were rather involved (e.g. [5]), Hines was not able to characterize the underlying fragment of set theory or even prove completeness of his calculus. Even the progress in resolution-based theorem provers in the Nineties that results from adding syntactic orderings and using rewriting techniques in the work of Rusinowitch [17] and others did not significantly improve this situation. The reason is that the usual approaches to theory integration, to postulate a set of inferences rules and verify them post hoc in semantic completeness proofs, does not sufficiently scale from simple structures to more complex ones like sets. Thus the questions related to Hines calculus are still open; the lack of automated proof support for intuitive set theory and set-based software development persists as an interesting and challenging gap both in the field of formal methods and in automated deduction.

In a series of papers [20,21] based on [18], an alternative approach to theory integration has therefore be developed. In contrast to previous post hoc approaches, theory-specific inference rules can now be formally and systematically derived from natural specifications. Here, we use this derivation method to integrate our calculus for ADL into focused resolution-based proof search procedures. The procedures are intended to close the aforementioned gap. They also solve the problems with [11].

Focusing means integrating mathematical and procedural knowledge, here via specific inference rules, rewriting techniques, ordering constraints, simplification techniques and decision procedures. The inference rules of our procedures are specific ordered chaining rules (c.f. [3]) for ADL that extend a Knuth-Bendix completion procedure for distributive lattices [19]. They are restricted to manipulations with maximal terms in maximal literals. Focusing seems indispensable for structures of a complexity comparable to ADL. Axiomatic reasoning would lead to an explosion of the search space. Our term-oriented ordering constraints, for instance, control the proliferation by theory unifications that would otherwise arise for joins and meets. The main idea of the derivation method from [20] is to replace axioms by 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. Completeness of the resulting focused procedures then follows from faithfulness of the construction. Fortunately, since the derivation method is modular with respect to extensions, the development of our procedures need not be done from scratch: we can base it on an intermediate focused procedures for distributive lattices [21]. By carefully choosing an appropriate representation of ADL, all inference rules of our procedures are restrictions of those for distributive lattices. Using a variant of extensionality for eliminating all negative literals, the most prolific rules of the previous procedure even disappear. Moreover, the ADL-axioms dealing with atoms are entirely casted into simplification and reduction

rules. We provide two focused procedures for ADL. They procedures differ as follows. The first one introduces atoms only lazily instead of boiling down the whole structure. Intuitively, new atoms are only used to witness that two sets are different. The second one, in contrast, eagerly atomizes expressions. This further simplifies the inference rules, but it also causes larger initial clause sets and a loss of structure. The particular benefits of both alternatives should be evaluated in practice. In both cases, atoms are introduced carefully enough to make our procedures decide certain subclasses, in particular finite structures.

Forgetting the ordering constraints, the inference rules of one of our procedures encompass those of [11]. Conversely, ADL provides an algebraic semantics to Hines' procedure. We can also transfer the two main simplification techniques of [11] to the ordered resolution framework and to ADL. In particular, the ordering constraints give restrictions on certain inferences on variables for free that would otherwise be difficult to verify and the simplification rules of ADL can be justified in a rather straightforward way using the standard generic notion of redundancy [2]. This is another considerable improvement over [11].

Our procedures for ADL can easily be extended to procedures for atomic boolean lattices and finite boolean lattices. In particular, this last result improves a previous calculus [21]. Our calculi automatically specialize to decision procedures for certain subclasses, notably finite stuctures.

The remainder of this text is organized as follows. To make the paper self-contained with respect to references, we briefly recapitulate the relevant properties of our core calculus for ADL from [22] in Section 2 to Section 6. The remaining sections are devoted to the automation of ADL. Section 7 recalls the basic properties of ordered resolution and redundancy elimination; Section 8 sketches the derivation method for focused proof-search procedures. In Section 9, lattice inequalities are reduced using the ADL-axioms. Section 10 develops clausal axioms for ADL based on our results from Part I. Section 11 introduces specific syntactic orderings for ADL-terms in order to integrate rewriting techniques. Section 12 recalls the focused ordered chaining calculi for distributive lattices from [21]. These are the basis for our development of two focused ordered chaining calculi for finite ADL in Section 13. Soundness and Completeness of these procedures are proved in Section 14; their ordering constraints are further strengthened in Section 15. Decidability for some special cases is proved in Section 16. The calculi are lifted to the non-ground case in Section 17. Simplification and variable elimination techniques for the calculi are developed in Section 18. Section 19 contains a conclusion and an outlook.

2 Lattices

A *lattice* [4,10] is a poset $(L, \leq)$ 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. Homomorphisms with these additional properties are sometimes called *hemimorphisms*. 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 is 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 corresponds to 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 of 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 laws for computing with sectional complements. First, we introduce some identities for simplifying expressions with sectional complements. Then we generalize some standard laws for complements to sectional complements: de Morgan laws, monotonicity laws and shunting laws. Some of them can be found in the literature [10,8]. 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 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 usual laws for complements are recovered by setting $a' = 1 - a$.

The laws (5)–(21) 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 to

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

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 $a \in L$, $a \neq 0$.*

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.*

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 (c.f [22]).

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

$$a \sim b \Leftrightarrow \forall \alpha \in A(L).(\alpha \leq a \Leftrightarrow \alpha \leq b). \tag{31}$$

$\sim$ 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 $a, b \in L$. This congruence is interesting in its own right. For non-atomic lattices, it yields a notion of observational equivalence induced by measurements of lattice properties via atoms.

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 with our focused calculi. (atomic) allows us to replace all negative inequalities by positive ones.

According to he following statement, atoms of distributive lattices 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 $L|b$.*

Consequently, we can use (24) instead of (23) in ADL and we need no special axioms for sectional complements.

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 Some Meta-Theorems

The techniques of Theorem 1 usually serve for proving the well-known representation theorems for atomic lattices, which are variants of Stone's theorem (c.f [4]). The following facts are proven in [22] 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 with n generators, for instance, has 2^n atoms (c.f Example 3 (iii)) and therefore 2^{2^n} elements.

The following theorem of McKinsey is very interesting for restricting our calculi in certain special cases.

Theorem 3 ([15]). *Let* K *be a class of algebras closed under direct products and let the clause* $\phi_1, \ldots, \phi_m \longrightarrow \psi_1, \ldots, \psi_n$ *hold in* K*. Then* $\phi_1, \ldots, \phi_m \longrightarrow \psi_i$ *holds in* K *for some* $1 \leq i \leq n$*.*

In particular, the converse does also hold. For falsificational reasoning, the following variant is important.

Corollary 1. *Let* K *be a class of algebras closed under direct products. The clause* $\phi_1, \ldots, \phi_m \longrightarrow \psi_1, \ldots, \psi_n$ *does not hold in* K *iff* $\phi_1, \ldots, \phi_m \longrightarrow \psi_i$ *does not hold in* K *for all* $1 \leq i \leq n$*.*

Proposition 1. *The classes* ADL *and* ABL *are closed under direct products.*

This finishes the recapitulation of results from [22].

7 Ordered Resolution and Redundancy

We now turn to the discussion of the focused calculi 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. The main ideas are due to Rusinowitch [17]. They have been further variated, for instance, by [2]. Ordered resolution calculi are among the most powerful and successful automated deduction procedures. Particular benefits are their potential to decide many problem classes and to integrate theory-specific reasoning facilities.

Let $T_\Sigma(X)$ be a set of terms with signature Σ and variables in X, let P be a set of predicate symbols. A *clause* $\phi_1, \ldots, \phi_n$ is a finite multiset of literals. A *literal* is an expression $p(t_1, \ldots, t_m)$ or $\neg p(t_1, \ldots, t_m)$ where p is an m-ary predicate symbol from P and $t_1, \ldots, t_m$ are terms from $T_\Sigma(X)$. Literals of the first form are called *positive*, those of the second form *negative*. Clauses are denoted by capital Greek letters, positive literals by small Greek letters. We also write $\pm\phi$ to denote that ϕ is either positive or negative and $\Gamma, \pm\phi$ instead of $\Gamma \cup \{\pm\phi\}$. Logically, a clause denotes the universal closure of the disjunction of its literal. A clause set denotes the conjunction of the clauses it contains. We write $S \cup \Gamma$ instead of $S \cup \{\Gamma\}$. A *Horn clause* contains at most one positive literal. The *size* $|\Gamma|$ of a clause Γ is the number of its literals.

We consider calculi constrained by syntactic orderings. This may considerably narrow the search space. A *term* and a *literal ordering* $\prec$ is a well-founded total ordering on the respective set of 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 $\pm\phi$ is *maximal* with respect to a multiset Γ of literals, if $\phi \not\prec \psi$ for all $\psi \in \Gamma$. It is *strictly maximal* with respect to Γ, if $\phi \not\preceq \psi$ for all $\psi \in \Gamma$. 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. Literals are disambiguated by assigning to negative ones a greater weight than to positive ones. See section 11 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 orderings by $\prec$.

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

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

- *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 $\Gamma'\sigma$.*
- *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 into an *ordered resolution basis* (an *orb*) $\mathsf{orb}(S)$. The transformation orb need not terminate. We call an orb *inconsistent*, if it contains the empty clause and *consistent* otherwise.

As usual, OR-proofs are defined inductively as finite trees with nodes labeled by clauses and edges determined by derivability with the OR-inference rules. The *size* of a proof is the number of its inferences; hence its number of edges. A clause Γ is the *k-ancestor* of a clause Δ, if Γ is an ancestor of Δ and there is a path with k edges between Γ and Δ. A *refutation* from a clause set S in OR is a proof with all leaves labeled by elements of S and with the empty clause as root. We say that OR *refutes* S, if there is a refutation of S in OR. The set of OR-refutations from S is denoted by $\mathsf{ref}(S)$. We write $\mathsf{ref}_{\mathsf{C}}(S)$, if OR is augmented with a set C of domain-specific inference rules. Inferences with redundant conclusions need not be considered in refutations.

The following properties are interesting for our purposes.

Proposition 2.

(i) Orbs of inconsistent clause sets are inconsistent.
(ii) Fair OR-implementations refute inconsistent clause sets in finite time.
(iii) Conclusions of OR-inferences with both premises from an orb are redundant.
(iv) For every inconsistent clause set containing an orb there is a refutation without any OR-inference that has both premises from the orb.

8 The Derivation Method

Completeness of ordered resolution calculi with inference rules similar to the one above in combination with focused inference rules have been previously shown by two alternative approaches. The first one [13] extends and adapts the standard semantic-trees method for resolution. The second one [3] is a model construction that is more inspired by completeness proofs for logic programming and can be seen as an application of the technique of model-theoretic forcing [18]. Both methods were successfully applied to simple theories like equality or pre-congruence. But already extensions to structures like abelian semigroups [25] or cancellative abelian monoids [24] lead to difficulties worth a Ph.D.thesis. A significant problem is that both approaches are post hoc: Inference rules must be guessed and justified a posteriori in the respective semantic completeness proofs. For complex theories, this guessing seems rather hopeless and even showing completeness of a candidate calculus can be very difficult, as the problems with [11] and several errors in published completeness proofs show. As is well-known, post hoc approaches to the specification and analysis of (security) protocols or concurrent algorithms show similar problems. There, the solution consists in developing sophisticated protocols or efficient algorithms in a formal and systematic way from natural specifications. In a similar spirit, a derivation method for focused calculi based on ordered resolution has been developed in [20]. In this section, we recall and further formalize this method.

Consider a set T of *theory clauses* and a set S of *non-theory* clauses that is disjoint from T. Our intention is to replace T by a set of inference rules. We call T *focused*, if there is some set C of inference rules such that for all S, $\mathsf{ref}(S \cup T)$ is inconsistent iff $\mathsf{ref}_{\mathsf{C}}(S)$ is inconsistent. The following two properties capture focusing operationally in terms of proof transformations.

Let Π be a refutation from $S \cup T$.

Let Γ and Δ be two clauses in a proof Π whose least common ancestor is a k_Γ-ancestor of Γ and a k_Δ-ancestor of Δ. Then Γ and Δ are *k-separated* in Π, where $k = \max(k_\Gamma, k_\Delta)$. Π is *T-separable*, if every pair Γ and Δ in T is at least k-separated, where $k = \max(|\Gamma|, |\Delta|) - 1$.

By Proposition 2 (iv), for every inconsistent clause set containing an orb there is a refutation in which all elements of the orb are 2-separated.

Π is *T-serial*, if every instance Γ of a clause from T of size k that occurs in Π occurs in a subproof of Π of size $k-1$ in which one minor formula of each inference is an instance of a literal from Γ.

Proposition 3. *A clause set T is focused if for every clause set S such that $S \cap T = \emptyset$ and $S \cup T$ is inconsistent, there exists a T-separable and T-serial refutation. When T is finite, then so is C.*

Proof. Let Π be a T-separable T-serial refutation. By seriality, for each leaf of Π labeled by a clause $\Gamma \in T$ of size k there is a subproof of size $k-1$ such that at least one minor formula of each inference is an instance of a literal from Γ. By separability, this is also the case for at most one minor formula.

Therefore this subproof is completely determined by the premises from S, the ordering constraints of OR and the combination of resolution steps and factoring steps that occur in it. Moreover, the minor formulas of the premises from S are completely determined by the literals of Γ, up to unification.

With each such subproof we associate an inference rule that takes the premises of the subproof S as premises and the conclusion of the subproof as conclusion.

Now let T be finite. For each $\Gamma \in T$, there are finitely many combinations of using ordered resolution and ordered factoring in a $k-1$-size subproof. Moreover, there are up to unification finitely many minor formulas from S that can occur in the subproof. Then the number of inference rules is also finite. □

Intuitively, T-separability guarantees that the distance between members of T in a refutation is big enough to partition the refutation into subproofs that consume all but one literal from each instance of a member of T separately. Seriality guarantees that this consumption is not interrupted by inferences between non-theory clauses. The inference rules constructed this way correspond to theory-resolution rules in the unordered case. Here, as we will see, the situation is much more complex because of the ordering constraints.

So the main work is the construction of a focused set, that is the enforcement of separability and seriality. Fortunately, in the examples under consideration, clauses have at most size 3. Thus only subproofs of size 3 must be considered; it suffices to guarantee that theory clauses are at most 3-separated. This considerably restricts the combinatorics of inferences. Moreover, by Proposition 2 (iv) we have 2-separateness, if the set of theory clauses is an orb.

This suggests the following scenario that divides the development into three phases.

1. Construct the orb of an input theory.
2. Derive a focused ground calculus: Establish 3-separateness and seriality of the orb in the ground case. Extract inference rules from patterns arising in refutations with the focused theory.
3. Lift the ground calculus to the non-ground case (by standard techniques).

The extraction of inference rules from the interaction of theory and non-theory clauses in the second phase is very similar to the technique of *symmetrization* in computer algebra (c.f [7]). The first two phases of this scenario are further illustrated by the following simple example.

Example 5. Take the theory of a transitive relation $<$. A focused ordered chaining calculus for this class has already been developed in [20]. Here, we recapitulate the most important ideas. In this case, T has only one element, the transitivity axiom, which is written in clausal form as

$$x \not< y, y \not< z, x < z. \qquad \text{(trans)}$$

Consider the first phase of the scenario. In [20] we have proposed syntactic orderings on terms, literals and clauses for which (trans) is an orb for T. Thus,

by Proposition 2 (iv), T is 2-separated and we can dispense with inferences of the form

$$\frac{x_1 \not< x_2, x_2 \not< x_3, x_1 < x_3 \qquad x_1 \not< x_3, x_3 \not< x_4, x_1 < x_4}{x_1 \not< x_2, x_2 \not< x_3, x_3 \not< x_4, x_1 < x_4}.$$

This is desirable, since such an inference eagerly introduces fresh variables—here x_4—while intuitively, this should only happen by need, that is when forced by some non-theory clause. See [20] for a deeper discussion.

Consider now the second phase of the scenario. The focused inference system for transitive relations from [20] contains two *Negative Chaining rules*. We consider a ground variant

$$\frac{\Gamma, a < b \qquad \Gamma', a \not< c}{\Gamma, \Gamma', b \not< c}. \tag{32}$$

This rule is further subject to ordering constraints. (32) is a derived rule in OR; a macro built from the pattern arising in the two-step proof

$$\frac{\dfrac{\Gamma \boxed{a} < b \quad \boxed{a} \not< b, b \not< c, a < c}{\Gamma, b \not< c, \boxed{a} < c} \qquad \Gamma', \boxed{a} \not< c}{\Gamma, \Gamma', b \not< c}$$

We put maximal terms in boxes, where they occur in minor formulas. The other inference rules are derived in a similar way from the interaction of non-theory clauses with (trans) and OR. Completeness of the calculus means that every refutation can be completely covered by such patterns.

But is this really the case? A first answer is *no*. Precisely two obstacles may arise in deriving the negative chaining rule (32) from a proof pattern. First, it may happen that $\Gamma', a \not< c$ is a second instance of (trans), $a \not< c, c \not< d, a < d$, say. This is the case when T is not 3-separated. Second, it may happen that some negative literal in Γ is bigger than $a < c$. Then the second inference in (32) violates the ordering constraints and every two-step proof in a refutation using the first OR-inference corresponding to (32) must continue with an inference on a bigger literal. This is the case when the refutation is not serial. Thus, in order to extract (32) from a refutation pattern, from proof patterns the existence of 3-separated serial refutations remains to be shown. Only under this additional condition does the derivation of (32) go through.

In [20], seriality has been established in a generic way. It has been shown, that one may always *select* an inference on an instance of a literal of a theory clause instead of resolving with a clause that blocks seriality. Such selections locally forget the ordering constraints, but they globally preserve the structure of proofs and refutations in particular. They can therefore be completely internalized in the derived inference rules such that the selection is not visible in the derived focused calculus. 3-separability has been shown in a second step, using particular properties of the transitivity law. This establishes the inference rules of the ground ordered chaining calculus for transitive relations in [20].

In the third phase of the scenario, this calculus must be to the non-ground case. In [20], it has been shown that this can be done using standard techniques.

Apart from transitive relations, the three-phase scenario of the derivation method has also been applied to derive focused calculi for various lattices [21]. These calculi are very important for our purposes, since our calculi for ADL and ABL are strongly based on them. Remind that the former are subclasses of the latter.

This is possible, since the derivation method is intrinsically modular. This feature makes it incrementally dynamic with respect to theory extensions. When new axioms are added to an orb in such a way that a changes of the syntactic ordering do not affect the orb, then, in order to complete the orb, only inferences between elements of the orb and the new axioms must be considered, the previous orb need not be recompiled. Thus the first phase of the scenario is modular with respect to extensions. Also the second phase is modular: separability need only be checked for the new axioms and therefore the macro building can be accordingly restricted. This feature has already been demonstrated in [21] by axiomatizing a concept of *ideal* for distributive and boolean lattices. In general, modularity is an important feature of the derivation methods, which supports the modular development and incremental extension of theory hierarchies.

9 Reduction of Lattice Inequalities

We now present transformations for inequalities over terms in ABL and ADL. In particular all lattices in ADL are sectionally complemented by Lemma 3. We therefore assume a signature with sectional complements.

Let $L = \{\sqcup, \sqcap, -, \alpha, 0\}$ be a signature for lattice terms. In particular, the unary function α denotes that some element is an atom. We confuse this use of α as a function symbol with our previous use of α as a constant denoting an atom, wherever this may not lead to confusion. That is, we identify the function symbol α with the element α that denotes a term denoting an atom.

For the sake of simplicity, we flatten terms, that is we consider joins 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 usual, 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

$$s_1 \ldots s_m \leq t_1 \ldots t_n$$

instead of $s_1 \sqcap \ldots \sqcap s_m \leq t_1 \sqcup \ldots \sqcup t_n$. We also write $s_1 s_2 \leq t_1 t_2$ instead of $s_1 \sqcap s_2 \leq t_1 \sqcup t_2$. A clause or clause set is *reduced* if all the inequalities it contains

are reduced. A formula is in *weak reduced clause normal form* (in $RCNF_w$), if it is in clause normal form (CNF) and every clause it contains is reduced. A clause set is in *reduced clause normal form* (in $RCNF$), if it is in $RCNF_w$ and all inequalities it contains are positive.

An inequality is *atomized*, if it is positive, reduced and of either the form

$$\alpha(s)t \leq 0, \qquad \alpha(s) \leq t,$$

where $\alpha(s)$ denotes an atom. A clause or clause set is *atomized* if all the inequalities it contains are atomized. A formula is in *atomized clause normal form* (in $ACNF$), if it is in $RCNF$ and every clause is atomized.

We now present equivalence transformations from CNF to different reduced normal forms for DL, ADL and ABL expressions. The transformations are stated in terms of inference rules.

Definition 2. *Let S be a clause set, let Γ be a clause and ϕ be a (positive or negative) literal. We write $(S, (\Gamma, \phi))$ instead of $S \cup \{\Gamma \cup \{\phi\}\}$. Let $\circ$ denote $\sqcup$, $\sqcap$ or $-$. We write $[s\circ]t$ to denote alternatively the term t and the term $s \circ t$. A statement must be read uniformly for the short or the long variant.*

α denotes an atom, r, s, s_1, s_2, t, t_1 and t_2 denote pure terms.

The transformation ν^a_{ADL} on clause sets is given by the following inference rules[1]*.*

1. *The following rule purifies the clause set.*

$$(S, (\Gamma, \pm p[f(u)])) \longrightarrow (S, (\Gamma, x \not\leq u, u \not\leq x, \pm p[f(x)])). \qquad (\nu_1)$$

   Here, p denotes an arbitrary predicate, f is neither $\sqcup$, nor $\sqcap$, nor $-$, u is either 0 or labeled with $\sqcup$, $\sqcap$ or $-$ at the root and x is a fresh variable.
2. *The following rules eliminate negative lattice inequalities.*

$$(S, (\Gamma, u \not\leq t)) \longrightarrow (S, (\Gamma, \alpha(f(\overline{x})) \leq u), (\Gamma, \alpha(f(\overline{x})) \sqcap t \leq 0)), \qquad (\nu_2)$$
$$(S, (\Gamma, \alpha[\sqcap s] \not\leq t)) \longrightarrow (S, (\Gamma, \alpha \sqcap t \leq 0)). \qquad (\nu_3)$$

   In (ν_2), u is a pure term that does not contain an atom, f is a fresh Skolem function and $\overline{x}$ denotes the set of free variables in u and t,
3. *The following rules introduce atoms and split certain terms containing atoms.*

$$(S, (\Gamma, u \leq t)) \longrightarrow (S, (\Gamma, \alpha(x) \sqcap u \leq 0, \alpha(x) \leq t)), \qquad (\nu_4)$$
$$(S, (\Gamma, \alpha \sqcap s \leq t)) \longrightarrow (S, (\Gamma, \alpha \sqcap s \leq 0, \alpha \leq t)), \qquad (\nu_5)$$
$$(S, (\Gamma, \alpha \leq t_1 \sqcup t_2)) \longrightarrow (S, (\Gamma, \alpha \leq t_1, \alpha \leq t_2)). \qquad (\nu_6)$$

   In (ν_4), u is a pure term that does not contain an atom and x is a fresh variable. In (ν_5), $s \neq 0 \neq t$.

[1] The notation ν^x_{K} denotes that the transformation ν is intended for class K and yields reduced inequalities, when $x = r$ and atomized inequalities, when $x = a$.

4. *The following rules eliminate sectional complements.*

$$(S, (\Gamma, \pm[r\sqcap](s_1 - s_2) \leq t)) \longrightarrow (S, (\Gamma, \pm[r\sqcap]s_1 \leq s_2 \sqcup t)), \tag{ν_7}$$
$$(S, (\Gamma, s \leq [r\sqcup](t_1 - t_2))) \longrightarrow (S, (\Gamma, s \leq [r\sqcup]t_1), (\Gamma, s \sqcap t_2 \leq [r\sqcup]0)), \tag{ν_8}$$
$$(S, (\Gamma, s \not\leq [r\sqcup](t_1 - t_2))) \longrightarrow (S, (\Gamma, s \not\leq [r\sqcup]t_1, s \sqcap t_2 \not\leq [r\sqcup]0)), \tag{ν_9}$$

5. *The following rules split with respect to joins and meets.*

$$(S, (\Gamma, [r\sqcap](s_1 \sqcup s_2) \leq t)) \longrightarrow (S, (\Gamma, [r\sqcap]s_1 \leq t), (\Gamma, [r\sqcap]s_2 \leq t)), \tag{ν_{10}}$$
$$(S, (\Gamma, [r\sqcap](s_1 \sqcup s_2) \not\leq t)) \longrightarrow (S, (\Gamma, [r\sqcap]s_1 \not\leq t, [r\sqcap]s_2 \not\leq t)), \tag{ν_{11}}$$
$$(S, (\Gamma, s \leq [r\sqcup](t_1 \sqcap t_2))) \longrightarrow (S, (\Gamma, s \leq [r\sqcup]t_1), (\Gamma, s \leq [r\sqcup]t_2)), \tag{ν_{12}}$$
$$(S, (\Gamma, s \not\leq [r\sqcup](t_1 \sqcap t_2))) \longrightarrow (S, (\Gamma, s \not\leq [r\sqcup]t_1, s \not\leq [r\sqcup]t_2)), \tag{ν_{13}}$$

6. *The following rules simplify clauses.*

$$(S, (\Gamma, s \leq s)) \longrightarrow S \tag{ν_{14}}$$
$$(S, (\Gamma, s \not\leq s)) \longrightarrow (S, \Gamma) \tag{ν_{15}}$$
$$(S, (\Gamma, 0 \leq s)) \longrightarrow S \tag{ν_{16}}$$
$$(S, (\Gamma, 0 \not\leq s)) \longrightarrow (S, \Gamma) \tag{ν_{17}}$$
$$(S, (\Gamma, \alpha \leq 0)) \longrightarrow (S, \Gamma) \tag{ν_{18}}$$
$$(S, (\Gamma, \alpha \not\leq 0)) \longrightarrow S \tag{ν_{19}}$$

7. *The following rules simplify lattice terms.*

$$(S, (\Gamma, \pm p[s \sqcup 0])) \longrightarrow (S, (\Gamma, \pm p[s])) \tag{ν_{20}}$$
$$(S, (\Gamma, \pm p[s \sqcap 0])) \longrightarrow (S, (\Gamma, \pm p[0])) \tag{ν_{21}}$$
$$(S, (\Gamma, \pm p[s - 0])) \longrightarrow (S, (\Gamma, \pm p[s])) \tag{ν_{22}}$$
$$(S, (\Gamma, \pm p[0 - s])) \longrightarrow (S, (\Gamma, \pm p[0])) \tag{ν_{23}}$$
$$(S, (\Gamma, \pm p[s \sqcup s])) \longrightarrow (S, (\Gamma, \pm p[s])) \tag{ν_{24}}$$
$$(S, (\Gamma, \pm p[s \sqcap s])) \longrightarrow (S, (\Gamma, \pm p[s])) \tag{ν_{25}}$$

Here, p denotes an inequality.

Further simplification rules for sectional complements can be stated using the results on sectional complements from [22].

Before proving correctness of the transformation, we add a few remarks. First, note that there is some non-determinism in the specification. This supports further optimization. In general, the purification rule and the simplification rules for clauses and lattice terms should be eagerly applied.

Second, the rules (ν_2) and (ν_3) require special attention. They eliminate negative inequalities from clauses. (ν_2) does so by introducing a fresh Skolem term and therewith a fresh atom at left-hand sides of inequalities. This rule is only applied, when no atom occurs in the left-hand side of the inequality to be

eliminated. (ν_3) deals with the cases, when the left-hand side does contain an atom. Then, no new atom is introduced. Note that no rule will ever shift an atom from the left-hand side to the right-hand side of an inequality. These properties are not only important for controlling the search space, they are also crucial for the decision procedures in Section 16. The number of atom introductions can be further minimized, when (ν_3) is eagerly applied.

Third, note that the same argument applies to the rules (ν_4) and (ν_5) that simplify with respect to positive atoms. Again, the introduction of new atoms is restricted as far as possible. This is again important for decision procedures.

We now define variants of this transformation that are further specialized to different lattices and different normal forms.

Definition 3.

(i) *The transformation* ν^r_{ADL} *is the restriction of* ν^a_{ADL} *to the rules* (ν_1)–(ν_3) *and* (ν_7)–(ν_{25}).

(ii) *The transformation* ν^r_{DL} *is the restriction of* ν^a_{ADL} *to the rules* (ν_1), (ν_{10})–(ν_{15}), (ν_{20}), (ν_{21}) *and* (ν_{24}), (ν_{25}).

(iii) *The transformations* ν^x_{ABL} *and* ν^r_{BL} *are the extensions of* ν^x_{ADL} *and* ν^r_{DL} *by the rules*

$$(S, (\Gamma, s \leq 1)) \longrightarrow S, \qquad (S, (\Gamma, s \not\leq 1)) \longrightarrow (S, \Gamma),$$

where x *is one of* a *or* r. *Moreover, rules for complements can be used instead of* (ν_7)–(ν_9) *or* $s_1 = t_1 = 1$ *should be set in these rules for dealing with complements.*

Lemma 4.

(i) *All rules of* ν^a_{ADL} *except* (ν_2) *preserve equivalence with respect to* ADL; (ν_2) *preserves inconsistency.*

(ii) $\nu^a_{\mathsf{ADL}}(S)$ *is in ACNF for every clause set* S.

(iii) $\nu^a_{\mathsf{ADL}}(S)$ *terminates, if the clause set* S *is finite.*

Proof. (ad i) We show that each rule of ν^a_{ADL} except (ν_2) transforms the clause set S into an equivalent clause set with respect to ADL; whereas (ν_2) transforms an inconsistent clause set into an inconsistent one. We only mention the respective property that justifies the transformation and leave the associated *CNF*-transformation to the reader.

The transformation (ν_1) yields an equivalent clause set, using the first-order property

$$\phi(s) \Leftrightarrow \forall x.(x = s \Rightarrow \phi(x)).$$

for renaming positive literals and

$$\phi(s) \Leftrightarrow \exists x.(x = s \wedge \phi(x)).$$

for renaming negative literals. Note that

$$\neg\exists x.(x = s \wedge \phi(x)) \Leftrightarrow \forall x.(x = s \Rightarrow \neg\phi(x)).$$

The transformations (ν_2) and (ν_3) yield equivalent clause sets, using (atomic), (28) and (25), when the existential quantifier in (ν_2) is kept. Skolemization of this quantifier in (ν_2) is not an equivalence preserving transformation, but it preserves inconsistency.

The transformations (ν_4)–(ν_6) yield equivalent clause sets, using (30), (27) and (24).

The transformations (ν_7)–(ν_9) yield equivalent clause sets, using the shunting laws (19) and (21).

The transformations (ν_{10})–(ν_{13}) yield equivalent clause sets, using (1), (2) and distributivity.

The transformations (ν_{14})–(ν_{19}) yield equivalent clause sets. (ν_{14}) and (ν_{15}) use reflexivity of the partial ordering. (ν_{16}) and (ν_{17}) use the fact that 0 is a minimal element of the lattice. (ν_{18}) and (ν_{19}) are based on (22).

The transformations (ν_{20})–(ν_{25}) yield an equivalent clause set, applying the identities $a \sqcup 0 = a$, $a \sqcap 0 = 0$, $s - 0 = s$, $0 - s = 0$, $a \sqcup a = a$ and $a \sqcap a = a$ to terms, replacing equals by equals.

(ad ii) Assume that S is not in $ACNF$. Then some inequality is not atomized. By induction on the structure of lattice terms it is easy to see that some rule of ν is applicable.

(ad iii) Let $|\alpha| = |0| = 0$ for all atoms α and for zero. Let $|t| = 2$ for every constant or variable. Let terms from $T_\Sigma(X)$ be compared by an ordering that contains the subterm ordering. Let $|s \circ t| = 1 + |s| + |t|$, where $\circ$ stands for $\sqcup$, $\sqcap$ or $-$. Let $|s \leq t| = |s| + |t|$. We measure an inequality by the triple $\langle p, s, sc \rangle$, where $p = 1$ if the inequality is negative and $p = 0$ if it is positive, s denotes the size of the inequality and sc denotes the number of sectional complements. Literals are compared lexicographically with respect to this measure. Clauses are compared as multisets of literals and clause sets as multisets of clauses. All these orderings are well-founded. Inspection shows that (ν_1)–(ν_{25}) are decreasing with respect to the well-founded ordering. Hence ν^a_{ADC} terminates, whenever the initial clause set is finite. □

Analogous statements holds for the other transformations. The following lemma considers the transformation of ADL expressions to $RCNF$.

Lemma 5.

(i) All rules of ν^r_{ADL} except (ν_2) preserve equivalence with respect to ADL*; (ν_2) preserves inconsistency.*
(ii) $\nu^r_{\mathsf{ADL}}(S)$ is in RCNF for every clause set S.
(iii) $\nu^r_{\mathsf{ADL}}(S)$ terminates, if the clause set S is finite.

The following lemma considers the transformation of ABL to $RCNF$ and $ACNF$.

Lemma 6.

(i) All rules of ν^a_{ABL} and ν^r_{ABL} except (ν_2) preserve equivalence with respect to ABL*; (ν_2) preserves inconsistency.*

(ii) $\nu^a_{\mathsf{ABL}}(S)$ *is in ACNF for every clause set* S.
(iii) $\nu^r_{\mathsf{ABL}}(S)$ *is in RCNF for every clause set* S.
(iv) $\nu^r_{\mathsf{ABL}}(S)$ *and* ν^a_{ABL} *terminate, if the clause set* S *is finite.*

The following lemma considers the transformation of DL and BL to *RCNF*. Here, of course, there are no atoms. The transformations have already been used in [21].

Lemma 7.

(i) All rules of ν^r_{DL} *preserve equivalence with respect to* DL.
(ii) All rules of ν^r_{BL} *preserve equivalence with respect to* BL.
(iii) $\nu^r_{\mathsf{DL}}(S)$ *and* ν^r_{BL} *are in* $RCNF_w$ *for every clause set* S.
(iv) $\nu^r_{\mathsf{DL}}(S)$ *and* ν^r_{BL} *terminate, if the clause set* S *is finite.*

In practice, our transformations leave much space for improvement. In several rules, for instance (ν_{10})–(ν_{13}), lattice terms or even clauses are copied. This can be avoided by renaming of subterms at the level of first-order formulas, along the lines of (ν_1). By renaming, the normalization or atomization of inequalities leads to a linear increase of the size of the initial formula. Moreover the transformation is structure-preserving. This renaming can be integrated into a linear structure-preserving transformation to clause normal form. However, this is beyond the scope of this paper. Moreover, the generalized de Morgan laws from Section can be used to push all sectional complements into the lattice terms before applying the transformation.

In the sequel, we will denote all the above transformations simply by ν, whenever confusion is not possible.

10 Clausal Axioms for Atomic Distributive Lattices

In Section 2 we have defined lattices as posets. Here, we define lattices on quosets. In general, in a quoset, joins and meets are unique up to the congruence $\sim\, = (\leq \cap \geq)$. In case of reduced and atomized inequalities, $\sim$ reduces to equality modulo associativity and commutativity. Moreover operationally, the only role of antisymmetry is to split equations into inequalities. We therefore disregard antisymmetry in our considerations and consider terms modulo associativity and commutativity instead.

In DL, that is in the non-atomic case, we cannot apply the transformation rule (ν_2) and similar ones to eliminate negative inequalities. As we have seen, by ν^r_{DL}, a clause set can only be reduced to $RCNF_w$, but not to *RCNF* or *ACNF*.

Definition 4. *We define the following sets of axioms for* DL, ADL *and* ABL.

(i) For distributive lattices, let the set D consist of the Horn clauses

$$x \leq x, \tag{ref}$$
$$x \not\leq y, y \not\leq z, x \leq z, \tag{trans}$$
$$x \not\leq z, xy \leq z, \tag{ml}$$
$$x \not\leq y, x \leq yz, \tag{jr}$$
$$x_1 \not\leq y_1 z, z \not\leq y_2, x_1 \leq y_1 y_2, , \tag{jcut}$$
$$x_1 \not\leq z, x_2 z \not\leq y_2, x_1 x_2 \leq y_2, \tag{mcut}$$
$$x_1 \not\leq y_1 z, x_2 z \not\leq y_2, x_1 x_2 \leq y_1 y_2. \tag{cut}$$

(ii) For atomic distributive lattices, let the set AD consist of D together with

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

and clausal variants of (22), (24) and (atomic).
(iii) For atomic boolean lattices, let the set AB consist of AD together with

$$a \leq 1 \tag{33}$$

Note that by Lemma 16 (iii) of [22], there is no need for further axioms for characterizing atoms. See also the remark at the beginning of Section 4.

Proposition 4. *The following sets axiomatize the (weakly) reduced clausal theories of the following classes up to normalization with idempotence and modulo associativity and commutativity.*

(i) D for DL,
(ii) AD for ADL,
(iii) AB for ABL.

Proof. (ad i) This has been shown in [20].

(ad ii) Obvious from (i), since sectional complements do not appear in reduced clauses.

(ad iii) Obvious from (ii), since complements do not appear in reduced clauses. □

11 The Syntactic Orderings

The computation of an orb, its termination and the procedural behavior of our calculi crucially depend on the syntactic term, literal and clause orderings. Here, we use the orderings that have been developed for DL and BL in [21]. Only the term ordering must be slightly modified to appropriately handle atoms. This extension is however, conservative. See [21] for a detailed discussion and motivation of this ordering.

For the term ordering, let $\prec$ be the multiset extension of some total ordering on the set of constants. We assign minimal weight to 0 and 1, if present. We also

force atoms to be smaller than all other terms. This can be done for example using a pair (a, g), such that $a = 1$ if the respective term is an atom and otherwise $a = 0$. g is the usual measure for a term. $\prec$ is trivially well-founded, if the set is finite or denumerably infinite. We measure both components of a reduced inequality as a multiset. By construction, $\prec$ is well-founded and compatible with AC: terms which are equal modulo AC are assigned the same measure.

We now consider the literal ordering. Let $\mathbb{B}$ be the two-element boolean algebra with ordering $<_{\mathbb{B}}$. Let $m = G \times \mathbb{B} \times \mathbb{B} \times G$, where G denotes a multiset of constants. Let A be a set of literals occurring in some clause Γ. The ordering $\prec_1 \subseteq m \times m$ is the lexicographic combination of $\prec$ for the first and last component of m and $<_{\mathbb{B}}$ for the others. A ground *literal measure* (for clause Γ) is the mapping $\mu_\Gamma : A \to m$ defined by $\mu_\Gamma : \phi \mapsto (t_\nu(\phi), p(\phi), s(\phi), t_\mu(\phi))$ for each (ground) literal $\phi \in A$ occurring in Γ. Hereby $t_\nu(\phi)$ ($t_\mu(\phi)$) denotes the maximal (minimal) term with respect to $\prec$ in ϕ. $p(\phi) = 1$ ($p(\phi) = 0$), if ϕ is negative (positive). $s(\phi) = 1$ ($s(\phi) = 0$), if $\phi = s < t$ and $s \succeq t$ ($s \prec t$). The (ground) *literal ordering* $\prec_2 \subseteq A \times A$ is defined by $\phi \prec_2 \psi$ iff $\mu_\Gamma(\phi) \prec_1 \mu_\Gamma(\psi)$ for $\phi, \psi \in A$. Hence $\prec_2$ is embedded in $\prec_1$ via the literal measure. The ordering $\prec_1$ is total and well-founded by construction. Via the embedding, $\prec_2$ inherits these properties.

All these orderings are extended to the non-ground level and the clause level according to section 7. In particular the polarity p assigns greater weight to an negative occurrence of a term than to a positive one. In unambiguous situations we denote all orderings by $\prec$.

12 Chaining Calculus for Finite Distributive Lattices

In [21], chaining calculi for finite distributive and boolean lattices were developed with the derivation method, using the syntactic orderings $\prec$ from Section 11.

Theorem 4 ([21]). *D is an orb for the reduced clausal theory of distributive lattices. $D = \mathsf{orb}(D)$.*

In the calculus DC of [21], the effect of the orb D is completely internalized into the derived theory-specific rules. These rules are more focused than mere reasoning with the standard lattice axioms. In particular, this means that eager introduction of variables stemming from resolving the transitivity law, for instance, can be avoided.

We again introduce indexed brackets to abbreviate the presentation of the calculi. A pair of brackets [.] in a clause denotes alternatively the clause without the brackets and the clause, where the brackets together with their content have been deleted. For instance, the clause $\Gamma, [r]s \leq t$ denotes $\Gamma, rs \leq t$ or $\Gamma, s \leq t$. In inference rules, brackets with the same index are synchronized. For instance, the inference rule

$$\frac{\Gamma, [r]_i s \leq t}{\Gamma', [u]_i v \not\leq w}, \qquad \text{denotes} \qquad \frac{\Gamma, rs \leq t}{\Gamma', uv \not\leq w} \qquad \text{or} \qquad \frac{\Gamma, s \leq t}{\Gamma', v \not\leq w}.$$

We now recall the ordered chaining calculus DC for finite distributive lattices. Here, we use ν^r_{DL} for establishing weak reducedness. For the sake of simplicity, we just write ν.

Definition 5 (Distributive Lattice Chaining). *Let $\succ$ be the atom and clause ordering of section 11. Let all clauses be in $RCNF_w$. The* ordered chaining calculus for finite distributive lattices DC *consists of the deductive inference rules and the redundancy elimination rules of* OR[2] *and the following inference rules.*

$$\frac{\Gamma, rs \not\leq t}{\Gamma, r \not\leq t} \quad \text{(ML)} \qquad \frac{\Gamma, r \not\leq st}{\Gamma, r \not\leq s} \quad \text{(JR)}$$

Here the minor formula is maximal with respect to the negative literals in Γ and strictly maximal with respect to the positive literals in Γ.

$$\frac{\Gamma, s_1 \leq [t_1]_j x \qquad \Gamma', [s_2]_m x \leq t_2}{(\Gamma, \Gamma', s_1[s_2]_m \leq [t_1]_j t_2)\nu} \quad \text{(Cut+)}$$

Here the terms containing x are strictly maximal in the minor formulas. The minor formulas are strictly maximal with respect to the side formulas in their respective premises.

$$\frac{\Gamma, s_1 \leq [t_1]_j x \qquad \Gamma', s_1[s_2]_m \not\leq [t_1]_j t_2}{(\Gamma, \Gamma', [u]_m x \not\leq t_2)\nu} \quad \text{(Cut-)}$$

Here the terms containing s_1 are strictly maximal in the minor formulas. In the first premise, the minor formula is strictly maximal with respect to the side formulas. In the second premise, the minor formula is maximal with respect to the side formulas. Moreover, $[u]_m x \neq t_2 \mod AC$, $u = s_2$ or else $u = s_1$, if s_2 is absent in the minor formula.

$$\frac{\Gamma, [s_2]_m x \not\leq t_2 \qquad \Gamma', s_1[s_2]_m \leq [t_1]_j t_2}{(\Gamma, \Gamma', s_1 \not\leq [u]_j x)\nu} \quad \text{(Cut-)}$$

Here the terms containing t_1 are strictly maximal in the minor formulas. In the first premise, the minor formula is maximal with respect to the side formulas. In the second premise, the minor formula is strictly maximal with respect to the side formulas. Moreover, $s_1 \neq [u]_j x \mod AC$, $u = t_1$ or else $u = t_2$, if t_1 is absent in the minor formula.

$$\frac{\Gamma, s \leq [t_1]_j x, s \leq [t_1]_j t_2}{(\Gamma, [s]_m x \not\leq t_2, s \leq [t_1]_j t_2)\nu''} \quad \text{(DF)}$$

Here, x 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 can be set to 1 in the antecedent of the conclusion. The leftmost minor formula is strictly maximal with respect to the side formulas and the rightmost minor formula.

$$\frac{\Gamma, [s_1]_m x \leq t, [s_1]_m s_2 \leq t}{(\Gamma, s_2 \not\leq [t]_j x, [s_1]_m s_2 \leq t)\nu} \quad \text{(DF)}$$

[2] Section 7 only defines a semantic *notion* of redundancy. Every set of inference rules implementing this notion is admitted.

Here, x is a elementary term, either s_1 is strictly maximal in the minor formulas or t is strictly maximal in the minor formulas and t can be set to 1 in the conclusion. The leftmost minor formula is strictly maximal with respect to the side formulas and the rightmost minor formula.

The calculus is meant modulo AC at the lattice level.

(JR) and (ML) stand for *join right* and *meet left*, in analogy to the sequent calculus. (Cut+) and (Cut-) stand for *positive* and *negative cut*, (DF) for *distributivity factoring*. The two (Cut-) rules and the two (DF) rules are dual, if also the indices of brackets are exchanged.

For distributive lattices with zero or one, and for boolean lattices, we add the inference rules

$$\frac{\Gamma, 0 \not\leq a}{\Gamma}, \tag{Zero}$$

$$\frac{\Gamma, a \not\leq 1}{\Gamma}. \tag{One}$$

For detecting triviality, that is $0 = 1$ we add the inference rule

$$\frac{\Gamma, 1 \leq 0}{\Gamma}. \tag{Trivial}$$

We still call the resulting calculi DC.

Proposition 5. *Let all clauses be in $RCNF_w$. The ground ordered chaining calculus DC is sound for finite distributive (and boolean) lattices: For every OR-proof using the inference rules of DC there is a OR-proof from the same premises to the same conclusion that uses the axioms in D.*

Theorem 5. *Let all clauses be in $RCNF_w$. The ground ordered chaining calculus DC is refutationally complete for (the elementary theory of) finite distributive (and boolean) lattices: For every ground prereduced clause set that is inconsistent in the first-order theory of distributive lattices there exists a refutation in DC.*

13 Focused Calculi for Finite ADL

We now present two ordered chaining calculi 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 17. We use the bracket notation introduced in Section 12.

The first calculus uses ν^r_{ADL} with lazy introduction of atoms.

Definition 6. *Let $\succ$ be the atom and clause ordering defined above. Let all clauses be in RCNF. Let ν^r_{ADL} be the transformation defined in Section 9. The* ordered chaining calculus for finite atomic distributive lattices ADC_r *consists*

of the deductive inference rules and the redundancy elimination rules of OR[3] *and the focused inference rules (Cut+) and (DF), normalizing with* ν^r_{ADL}*. The calculus is meant modulo AC at the lattice level.*

Comparing ADC_r with DC shows that addition of mathematical structure has lead to a simpler calculus. The elimination of negative inequalities by (atomic) via (ν_2) is very beneficial, since the negative chaining rules are the most prolific rules of DC (c.f. [21] for a discussion). Moreover, the entire impact of sectional complements, atoms and atomicity could be integrated into the simplification rules of ν^r_{ADC}. This justifies the mathematical efforts in [22] for defining rules for sectional complements and particularly useful axioms for atoms and atomicity.

We now present a second calculus which is based on ν^a_{ADL} with eager introduction of atoms, whence atomized clauses.

Definition 7. *Let* $\succ$ *be the atom and clause ordering defined above. Let all clauses be in ACNF. Let* ν^a_{ADL} *be the transformation defined in Section 9. The* ordered chaining calculus for finite atomic distributive lattices ADC_a *consists of the deductive inference rules and the redundancy elimination rules of* OR[4] *and the following focused inference rules. (Cut+) is restricted to*

$$\frac{\Gamma, \alpha \leq x \qquad \Gamma', [s]x \leq 0}{(\Gamma, \Gamma', \alpha[s] \leq 0)\nu^a_{\mathsf{ADL}}} \qquad \frac{\Gamma, \alpha \leq \beta \qquad \Gamma', \beta \leq t}{\Gamma, \Gamma', \alpha \leq t}$$

(DF) is restricted to

$$\frac{\Gamma, \alpha s'_1 x \leq 0, \alpha s'_1 s_2 \leq 0}{(\Gamma, s_2 \not\leq x, \alpha s'_1 s_2 \leq 0)\nu^a_{\mathsf{ADL}}}$$

The constraints are also further strengthened. Neither s'_1 *nor* x *may contain an atom and* $\alpha s'_1$ *must be strictly maximal. There is the additional rule*

$$\frac{\Gamma, \alpha, \beta, s \leq 0}{(\Gamma\sigma, \alpha\sigma s\sigma \leq 0\sigma)\nu^a_{\mathsf{ADL}}}, \tag{MI}$$

where σ *is a most general unifier of* α *and* β*,* $\alpha\sigma$ *is maximal in the respective term and the* σ*-instance of the minor formula is strictly maximal in the left-hand rule and maximal in the right-hand rule.*

The calculus is meant modulo AC at the lattice level.

At the present stage of work it is difficult to compare ADC_r and ADC_a. ADC_r is more economic and algebraic. It avoids introducing new atoms as far as possible, and is therefore less element-wise. ADC_a is more reductive with respect to the term structure and applications of inference rules. It is however, due to the use of (ν_4), non-ground (although only syntactic unification is involved). We believe that the usefulness of the calculi should be best compared by practical experiments.

[3] 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^r_{ADL}.

[4] 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^a_{ADL}.

14 Soundness and Completeness

Soundness of ADC_r and ADC_a is straightforward.

Proposition 6. *The ground ordered chaining calculi* ADC_r *and* ADC_a *are sound for finite atomic distributive lattices.*

(i) Let all clauses be in RCNF. For every OR*-proof using the inference rules of* ADC_r *there is a* OR*-proof from the same premises to the same conclusion that uses the axioms in D.*

(ii) Let all clauses be in ACNF. For every OR*-proof using the inference rules of* ADC_a *there is a* OR*-proof from the same premises to the same conclusion that uses the axioms in D.*

Proof. Obviously, all inference rules of ADC_r and ADC_a are special cases of inference rules from DC. The rule (MI) is a special case of the non-ground calculus for DL. These inference rules are sound for the reduced clausal theory of distributive lattices [21]. Since every atomic distributive lattice is a distributive lattice and every clause set in *ACNF* or *RCNF* is also in $RCNF_w$, the inference rules of ADC_r and ADC_a are also sound for finite atomic distributive lattices. Moreover, by Lemma 4 and Lemma 5, all transformation rules in ν^r_{ADL} and ν^a_{ADL} are sound. □

In order to prove completeness, we now apply the derivation method to construct the focused calculi ADC_r and ADC_a. The derivation is modular with respect to the derivation of the calculus DC for finite distributive lattices in [21]. We construct an orb for ADL relative to that for DL and derive the inference rules of ADC_r and ADC_a relative to those of DC. But first, we prove a technical lemma.

Lemma 8. *Every ordered resolution inference with a clausal variant of (3), (22), (24) and (atomic) is redundant.*

Proof. Every such inference leads to a clause set that is smaller than and equivalent to the premise clause set. This is immediately evident for (3), (22) and (24). In case of (atomic), a clause negative $\Gamma, u \not\leq t$ is split into $\Gamma, \alpha \leq u$, and $\Gamma, \alpha t \leq 0$. If $u \succ t$, then both resolvents are smaller than the premise. If $u \prec t$, then we may assume that t is not an atom. Then the first conclusion is smaller than the premise. To make the second premise smaller, we replace $\Gamma, u \leq t$ by the equivalent expression $\Gamma, u \leq tt$ everywhere in the higher part of the proof. □

Lemma 8 immediately implies the following result, which yields the first step of the derivation of the inference rules, the construction of the orb.

Proposition 7. *AD is an orb for the theory of reduced clauses and atomized clauses of* ADL.

Proof. By Proposition 4 (i), D is an orb for the reduced clausal theory of DL. By Lemma 8, every ordered resolution inference between (22), (atomic) and members of D and among (22) and (atomic) is redundant. Therefore AD is an orb for the theory of reduced clauses and atomized clauses of ADL. □

We now proceed to the second step of the derivation method.

Lemma 9. *The set AD is focused.*

Proof. By Proposition 3 we must show the existence of a T-separable and T-serial refutation. The orb D is focused by Theorem 5. By Lemma 8, also all elements of AD are AD-separated. The proof of T-seriality is by induction on the size of clauses in a refutation. The argument is independent from the particular structure of the orb. It has already been given in [19]. The main idea is that the needed inference can be permuted up in the refutation tree, whereas the second step of the blocking inference can be permuted down. One then obtains a new refutation with the desired macro inference at the appropriate place. Due to factoring it may be the case that some subtrees of the proof tree must be copied. This construction is iterated on the proof tree. The proof immediately applies to the present case. Hence AD is focused. □

We are now prepared for our main theorems. The first theorem expresses completeness of ADC_r.

Theorem 6. . *Let all clauses be in RCNF. The ground ordered chaining calculus ADC_r is refutationally complete for (the elementary theory of) finite atomic distributive lattices: For every ground clause set in RCNF that is inconsistent in the first-order theory of finite atomic distributive lattices there exists a refutation in ADC_r.*

Proof. By Proposition 9, AD is focused. It consists of the focused set A for distributive lattices plus clauses corresponding to the equivalences (3), (22), and (atomic). There is no need to derive further inference rules for (22) and (atomic), since by Lemma 8, these expressions yield equivalence transformations on clauses that yield smaller clauses. They are therefore not deduction, but reduction or simplification rules that have been integrated into the transformation ν^r_{ADL}.

By this transformation, every clause set is transformed into a set of clauses in *RCNF*. Therefore the negative cut rules (Cut-), the meet left rules and join right rules (ML) and (JR) and the (Zero) rule of DC are no longer applicable, only the positive cut rule (Cut+) and the distributivity factoring rules (DF) of DC must be kept.

It remains to argue why deduction steps are intertwined with simplification steps from ν^r_{ADL}. This is the case, since in the conclusion of (Cut+), there can be multiple occurrences of an elementary term in a side of a reduced inequality. Moreover, in the conclusion of (DF), a negative inequality must be eliminated and the resulting positive inequalities must be further transformed. □

The second theorem expresses completeness of ADC_a.

Theorem 7. . *Let all clauses be in ACNF. The ground ordered chaining calculus ADC_a is refutationally complete for (the elementary theory of) finite atomic distributive lattices: For every ground clause set in ACNF that is inconsistent in the first-order theory of finite atomic distributive lattices there exists a refutation in ADC_a.*

Proof. Obviously, ADC_a is a further specialization of ADC_r, with the exception of the rules (JI) and (MI). We show how the (Cut+) rules and the (DF) rule of ADC_a arise from the special format of atomized inequalities.

(i) Consider the (Cut+) rule

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

First, there is no inference for $x = 0$. It then follows from the shape properties of atomized inequalities, that $s = \alpha$. Moreover, these properties imply that $[s_2]x \leq t_2$ specializes to either $[s]x \leq 0$ or $\beta \leq t$, where $x = \beta$ denotes an atom. The first restriction yields the left-hand inference rule, the second case the right-hand one.

$$\frac{\Gamma, \alpha \leq x \qquad \Gamma', [s]x \leq 0}{(\Gamma, \Gamma', \alpha[s] \leq 0)\nu} \qquad \frac{\Gamma, \alpha \leq \beta \qquad \Gamma', \beta \leq t}{\Gamma, \Gamma', \alpha \leq t}$$

The constraints are inherited from those of ADC_r.

(ii) Consider the first (DF) rule,

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

Similar arguments to (i) show that the left minor formula of the premise specializes to $\alpha \leq x$ and the right one to $\alpha \leq t$. The constraints require that α is greater than x. Hence x must be another atom, $x = \beta$, say. If $\alpha \neq \beta$, then $\alpha \leq x$ is false and the premise $\Gamma, x \leq t, \alpha \leq t$ reduces to $\Gamma, \alpha \leq t$. Hence (DF) is not applicable (without simplifying the premise, the conclusion would be $\Gamma, \beta t \leq 0, \alpha \leq t$, which is subsumed by the premise and therefore redundant).

If $\alpha = \beta$, then the premise becomes a tautology and can be discarded by ν.

Consequently, the first (DF) rule is not applicable to clause sets in *ACNF*.

(iii) Consider the second (DF) rule,

$$\frac{\Gamma, [s_1]x \leq t, [s_1]s_2 \leq t}{(\Gamma, s_2 \leq tx, [s_1]s_2 \leq t)\nu} \tag{DF}$$

If $t \neq 0$, then $s_1 x = \alpha$ and $s_1 s_2 = \beta$ (s_1 could be 0). Thus the inference specializes to

$$\frac{\Gamma, \alpha \leq t, \beta \leq t}{(\Gamma, \alpha\beta \leq 0, \beta \leq t)\nu}$$

It is redundant. In the conclusion, either $\alpha \neq \beta$. Then $\alpha\beta \leq 0$ holds and the conclusion reduces to true. Or else, $\alpha = \beta$ and the conclusion reduces to the first premise.

If $t = 0$, there are two further possibilities. The first possibility is given by the inference

$$\frac{\Gamma, s_1\alpha \leq 0, s_1(\beta s_2') \leq 0}{(\Gamma, \beta s_2' \not\leq \alpha, s_1\beta s_2 \leq 0)\nu}$$

It is redundant. We may assume that s_2' contains no further atom. If $\alpha \neq \beta$, then the conclusion becomes a tautology. If $\alpha = \beta$, then the premise subsumes the conclusion.

The second possibility is given by the inference

$$\frac{\Gamma, \alpha s_1' x \leq 0, \alpha s_1' s_2 \leq 0}{(\Gamma, s_2 \not\leq x, \alpha s_1' s_2 \leq 0)\nu} \qquad \text{(DF)}$$

This inference is irredundant. The constraints from ADC_r can be further strengthened. Neither s_1' nor x may contain an atom and $\alpha s_1'$ must be strictly maximal.

These are all possibilities for (Cut+) and (DF).

(iii) The rule (MI) arises as a special case of similar rules for infinite distributive lattices [20]. These rules can be restricted to atoms at left-hand sides of inequalities, since these atoms are the only non-ground terms in this case. □

Our completeness results immediately transfer to boolean lattices.

Corollary 2. *Consider the transformations ν^r_{ABL} and $\nu_A BL^a$. Given these adaptations, ADC_r and ADC_a are refutationally complete for (the elementary theory of) the following classes.*

(i) Finite atomic boolean lattices.
(ii) Finite preatomic boolean lattices (c.f. [22] for a definition).
(ii) Finite boolean lattices.

Proof. (ad i) Every atomic distributive lattice with a maximal element is atomic boolean. Dually to the case of (3) in Lemma 8, every ordered resolution inference with (33) is redundant. Thus $AD \cup \{(33)\}$ is an orb for ABL. Moreover, it follows that the set is focused. Since the effect of (33) is completely handled by ν^r_{ABL} and ν^a_{ABL}, the inference rules of ADC_r and ADC_a need not be changed. They apply immediately to the respective boolean cases.

(ad ii) Every preatomic boolean lattice is atomic boolean.

(ad iii) Every finite boolean lattice is atomic. □

Lemma 2 improves the completeness result of DC for finite boolean lattices in [21]. Note that it is not the case that every finite distributive lattice is atomic. It is therefore not possible to eliminate negative literals in the distributive case like in the boolean case.

15 Strengthening the Ordering Constraints

In this section we show that the ordering constraints on inequalities in ADC_r and ADC_a can further be strengthened to maximal terms of reduced inequalities in (Cut+). This is analogous to the case of (cut)-based solutions to the uniform word problem for distributive lattices obtained by Knuth-Bendix completion in [19].

Theorem 8. *In ADC_r and ADC_a, the ordering constraints of (Cut+) can be strengthened to x strictly maximal in the respective inequalities.*

Proof. We consider only the case of ADC_r, since ADC_a is a special case thereof. We show by induction on the size of proofs that every refutation in ADC_r can be rearranged to a refutation in which the stronger ordering constraints are satisfied. We proceed in two steps. First we show that the rearrangement is possible for all refutations without (DF). Second we show that whenever there is a (DF)-inference between two (Cut+)-inferences to be rearranged, then the (DF)-inference can be pushed to the top of this three-step sequence.

So let us assume a refutation without any (DF)-inferences. Since the transformations are local in the minor formulas, we disregard all side formulas. The base case is split into four cases. Let s and t be elementary terms. Note that reduction to a literal $\alpha \leq 0$ is the only possible last step of a refutation in ADL.

$$\dfrac{\dfrac{\alpha \leq st \quad t \leq 0}{\alpha \leq s} \qquad s \leq 0}{\alpha \leq 0} \qquad \dfrac{\dfrac{\alpha \leq t \quad t \leq s}{\alpha \leq s} \qquad s \leq 0}{\alpha \leq 0}$$

$$\dfrac{\alpha \leq s \qquad \dfrac{\alpha \leq t \quad st \leq 0}{s \leq 0}}{\alpha \leq 0} \qquad \dfrac{\alpha \leq s \qquad \dfrac{s \leq t \quad t \leq 0}{s \leq 0}}{\alpha \leq 0}$$

In each case, when $t \succ s$, there is nothing to rearrange. Otherwise, if $s \succ t$, one simply permutes the resolution steps. The first proof, for instance, is replaced by

$$\dfrac{\dfrac{\alpha \leq st \quad s \leq 0}{\alpha \leq t} \qquad t \leq 0}{\alpha \leq 0}$$

For the induction step we assume that all smaller proofs have been rearranged and we can consider the top-most inference in the tree. The two possible derivations are

$$\dfrac{\dfrac{s_1 \leq xyt_1 \quad s_2 y \leq t_2}{s_1 s_2 \leq xt_1 t_2} \qquad s_3 x \leq t_3}{s_1 s_2 s_3 \leq t_1 t_2 t_3}$$

$$\dfrac{\dfrac{s_1 xy \leq t_1 \quad s_2 \leq yt_1}{s_1 s_2 x \leq t_1 t_2} \qquad s_3 \leq xt_3}{s_1 s_2 s_3 \leq t_1 t_2 t_3}$$

Again, if $y \succ x$, there is nothing to rearrange. Otherwise, if $x \succ y$, on simply permutes the derivation steps. The first proof, for instance, is replaced by

$$\dfrac{\dfrac{s_1 \leq xyt_1 \quad s_3 x \leq t_3}{s_1 s_3 \leq yt_1 t_3} \qquad s_2 y \leq t_2}{s_1 s_2 s_3 \leq t_1 t_2 t_3}$$

Note that we have always performed the idempotence steps implicitly, but also explicit factoring could be used. The situation is then precisely like that of elimination of blocking inferences (c.f. [21]).

We now argue that a (DF)-inference can always be pushed to the top of a three-step sequence with a potential rearrangement. We restrict our attention

to the first of the (DF) inference rules. The argument with the second one is precisely dual.

According to the ordering constraints, there is precisely one situation, when there is a "sandwich" (DF)-inference between two (Cut+)-inferences. It is

$$\dfrac{\dfrac{\dfrac{\Gamma, s_1 \leq t_1 x, s_1 \leq t_1 y \quad \Gamma', x \leq t_2}{\Gamma, \Gamma', s_1 \leq t_1 t_2, s_1 \leq t_1 y}}{\Gamma, \Gamma' s_1 t_2 \not\leq y, s_1 \leq t_1 y} \qquad \Gamma'', s_2 t_1 \leq t_3}{\Gamma, \Gamma', \Gamma'', s_1 t_2 \not\leq y, s_1 s_2 \leq y t_3}$$

Here, x is a generator and t_1 is a generator that is strictly maximal in the minor formulas of (DF). We replace the inference by

$$\dfrac{\dfrac{\Gamma, s_1 \leq t_1 x, s_1 \leq t_1 y}{\Gamma, s_1 x \not\leq y s_1 \leq t_1 y} \quad \Gamma'', s_2 t_1 \leq t_3}{\Gamma, \Gamma'', s_1 x \not\leq y, s_1 s_3 \leq y t_3}$$

Using negative chaining with $\Gamma', s_1 s_2 \leq y t_3$, we obtain the same conclusion as that of the first inference. Also the strong ordering constraints are then satisfied. Or else we can use (atomic) to transform the conclusion and use (Cut+). We then obtain an equivalent result. □

16 Decidability

In this section we briefly address the use of ADL_r and ADL_a as decision procedures. These properties arise as corollaries to Theorem 6 and Theorem 7.

Corollary 3. ADC_r *and its extension from Theorem 8 decide the reduced clausal theories of the following classes.*

(i) Finite ADL.
(ii) Finite BL.

Proof. Note that by Mc Kinsey's theorem (Theorem 3) and Proposition 1, we can restrict our attention to the respective Horn theories. Thus in particular the (DF) rules are not applicable.

(ad i) Finitely presented atomic distributive lattices are finite. Modulo ACI, the inferences of ADC_r introduce only finitely many new atoms. Since the initial clause set is ground, Skolemization does not introduce variables. Moreover, only finitely many Skolem constants must be introduced. This is justified by the following argument. First, new atoms only arise at left-hand sides of inequalities by (atomic). Second, by Lemma 1 (iv), an inequality $\alpha s \not\leq t$ is equivalent to $\alpha t \leq 0$. This can be used in hypothesis elimination. Thus new atoms must only be introduced for inequalities whose left-hand sides do not contain an atom. But only finitely such terms can be built up to ACI in a finite atomic distributive lattice and also only finitely many right-hand sides, since nothing new is added there. Thus there are only finitely many Skolem functions that can be introduced for

these finitely many inequalities. Since besides this, the inference rules of ADC_r do not add any new symbols, only finitely many inferences lead to irredundant conclusions. Together with refutational completeness this implies that the procedure terminates after finitely many steps. The resulting orb contains the empty clause if and only if the initial clause set was inconsistent.

(ad ii) Immediate from (i) and refutational completeness of the extension of ADL_r to finite boolean lattices (Corollary 2). □

Corollary 4. ADC_a *and its extension from Theorem 8 decide the following classes.*

(i) Finite ADL.
(ii) Finite BL.

Proof. Again, we can restrict our attention to the Horn theories.

(ad i) The proof is similar to that of Corollary 3. Now, the rules (ν_2) and (ν_4) corresponding to (atomic) and extensionality (30) introduce new atoms at left-hand sides of inequalities for each inequality that does not contain already an atom. Since negative and positive inequalities with this property are disjoint, also the scopes of the Skolemization is disjoint. Thus all atoms generated by (atomic) are Skolemized by constants and therefore of the form $\alpha(c)$, whereas all atoms generated by (30) are of the form $\alpha(x)$, for some fresh variable x. Now when two atoms are unified, the substitution either identifies two variables or maps a variable to a constant. Thus again, only finitely many different atoms are generated and therefore the number of lattice inequalities generated by the procedure is finitely bounded modulo ACI. Since besides this, the inference rules of ADC_a do not add any new symbols, only finitely many inferences lead to irredundant conclusions. Together with refutational completeness this again implies that the procedure terminates after finitely many steps. The resulting resolution basis contains the empty clause if and only if the initial clause set was inconsistent.

(ad ii) Again immediate from (i). □

Corollary 5. ADC_r *and* ADC_a *have the following properties.*

(i) They decide the uniform word problem for ADL *and* ABL
(ii) They decide the universal theories of ADL *and* ABL.

Proof. (ad i) By Corollary 3 and Corollary 4, since every finitely presented distributive and boolean lattice is finite.

(ad ii) Immediate from (i) together with McKinsey's theorem 3 and Proposition 1. □

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 [9], while the elementary theory of boolean lattices and atomic boolean lattices is decidable [23,14].

17 The Non-Ground Case

In this section we extend ADC_r to the non-ground case. This means in particular, that the lattices under consideration can be infinite. For the case of DC, the negative chaining rules now involve ACI-unification, which may lead to an explosion of the search space, since the number of most general unifiers may be enormous. In the atomic case, fortunately, these chaining rules are fortunately unnecessary, like in the finite case.

In order to avoid the variable-critical pairs that arise in non-symmetric rewriting and completion, we restrict ourselves to the case that all free function are *non-monotonic*. This holds in particular for Skolem-functions.

Finally, some care has to be taken with idempotence, which can no longer be handled implicitly by a simplification rule. This yields factoring rules also at the lattice level.

Definition 8 (Chaining for Atomic Distributive Lattices). *Let $\succ$ be the (non-ground) literal and clause ordering of section 11. Let all clauses be reduced. The* ordered chaining calculus for atomic distributive lattices ADC_ω *consists of the deductive inference rules and the redundancy elimination rules of* OR *and the following inference rules. We also write ν instead of ν^r_{ADL}.*

$$\frac{\Gamma, \alpha(s) \leq t}{(\Gamma\sigma)\nu}, \tag{Atom1}$$

where $\sigma : t \mapsto 0$.

$$\frac{\Gamma, s_1 \ldots s_i \ldots s_k \ldots s_m \leq t}{(\Gamma\sigma, s_1\sigma \ldots s_i\sigma \ldots s_m\sigma \leq t\sigma)\nu}, \tag{MI}$$

where σ is a most general unifier of s_i and s_k, $s_i\sigma$ is maximal in the respective term and the σ-instance of the minor formula is strictly maximal in the left-hand rule and maximal in the right-hand rule.

$$\frac{\Gamma, s \leq t_1 \ldots t_i \ldots t_k \ldots t_m}{(\Gamma, s\sigma \leq t_1\sigma \ldots t_i\sigma \ldots t_m\sigma)\nu}, \tag{JI}$$

where σ is a most general unifier of t_i and t_k, $t_i\sigma$ is maximal in the respective term and the σ-instance of the minor formula is strictly maximal in the left-hand rule and maximal in the right-hand rule.

$$\frac{\Gamma, s_1 \leq t_1 x \qquad \Gamma', [s_2]x' \leq t_2}{(\Gamma\sigma, \Gamma'\sigma, s_1\sigma[s_2\sigma] \leq t_1\sigma t_2\sigma)\nu}. \tag{Cut}$$

Here, neither x nor x' is a variable, σ is a most general unifier of x and x', $x\sigma$ is strictly maximal in the respective terms, $t_1\sigma x\sigma \not\preceq s_1\sigma$, $[s_2\sigma]x'\sigma \not\preceq t_2\sigma$ and the σ-instances of the minor formulas are strictly maximal with respect to the σ-instances of the side formulas in their respective instances.

$$\frac{\Gamma, s_1 \leq t_1 x \qquad \Gamma', [s_2]s_2' \leq t_2}{(\Gamma\sigma, \Gamma'\sigma, s_1[s_2] \leq t_1 t_2 x')\nu}. \tag{Cut}$$

Here, x is a variable, s_2' is not a variable, $\sigma : x \mapsto s_2' \sqcup x'$, x' is a fresh variable, s_2' is strictly maximal in the respective terms, $t_1\sigma x\sigma \not\trianglelefteq s_1\sigma$, $[s_2\sigma]x'\sigma \not\trianglelefteq t_2\sigma$ and the σ-instances of the minor formulas are strictly maximal with respect to the σ-instances of the side formulas in their respective instances.

A dual rule exists, if the variable occurs at the left-hand side of the second minor formula.

$$\frac{\Gamma, s_1 \leq t_1 x_1 \qquad \Gamma', [s_2]x_2 \leq t_2}{(\Gamma\sigma, \Gamma'\sigma, s_1[s_2]x_2'' \leq t_1 t_2 x_1'')\nu}. \tag{Cut}$$

Thereby, x_1 and x_2 are variables, $\sigma : x_1 \mapsto x_1' \sqcup x_1''$, $x_2 \mapsto x_1' \sqcap x_2''$, x_1', x_1'', and x_2'' are fresh variables, x_1' is strictly maximal in the respective terms, $t_1\sigma x\sigma \not\trianglelefteq s_1\sigma$, $[s_2\sigma]x'\sigma \not\trianglelefteq t_2\sigma$ and the σ-instances of the minor formulas are strictly maximal with respect to the σ-instances of the side formulas in their respective instances.

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

Here, x is not a join, σ is a most general unifier of s and s' and of t_1 and t_1', either $t_1\sigma \not\trianglelefteq s\sigma x\sigma$ and $t_1'\sigma \not\trianglelefteq s'\sigma t_2\sigma$ or $s\sigma \not\trianglelefteq [t_1\sigma]x\sigma$ and $s\sigma \not\trianglelefteq [t_1'\sigma]_m t_2\sigma$ and s can be set to 1 in the conclusion. The σ-instance of the leftmost minor formula is strictly maximal with respect to the σ-instances of the side formulas and the σ-instance of the rightmost minor formula.

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

Here, t_1' is not a join, $\sigma : x \mapsto t_1' \sqcup x'$ is a most general unifier of s and s' and of t_1 and x, either $t_1\sigma \not\trianglelefteq s\sigma t_1'\sigma$ and $x\sigma \not\trianglelefteq s'\sigma t_2\sigma$ or $s\sigma \not\trianglelefteq [t_1\sigma]x\sigma$ and $s\sigma \not\trianglelefteq [x\sigma]t_2\sigma$ and s can be set to 1 in the conclusion. The σ-instance of the leftmost minor formula is strictly maximal with respect to the σ-instances of the side formulas and the σ-instance of the rightmost minor formula.

The rules introducing new variables can be very prolific. It is therefore indispensable to eliminate them as far as possible. This is the subject of Section 18.

Theorem 9. ADC_ω *is refutationally complete.*

Proof. The proof is based on lifting ADC_r. One must therefore show the following. Let C_1 and C_2 be clauses. For all ground instances $C_1\sigma$ and $C_2\sigma$ such there is an inference in ADC_r with conclusion $C_3\sigma$, there is an inference between C_1 and C_2 in ADC_ω with conclusion C_3 such that $C_3\sigma$ is a ground instance of C_3.

The proof is a simple adaptation of that for the non-gound case of DC in [21]. It is straightforward for most of the inference rules. Some rules of ν are now no longer simplifications. This is the case for (ν_{24}) and (ν_{25}) . The first rule now becomes (Atom1). It is derived from (22). The second rule becomes (JI) and (MI). These factoring rules are due to the fact that we always worked modulo idempotence in the ground case.

In case of (Cut), when the term to be cut out is a variable, the situation is more involved. Consider the clauses

$$\Gamma, s_1 \leq t_1 x, \qquad \Gamma', s_2 s_2' \leq t_2.$$

Then x can be substituted for any ground instance $s_2' \sqcup t_1'$, where s_2' is a generator and t_1' an arbitrary join of generators. We obtain the ground instance

$$\frac{\Gamma, s_1 \leq t_1 s_2' t_1' \qquad \Gamma', s_2 s_2' \leq t_2}{\Gamma, \Gamma', s_1 s_2 \leq t_1 t_1' t_2}.$$

of (Cut). Since t_1' is arbitrary, we can represent it by a fresh variable x'. The corresponding substitution σ maps x to $s_2' \sqcup x'$. This yields one of the non-ground (Cut)-rules. The other cases are similar.

A final remark concerns the ordering restrictions of the factoring rules (JI) and (MI). It is obvious that they can be synchronized with those of resolution and chaining. □

18 Simplification and Variable Elimination

Simplification techniques are indispensable for efficient ordered resolution calculi. Resolution blows up the search space and may lead to a combinatorial explosion whereas simplification cuts it down. The more powerful the simplification techniques a prover provides, the greater the chance to receive an answer from the prover instead of getting lost in space-time. Some simplification techniques, like for instance subsumption, concern only the clausal structure. They are common to all ordered resolution calculi. We do not discuss them in this text. See [3] for a discussion. We also do not discuss the various simplification rules that are applicable to DL in general, (c.f. [21]). We restrict our attention to theory-specific and therefore focused simplification techniques for ADL and extensions.

The main idea of simplification is that a clause Γ in a clause set S is simplified by a clause Δ, if adding Δ to S makes Γ redundant and if Δ is a consequence of S. Then of course Δ can be added to S and Γ can be discarded.

Hines [11] discusses several simplification rules. We show that we can reproduce them in the ordered-resolution framework.

The first method is called *chainless sets*. Let α be some atom introduced by one of (ν_2) or (ν_4). Thus α is introduced at the left-hand side of an inequality. The transformations ν have the invariant, that α will never be shuffled to a right-hand side. We now show that also the inference rules (Cut) and (DF) have this invariant.

For (Cut)—either in the ground or the non-ground case—consider the inequality $\alpha \leq t$. The only way to shuffle α to the right-hand side of a premise would be that the second premise is of the form $s \leq xf(x)$. But then $f(x)$ must be bigger than x and the (Cut)-inference would violate the ordering constraints.

The first (DF) rule also does not shuffle α to a right-hand side. In the second one, this can temporarily happen, if $\alpha = x$, but the normalization with ν shuffles α back to the left-hand side again. Such an atom is therefore called *chainless*.

We now show that α cannot be removed by chaining in $\alpha \leq 0$. Consider again $\alpha \leq 0$. In order to chain on α, the second premise must be either of the form $\beta \leq x\gamma$ or $\beta \leq x$, where γ and δ are also smaller than α. In the first case, the second premise is maximal and redundant: it is entailed by $\beta \leq \gamma$. In the second case, the premise is already false.

Thus the only way to eliminate a literal $\alpha\beta \leq 0$ with chainless α is by reduction with ν by rule (ν_{16}) and unification with respect to (MI). Thus if the premise is false, then the conclusion is false and consequently, if the conclusion holds, then so does the premise. That is adding the conclusion makes the premise redundant. It can therefore be discarded.

Another simplification rule works for distributive lattices. Consider the chaining inference

$$\frac{\Gamma, s_1 \leq t_1 x \qquad \Gamma', s_2 s_2' \leq t_2}{\Gamma\sigma, \Gamma', s_1\sigma s_2 \leq t_1\sigma t_2}$$

If x is a variable that occurs neither in s_1 and t_1, nor in Γ and Δ, then the conclusion of (Cut+) reduces to

$$\Gamma, \Gamma', s_1 s_2 \leq t_1 t_2.$$

It is already properly subsumed by the first premise according to our above considerations or it is identical to it up to renaming of variables. Thus this variable chaining is redundant and need not be performed.

A very interesting simplification holds under the additional assumption that there is no universal set $\neg\exists y\forall x.x \leq y$, or equivalently $\forall x\exists y.x \not\leq y$. Then, every clause of the form $\Gamma, x \leq a$ is subsumed by Γ, if the variable x does not occur in Γ or a. Hines [11] has shown that the two above simplifications can be combined.

$$\Gamma, \bigvee a_i \leq b_i x, \bigvee x \leq c_i \Leftrightarrow \Gamma, a_i \leq b_i$$

Provided x does not occur in a_i, b_i, c_i or Γ. This simplification is also possible in our setting. We must show that the following formula holds.

$$\forall abc.(a \not\leq b \Leftrightarrow \exists x.(a \not\leq bx \wedge x \not\leq c).$$

We do not present a formal argument. Instead we refer to the representation theorem and to set-theoretic intuition. If there is no universal set and if $a \not\leq b$ and c is some other set, then we can always find an atom α that is neither below a, nor b and c. Thus adding α to b still does not force $a \leq b \sqcup \alpha$. Conversely, given an element such that $a \not\leq b \sqcup x$, of course $a \not\leq b$ also holds.

Using these simplification rules, we can restrict to a certain extend the prolific chaining inferences with variables. The development of more such rules is a main task for the future. The ultimate goal is of course the avoidance of all variable chainings. It is a very challenging open question whether this goal can be reached.

19 Conclusion

In [22], we proposed atomic distributive lattices as an algebraic core calculus for reasoning about sets in program development methods like B or Z. Here, we have developed axiomatizations that support the effective reduction and simplification of terms, inequalities and clauses and yield several modular extensions of a focused ordered resolution calculus for distributive lattices to atomic distributive and atomic boolean lattices. In particular, these extensions simplify their predecessors. This nicely mirrors the fact that atomic lattices are mathematically simpler than non-atomic ones. We do not know of any other theories of comparable complexity that have been integrated into an automated proof-search procedure so far.

Our results are only a first step towards interesting practical applications in formal methods. We envision the following further work. First and most important, the calculus and the associated proof-search method should be implemented and integrated into an applicable formal method. Second, in order to achieve the first goal, more structure should be added, for instance, types for sets, pairs, comprehension, infinite sets, a choice function (c.f. [1]) and basic data-structures and entities like lists, trees and numbers. Third, the transformations ν should be optimized, further simplification techniques should be developed. Our theoretical results then open the way for operational automated reasoning about sets in the context of industrial-strength formal methods.

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. W. Bledsoe, K. Kunen, and R. Shostak. Completeness results for inequality provers. *Artificial Intelligence*, 27:255–288, 1985.
6. W. W. Bledsoe and L. M. Hines. Variable elimination and chaining in a resolution-based prover for inequalities. In W. Bibel and R. Kowalski, editors, *5th Conference on Automated Deduction*, volume 87 of *LNCS*, pages 70–87. Springer-Verlag, 1980.
7. P. Le Chenadec. *Canonical Forms in Fintetely Presented Algebras*. Research Notes in Theoretical Computer Science. Pitman, 1986.
8. R. P. Dilworth. Lattices with unique complements. *Trans. Amer. Math. Soc.*, 57:123–154, 1945.
9. A. Grzegorczyk. Undecidability of some topological theories. *Fund. Math.*, 38:137–152, 1951.
10. H. Hermes. *Einführung in die Verbandstheorie*. Springer-Verlag, 1967.
11. L. Hines. Str+ve$\subseteq$: The Str+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.

12. L. M. Hines. Completeness of a prover for dense linear orderings. *J. Automated Reasoning*, 8:45–75, 1992.
13. J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, 1991.
14. D. Kozen. Complexity of Boolean algebras. *Theoretical Computer Science*, 10:221–247, 1980.
15. J. McKinsey. The decision problem for some classes of sentences without quantifiers. *Journal of Symbolic Logic*, 8:61–76, 1943.
16. M. M. Richter. Some reordering properties for inequality proof trees. In E. Börger, G. Hasenjaeger, and D. Rödding, editors, *Logic and Machines: Decision Problems and Complexity, Proc. Symposium "Rekursive Kombinatorik"*, volume 171 of *LNCS*, pages 183–197. Springer-Verlag, 1983.
17. M. Rusinowitch. *Démonstration Automatique: Techniques de Réecriture*. Science Informatique. InterEditions, Paris, 1989.
18. G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic*. PhD thesis, Institut für Informatik, Universität des Saarlandes, 1998.
19. 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.
20. 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.
21. 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.
22. G. Struth. A calculus for set-based program development I: Mathematical foundations. Technical Report 2003-15, Institut für Informatik; Universität Augsburg, 2003.
23. A. Tarski. Arithmetical classes and types of Boolean algebras. *Bull. Am. Math. Soc.*, 55(64):1192, 1949.
24. U. Waldmann. *Cancellative Abelian Moioids in Refutational Theorem Proving*. PhD thesis, Institut für Informatik, Universität des Saarlandes, 1997.
25. U. Werz. First-order theorem proving modulo equations. Technical Report MPI-I-92-216, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1992.