A Generalized Resolution Theorem

Pascal Hitzler

Artificial Intelligence Institute
Department of Computer Science
Dresden University of Technology

Abstract

W.C. Rounds and G.-Q. Zhang have recently proposed to study a form of resolution on algebraic domains [1]. This framework allows reasoning with knowledge which is hierarchically structured and forms a (suitable) domain, more precisely, a coherent algebraic cpo as studied in domain theory. In this paper, we give conditions under which a resolution theorem — in a form underlying resolution-based logic programming systems — can be obtained. The investigations bear potential for engineering new knowledge representation and reasoning systems on a firm domain-theoretic background.

Keywords: domain theory; automated theorem proving; domain logics; resolution

1 Introduction

Domain Theory [2] is an abstract mathematical theory for programming semantics and has grown into a respected field on the borderline between mathematics and computer science. Relationships between domain theory and logic were noted early on by Scott [3], and subsequently developed by many authors, including Smyth [4], Abramsky [5], and Zhang [6]. There has been much work on the use of domain logics as logics of types and of program correctness, with a focus on functional and imperative languages. However, there has been only little work relating domain theory to logic programming or other AI paradigms, two exceptions being the application of methods from quantitative domain theory to the semantic analysis of logic programming paradigms studied by Hitzler and Seda [7, 8], and the work of Rounds and Zhang on the use of domain logics for disjunctive logic programming and default reasoning [9, 1].

The latter authors, in [1], introduced a form of clausal logic generalized to coherent algebraic domains, motivated by theoretical investigations into the logical nature of ordered spaces occuring in domain theory. In essence, they propose to interpret finite sets of compact elements as abstract formal clauses, yielding a theory which links standard domain-theoretic notions to corresponding logical notions. Amongst other things, they establish a sound and complete proof theory based on a generalized resolution rule, and a form of disjunctive logic programming in domains. A corresponding semantic operator turns out to be Scott-continuous.

In this paper, we study this clausal logic, henceforth called *logic RZ* for convenience. The occurrence of a proof theory based on a generalized resolution rule poses the question whether results underlying resolution-based logic programming systems can be carried over to the logic RZ. One of the most fundamental results underlying these systems is the *resolution theorem* which states that a clause X is a logical consequence of a theory T if and only if it is possible to derive a contradiction, i.e. the empty clause, via resolution from the theory $T \cup \{\neg X\}$ [10, 11].

What we just called *resolution theorem* is certainly an immediate consequence of the fact that resolution is sound and complete for classical logic. However, it is not obvious how it can be transfered to the logic RZ, mainly because it necessitates negating a clause, and negation is not available in the logic RZ in explicit form. This observation will lead our thoughts, and in the end we will develop conditions on the underlying domain which ensure that a negation is present which allows to prove an analogon of the theorem.

The paper is structured as follows. In Section 2 we review the most fundamental definitions from the logic RZ, as laid out in [1]. In Section 2.2 we recall the corresponding proof theory, based on a form of resolution for this framework. In Section 3 we will simplify the proof theory and provide a rule system which is simpler and easier to work with. The remainder of the paper is devoted to determining conditions under which a resolution theorem, in the form mentioned above, can be proven for the logic RZ. These conditions will involve atomicity of the underlying domain, studied in Section 4, and a form of negation for these spaces, studied in Section 5. We will conclude in Section 6.

An extended abstract of this paper appeared in [12].

2 Preliminaries

2.1 The Logic RZ

A *partially ordered set* is a pair $(D, \sqsubseteq)$, where D is a nonempty set and $\sqsubseteq$ is a relexive, antisymmetric, and transitive relation on D. A subset X of a partially ordered set is *directed* if for all $x, y \in X$ there is $z \in X$ with $x, y \sqsubseteq z$. An *ideal* is a directed and downward closed set. A *complete partial order*, *cpo* for short, is a partially ordered set $(D, \sqsubseteq)$ with a least element $\bot$, called the *bottom element* of $(D, \sqsubseteq)$, and such that every directed set in D has a least upper bound, or supremum, $\bigsqcup D$. An element $c \in D$ is said to be *compact* or *finite* if whenever $c \sqsubseteq \bigsqcup L$ with L directed, then there exists $e \in L$ with $c \sqsubseteq e$. The set of all compact elements of a cpo D is denoted by $\mathsf{K}(D)$. An *algebraic cpo* is a cpo such that every $e \in D$ is the directed supremum of all compact elements below it.

A set $U \subseteq D$ is said to be *Scott open*, or just *open*, if it is upward closed and for any directed $L \subseteq D$ we have $\bigsqcup L \in U$ if and only if $U \cap L \neq 0$. The *Scott topology* on D is the topology whose open sets are all Scott open sets. An open set is *compact open* if it is compact in the Scott topology. A *coherent algebraic cpo* is an algebraic cpo such that the intersection of any two compact open sets is compact open. This coincides with the coherency notion defined in [2], which may be consulted as basic reference for domain theory. We will not make use of many topological notions in the sequel. So let us note that coherency of an algebraic cpo implies that the set of all minimal upper bounds of a finite number of compact elements is finite, i.e. if $c_1, \ldots, c_n$ are compact elements, then the set $\mathsf{mub}\{c_1, \ldots, c_n\}$ of minimal upper bounds of these elements is finite. Note that $\mathsf{mub}\,\emptyset = \{\bot\}$, where $\bot$ is the least element of D.

In the following, $(D, \sqsubseteq)$ will always be assumed to be a coherent algebraic cpo. We will also call these spaces *domains*. Two elements $c, d \in D$ are called *inconsistent*, symbolically $c \not\uparrow d$, if c and d have no common upper bound.

Following [13], an element $a \in D$ is called an *atom*, or an *atomic element*, if whenever $x \sqsubseteq a$ we have $x = a$ or $x = \bot$. The set of all atoms of a domain is denoted by $\mathsf{A}(D)$.

2.1 Definition Let D be a coherent algebraic cpo with set $\mathsf{K}(D)$ of compact elements. A *clause* is a finite subset of $\mathsf{K}(D)$. We denote the set of all clauses over D by $\mathcal{C}(D)$. If X is a clause and $w \in D$, we write $w \models X$ if there exists $x \in X$ with $x \sqsubseteq w$, i.e. X contains an element below w.

A *theory* is a set of clauses, which may be empty. An element $w \in D$ is a *model* of a theory T, written $w \models T$, if $w \models X$ for all $X \in T$ or, equivalently, if every clause $X \in T$ contains an element below w.

A clause X is called a *logical consequence* of a theory T, written $T \models X$, if $w \models T$ implies $w \models X$. If $T = \{E\}$, then we write $E \models X$ for $\{E\} \models X$. Note that this holds if and only if for every $w \in E$ there is $x \in X$ with $x \sqsubseteq w$.

For two theories T and S, we say that $T \models S$ if $T \models X$ for all $X \in S$. We say that T and S are (*logically*) *equivalent*, written $T \sim S$, if $T \models S$ and $S \models T$. In order to avoid confusion, we will throughout denote the empty clause by $\{\}$, and the empty theory by $\emptyset$. A theory T is (*logically*) *closed* if $T \models X$ implies $X \in T$ for all clauses X. It is called *consistent* if $T \not\models \{\}$ or, equivalently, if there is w with $w \models T$.

Rounds and Zhang originally set out to characterize logically the notion of *Smyth powerdomain* of coherent algebraic cpos. It naturally lead to the clausal logic RZ from Definition 2.1. Indeed, as was shown in [1], the Smyth powerdomain of any coherent algebraic domain is isomorphic to the set of all consistent closed theories over the domain, ordered by set-inclusion. A corollary from the proof is that a clause is a logical consequence of a theory if and only if it is a logical consequence of a finite subset of the theory, which is a compactness theorem for the logic RZ.

2.2 Example In [1], the domain $\mathbb{T}^\omega$ from [14], here denoted $\mathbb{T}^\mathcal{V}$, was given as a running example. Consider some three-valued logic in the propositional case, with the usual (knowledge)-ordering on the set $\mathbb{T} = \{\mathbf{f}, \mathbf{u}, \mathbf{t}\}$ of truth values given by $\mathbf{u} < \mathbf{f}$ and $\mathbf{u} < \mathbf{t}$. This induces a pointwise ordering on the space $\mathbb{T}^\mathcal{V}$ of all interpretations (or *partial truth assignments*), where $\mathcal{V}$ is the (countably infinite) set of all propositional variables in the language under consideration. The partially ordered set $\mathbb{T}^\mathcal{V}$ is a coherent algebraic cpo. Compact elements in $\mathbb{T}^\mathcal{V}$ are those interpretations which map all but a finite number of propositional variables to $\mathbf{u}$. We denote compact elements by strings such as $pq\overline{r}$, which indicates that p and q are mapped to $\mathbf{t}$ and r is mapped to $\mathbf{f}$.

We note that $\{e \mid e \models \phi\}$ is upward-closed for any logical formula ϕ if considering e.g. Kleene's strong three-valued logic, which has been recognized as being important in a logic programming context [15]. A clause in $\mathbb{T}^{\mathcal{V}}$ is a formula in disjunctive normal form, e.g. $\{pq\overline{r}, \overline{p}q, r\}$ translates to $(p \wedge q \wedge \neg r) \vee (\neg p \wedge q) \vee r$.

We also note that every compact element in $\mathbb{T}^{\mathcal{V}}$ can be uniquely expressed as the supremum of a finite number of atomic elements, and the set of all atomic elements is $\mathsf{A}\left(\mathbb{T}^{\mathcal{V}}\right) = \mathcal{V} \cup \{\overline{v} \mid v \in \mathcal{V}\}$. Furthermore, there exists a bijective function $\overline{} : \mathsf{A}\left(\mathbb{T}^{\mathcal{V}}\right) \to \mathsf{A}\left(\mathbb{T}^{\mathcal{V}}\right) : p \to \overline{p}$ which extends naturally to a Scott-continuous involution on all of $\mathbb{T}^{\mathcal{V}}$ via $\overline{p_1 \dots p_n} = \overline{p_1} \dots \overline{p_n}$. In the following, a clause over a domain D will be called an *atomic clause* if it is a finite subset of $\mathsf{A}(D)$. Atomic clauses on $\mathbb{T}^{\mathcal{V}}$ correspond to propositional clauses in the classical sense. Note that $p \not\uparrow \overline{p}$ for $p \in \mathsf{A}\left(\mathbb{T}^{\mathcal{V}}\right)$ and in general for all $c \in \mathsf{K}\left(\mathbb{T}^{\mathcal{V}}\right)$ we have $c \not\uparrow \overline{c}$.

The following example shows how knowledge can be represented in algebraic domains. For convenience, examples will be presented as subsets of $\mathbb{T}^{\mathcal{V}}$, in the notation from Example 2.2.

2.3 Example Consider the subspace of $\mathbb{T}^{\mathcal{V}}$ constituted by the elements $\bot$, b (is a bird), f (flies), $\overline{f}$ (does not fly), a (lives in australia), s (lives near south pole), $b\overline{f}s$ (is a penguin), and $b\overline{f}a$ (is an ostrich). Then e.g. $\left\{\{b\}, \left\{\overline{f}\right\}\right\} \models \{a, s\}$.

As to the knowledge representation capabilities of the logic RZ, we remark that some first investigations have exhibited a strong link to formal concept analysis [16, 17].

2.2 Resolution in the logic RZ

In [1], a sound and complete proof theory, using *clausal hyperresolution*, was given as follows, where $\{X_1, \dots, X_n\}$ is a clause set and Y a clause.

$$\frac{X_i; \quad a_i \in X_i \quad (i \le n); \quad \mathsf{mub}\{a_i \mid i \le n\} \models Y}{Y \cup \bigcup_{i \le n} (X_i \setminus \{a_i\})} \quad \text{(hr)}$$

This rule is sound in the following sense: Whenever $w \models X_i$ for all i, then for any admissible choice of the a_i and Y in the antecedent, we have $w \models Y \cup \bigcup_{i=1}^{n} (X_i \setminus \{a_i\})$.

For completeness, it is necessary to adjoin to the above clausal hyperresolution rule a special rule which allows the inference of any clause from the empty clause. We indicate this rule as follows.

$$\frac{\{\}; \qquad Y \in \mathcal{C}(D)}{Y} \quad \text{(spec)}$$

With this addition, given a theory T and a clause X with $T \models X$, we have that $T \vdash^* X$, where $\vdash^*$ stands for a finite number of applications of the clausal hyperresolution rule together with the special rule.

Furthermore, [1, Remark 4.6] shows that binary hyperresolution, together with (spec), is already complete, i.e. the system consisting of the *binary clausal hyperresolution* rule

$$\frac{X_1 \quad X_2; \quad a_i \in X_i; \quad \mathsf{mub}\{a_1, a_2\} \models Y}{Y \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})} \quad \text{(bhr)}$$

together with the special rule is sound and complete.

If the set $\{a_1, a_2\}$ is inconsistent, then $\mathsf{mub}\{a_1, a_2\} = \{\}$. Since $\{\} \models \{\}$, clausal hyperresolution generalizes the usual notion of resolution, given by the following rule.

$$\frac{X_1 \quad X_2; \quad a_i \in X_i; \qquad a_1 \not\uparrow a_2}{(X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})} \quad (\mathsf{r})$$

2.4 Example Returning to Example 2.3, note that e.g. $\{\{b\}, \{\overline{f}\}\} \vdash \{b\overline{f}s, b\overline{f}a\}$ using (bhr).

3 Simplifying the Resolution System

Note that two special instances of the clausal hyperresolution rule are as follows, which we call the *reduction rule* and the *extension rule*.

$$\frac{X; \quad \{a, y\} \subseteq X; \quad y \sqsubseteq a}{X \setminus \{a\}} \quad (\mathsf{red}),$$

$$\frac{X; \quad y \in \mathsf{K}(D)}{\{y\} \cup X} \quad (\mathsf{ext})$$

Indeed, the first rule follows from (hr) since $a \in X$ and $\{a\} \models \{y\}$, while the latter rule follows since $\{a\} \models \{a, y\}$ for all $y \in \mathsf{K}(D)$. The special rule (spec) can be understood as an instance of (ext). Note also that resolution (r) together with (ext) and (red) is not complete. In order to see this, we refer again to Example 2.2. Let $T = \{\{p\}, \{q\}\}$ and $X = \{pq\}$. Then $T \models X$ but there is no way to produce X from T using (r), (ext) and (red) alone. Indeed, it is easy to show by induction that any X which can be derived from T by using only (r), (ext) and (red), contains either p or q, which suffices.

It is our desire to provide a sound and complete system whose rules are as simple as possible. Consider the following rule, which we call *simplified hyperresolution*. It is easy to see that it is an instance of (hr) and more general than (r).

$$\frac{X_1 \quad X_2; \quad a_i \in X_i}{\mathsf{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})} \quad (\mathsf{shr})$$

3.1 Theorem The system consisting of (shr), (ext) and (red) is complete.

Proof: In order to show completeness, we derive (bhr) from (shr), (ext) and (red). Let X_1, X_2 be given with $a_1 \in X_1$ and $a_2 \in X_2$ with $a_1 \uparrow a_2$. Furthermore, let Y be a clause with $\mathsf{mub}\{a_1, a_2\} \models Y$. Let $\mathsf{mub}\{a_1, a_2\} = \{b_1, \ldots, b_n\}$. Then for every b_i there exists $y_i \in Y$ with $y_i \sqsubseteq b_i$. Using (shr), from X_1 and X_2 we can derive $X_3 = \mathsf{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$, and with repeated application of (ext) and (red) we obtain from this $X_4 = \{y_1, \ldots, y_n\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$. Finally, using (ext) repeatedly, we can add to X_4 all remaining elements from Y. The argumentation for $a_1 \not\uparrow a_2$ is similar. This completes the proof. ■

We note that a rule with weaker preconditions than (red) suffices, which we call the *weakening rule*:

$$\frac{X; \qquad a \in X; \qquad y \sqsubseteq a}{\{y\} \cup (X \setminus \{a\})} \quad (\mathsf{w})$$

Indeed, (red) can be derived from (w) as follows. Let $\{a, y\} \subseteq X$ with $y \sqsubseteq a$. Then in particular $a \in X$, i.e. using (w) we can derive $\{y\} \cup (X \setminus \{a\})$ which is equal to $X \setminus \{a\}$ since y is already contained in X. On the other hand, (w) can be derived from (red) and (ext) as follows. Let $a \in X$ and $y \sqsubseteq a$. If $a = y$ then there is nothing to show, so assume $a \neq y$. Then $X \vdash X \cup \{y\}$ by the extension rule, so the reduction rule can be applied, yielding $(X \cup \{y\}) \setminus \{a\}$ as required.

The following technical result is inspired by [18, Theorem 7].

3.2 Proposition For clauses $X_1, \ldots, X_n$ we have $\{X_1, \ldots, X_n\} \models X$ if and only if $\{\{a_1\}, \ldots, \{a_n\}\} \models X$ for all $(a_1, \ldots, a_n) \in X_1 \times \ldots \times X_n$.

Proof: Assume $\{X_1, \ldots, X_n\} \models X$ and let $a_i \in X_i$ be arbitrarily chosen for $i = 1, \ldots, n$. Then $\{a_i\}\ X_i$ for all $i = 1, \ldots n$ by (ext) and therefore $\{\{a_1\}, \ldots, \{a_n\}\} \models \{X_1, \ldots, X_n\} \models X$.

Conversely, assume that $\{\{a_1\}, \ldots, \{a_n\}\} \models X$ for all $(a_1, \ldots, a_n) \in X_1 \times \ldots \times X_n$ and let $w \in D$ with $w \models \{X_1, \ldots, X_n\}$, i.e. $w \models X_i$ for all $i = 1, \ldots, n$. Then for all $i = 1, \ldots, n$ there is $a_i \in X_i$ with $a_i \sqsubseteq w$. So for all $i = 1, \ldots, n$ choose a_i with $a_i \sqsubseteq w$. Then $w \models \{\{a_1\}, \ldots, \{a_n\}\}$ and by assumption we obtain $w \models X$. ■

We call the system consisting of the rules (red), (ext) and (shr) the RAD system, from *Resolution in Algebraic Domains*. For two theories T and S, we write $T \vdash^* S$ if $T \vdash^* A$ for each $A \in S$, and for clauses X and Y we write $X \vdash^* Y$, respectively $X \vdash^* T$, for $\{X\} \vdash^* Y$, respectively $\{X\} \vdash^* T$. The symbol $\vdash$ denotes derivation by a single application of one of the rules in RAD. With slight abuse of notation, for two theories T and S we allow to write $T \vdash S$ if $T \vdash X$ for some clause X and $S \subseteq T \cup \{X\}$.

We interpret the RAD rules in the setting of Example 2.2. We already know that clauses correspond to formulas in disjunctive normal form (DNF), and theories to sets of DNF formulas. The weakening rule acts on single clauses and replaces a conjunction contained in a DNF formula by a conjunction which contains a subset of the propositional variables contained in the original conjunction, e.g. $(p \wedge q) \vee r$ becomes $p \vee r$. The extension rule disjunctively extends a DNF formula by a further conjunction of propositional variables, e.g. $(p \wedge q) \vee r$ becomes $(p \wedge q) \vee r \vee (s \wedge q)$. The simplified hyperresolution rule finally takes two DNF formulas, deletes one conjunction from each of them, and forms a disjunction from the resulting formulas together with the conjunction of the deleted items, e.g. $(p \wedge q) \vee r$ and $\neg p \vee (s \wedge r)$ can be resolved to $(p \wedge q) \vee (r \wedge \neg p) \vee (s \wedge r)$.

A more abstract interpretation of the RAD system comes from a standard intuition underlying domain theory. Elements of the domain D are interpreted as pieces of information, and if $x \sqsubseteq y$, this represents that y contains more information than x. Compact elements are understood as items which are computationally accessible. From this point of view, RAD gives a calculus for reasoning about disjunctive information in computation, taking a clause, i.e. a finite set of computationally accessible information items as disjunctive knowledge about these items. The rules from RAD yield a system for deriving further knowledge from the given disjunctive information. The weakening rule states that we can replace an item by another one which contains less information. The extension rule states that we can always extend our knowledge disjunctively with further bits of information. Both rules decrease our knowledge. The simplified hyperresolution rule states that we can disjunctively merge two collections of disjunctive information, while strengthening our knowledge by replacing two of the items from the collections by an item which contains both pieces of information, and deleting the original items.

3.3 Example For Example 2.3, note that $\{\{b\},\{\overline{f}\}\} \vdash \{b\overline{f}s, b\overline{f}a\}$ using (shr), $\{b\overline{f}s, b\overline{f}a\} \vdash \{s, b\overline{f}a\}$ using (w), and finally $\{s, b\overline{f}a\} \vdash \{s, a\}$ using (w) again.

4 Atomic Domains

We simplify proof search via resolution by requiring stronger conditions on the domain.

4.1 Definition An *atomic domain* is a coherent algebraic cpo D with the following property: For all $c \in \mathsf{K}(D)$, the set $\mathsf{A}(c) = \{p \in \mathsf{A}(D) \mid p \sqsubseteq c\}$ is finite and $c = \bigsqcup \mathsf{A}(c)$.

The domain $\mathbb{T}^{\mathcal{V}}$ from Example 2.2 is an example of an atomic domain. In the remainder of this section, D will always be an atomic domain.

We seek to represent a clause X by a finite set $\mathsf{A}(X)$ of atomic clauses which is logically equivalent to X. Given $X = \{a_1, \ldots, a_n\}$, we define $\mathsf{A}(X)$ as follows.

$$\mathsf{A}(X) = \{\{b_1, \ldots, b_n\} \mid b_i \in \mathsf{A}(a_i) \text{ for all } i = 1, \ldots, n\}$$

Then the following theorem holds.

4.2 Theorem For any clause X we have $\mathsf{A}(X) \sim \{X\}$.

Proof: For a clause $X = \{a_1, \ldots, a_n\}$ set $X/a_1 = \{\{b, a_2, \ldots, a_n\} \mid b \in \mathsf{A}(a_1)\}$. Then $X/a_1 \models X$. Indeed, since $\bigsqcup \mathsf{A}(a_1) = a_1$ we obtain $\mathsf{mub}\,\mathsf{A}(a_1) \models \{a_1\}$, and therefore $X/a_1 \vdash^* X$ from (hr).

Now let $X = \{a_1, \ldots, a_n\}$ and let $Y = \{b_1, \ldots, b_n\} \in \mathsf{A}(X)$ with $b_i \in \mathsf{A}(a_i)$ for all i. Then $b_i \sqsubseteq a_i$ for all i and hence $X \vdash^* Y$ by repeated application of the weakening rule. Conversely, define for any compact element a and any set T of clauses: $T/a = \{Z \in T \mid a \notin Z\} \cup \{\{b\} \cup (Z \setminus \{a\}) \mid b \in \mathsf{A}(a), a \in Z \in T\}$. So for any clause Z and $a \in Z$ we have $\{Z\}/a = Z/a$ and we obtain that $T/a \models T$ for all sets of clauses T and $a \in \mathsf{K}(D)$. Now let $X = \{a_1, \ldots, a_n\}$. Then $(\ldots(X/a_1)/a_2 \ldots)/a_n = \mathsf{A}(X)$ and consequently $\mathsf{A}(X) \models X$, which completes the proof. ■

In view of Theorem 4.2, it suffices to study $T \vdash^* X$ for theories T and atomic clauses X. We can actually obtain a stronger result, as follows, which provides some kind of normal forms of derivations. For a theory T, define $\mathsf{A}(T) = \{\mathsf{A}(X) \mid X \in T\}$.

4.3 Theorem Let D be an atomic domain, T be a theory, X be a clause and

$$T \vdash T_1 \vdash \cdots \vdash T_N \vdash X$$

be a derivation in RAD. Then there exists a derivation

$$\mathsf{A}(T) \vdash^* \mathsf{A}(T_1) \vdash^* \cdots \vdash^* \mathsf{A}(T_N) \vdash^* \mathsf{A}(X)$$

using only the *atomic extension rule*

$$\frac{X; \quad y \in \mathsf{A}(D)}{\{y\} \cup X} \quad (\mathsf{axt})$$

and the *multiple atomic shift rule* (mas), as follows.

$$\frac{a_i \in X_i; \quad \mathsf{mub}\{a_i \mid i \leq n\} = \{x_j \mid j \leq m\}; \quad b_i \in \mathsf{A}(x_i)}{\{b_1, \dots, b_m\} \cup \bigcup_{i \leq n}(X_i \setminus \{a_i\})}$$

Furthermore, all clauses occuring in the derivation are atomic.

Proof: Let X_1, X_2, X be clauses. We distinguish three cases, from which the assertion follows easily by induction on N.

1. $X_1 \vdash X$ using the reduction rule. First note that the following *atomic shift rule* (ash) is a special instance of the multiple atomic shift rule.

$$\frac{a_1 \in X_1 \quad a_2 \in X_2; \quad a \in \mathsf{A}(x) \text{ for all } x \in \mathsf{mub}\{a_1, a_2\}}{\{a\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})}$$

Indeed, (ash) follows from (mas) with $n = 2$ and $a = b_1 = \ldots = b_k$. Now let $a, y \in X_1$ with $y \sqsubseteq a$ and $X = X_1 \setminus \{a\} = \{y, x_1, \dots, x_n\}$. Let $A \in \mathsf{A}(X)$, say $A = \{y', x'_1, \dots, x'_n\}$ with $y' \in \mathsf{A}(y)$ and $x'_i \in \mathsf{A}(x_i)$ for all i. Without loss of generality we can assume that $\mathsf{A}(y) \subset \mathsf{A}(a)$, so there is $\{a'\} \cup A \in \mathsf{A}(X_1)$ for some $a' \in \mathsf{A}(a) \setminus \mathsf{A}(y)$. So we now have $a', y' \sqsubseteq a$ and $y' \sqsubseteq y$, i.e. $\{y', a', x'_1, \dots, x'_n\} \in \mathsf{A}(X_1)$ and $\{y', y', x'_1, \dots, x'_n\} = A \in \mathsf{A}(X_1)$. So $a' \in \{y', a', x'_1, \dots, x'_n\}$, $y' \in \{y', y', x'_1, \dots, x'_n\}$ and since $y' \sqsubseteq x$ for all $x \in \mathsf{mub}\{y', a'\}$ we can derive $\{y'\} \cup (\{y', a', x'_1, \dots, x'_n\} \setminus \{a'\}) \cup (\{y', y', x'_1, \dots, x'_n\} \setminus \{y'\}) = \{y', x'_1, \dots, x'_n\} = A$ using the atomic shift rule.

2. $X_1 \vdash X$ using the extension rule, i.e. $X = X_1 \cup \{y\}$ for some y. Let $A \in \mathsf{A}(X)$. Then $A = \{y'\} \cup Y$ for some $y' \in \mathsf{A}(y)$ and $Y \in \mathsf{A}(X_1)$. Using the atomic extension rule we can derive $Y \vdash A$ and therefore $\mathsf{A}(X_1) \vdash A$ using the atomic extension rule only, which suffices.

3. $\{X_1, X_2\} \vdash X$ using the simplified hyperresolution rule. Let $a_1 \in X_1$, $a_2 \in X_2$ and $X = \mathsf{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$. Furthermore, let $M = \mathsf{mub}\{a_1, a_2\} = \{m_1, \dots, m_k\}$ and let $A \in \mathsf{A}(X)$, i.e. $A = \{m'_1, \dots, m'_k\} \cup B_1 \cup B_2$, where $m'_i \in \mathsf{A}(m_i)$ for all i, $B_1 \in \mathsf{A}(X_1 \setminus \{a_1\})$ and $B_2 \in \mathsf{A}(X_2 \setminus \{a_2\})$. Note that for all $a'_1 \in \mathsf{A}(a_1)$ we have that $B_1 \cup \{a'_1\} \in \mathsf{A}(X_1)$ and for all $a'_2 \in \mathsf{A}(a_2)$ we have that $B_2 \cup \{a'_2\} \in \mathsf{A}(X_2)$. Let $\mathsf{A}(a_1) = \{a'_1, \dots, a'_{k_1}\}$ and $\mathsf{A}(a_2) = \{a'_{k_1+1}, \dots, a'_{k_1+k_2}\}$. For $i = 1, \dots, k_1$ let $Y_i = B_1 \cup \{a'_i\} \in \mathsf{A}(X_1)$ and for $i = k_1, \dots, k_1 + k_2$ let $Y_i = B_2 \cup \{a'_i\} \in \mathsf{A}(X_2)$. Since $a_1 = \bigsqcup \mathsf{A}(a_1)$ and $a_2 = \bigsqcup \mathsf{A}(a_2)$ we have $\mathsf{mub}\,(\mathsf{A}(a_1) \cup \mathsf{A}(a_2)) = \mathsf{mub}\{a_1, a_2\} = \{m_1, \dots, m_k\} = M$. From the multiple atomic shift rule we obtain (with $i \leq k_1 + k_2$ and $j \leq k$)

$$\frac{a_i \in Y_i \quad \mathsf{mub}\{a'_1, \dots, a'_{k_1+k_2}\} = M, \quad m'_j \in \mathsf{A}(m_j)}{\{m'_1, \dots, m'_k\} \cup \bigcup_{i \leq k_1+k_2}(Y_i \setminus \{a_i\})}$$

Since $Y_i \setminus \{a'_i\} \subseteq B_1$ for $i = 1, \dots, k_1$ and $Y_i \setminus \{a'_2\} \subseteq B_2$ for $i = k_1, \dots, k_1 + k_2$, we obtain $\{m'_1, \dots, m'_k\} \cup \bigcup(Y_i \setminus \{a_i\}) \subseteq A$ which suffices by the atomic extension rule. ■

Note that the atomic extension rule is a special case of the extension rule, and that the multiple atomic shift rule can be obtained as a subsequent application of first the hyperresolution rule (with $Y = \mathsf{mub}\{a_1, \ldots, a_n\}$) and then multiple instances of the reduction rule, hence both rules are sound.

4.4 Remark We note that Theorem 4.3 does not hold if (mas) is replaced by its binary version (bas), as follows.

$$\frac{a_1 \in X_1, a_2 \in X_2; \mathsf{mub}\{a_1, a_2\} = \{x_1 \mid i \leq k\}; b_i \in \mathsf{A}(x_i)}{\{b_1, \ldots, b_k\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})}$$

In order to see this, consider three atomic elements a_1, a_2, a_3 which are mutually consistent with supremum $\sup\{a_i, a_j\} = a_{ij}$, but do not have a common upper bound. Then $\{\{a_1\}, \{a_2\}, \{a_3\}\} \models \{\}$, but the empty clause $\{\}$ cannot be derived from the theory $T = \{\{a_1\}, \{a_2\}, \{a_3\}\}$ using (axt) and (bas) alone. Indeed it is easy to show by induction that every clause which is derived from T using applications of (axt) and (bas) always contains one of the elements a_1, a_2 or a_3.

5 Domains with Negation

We introduce and investigate a notion of negation on domains, motivated by classical negation as in Example 2.2.

5.1 Definition An atomic domain is called an *atomic domain with negation* if there exists an involutive and Scott-continuous *negation function* $^- : D \to D$ with the following properties:

(i) $^-$ maps $\mathsf{A}(D)$ onto $\mathsf{A}(D)$.

(ii) For all $p, q \in \mathsf{A}(D)$ we have $p \not\uparrow q$ if and only if $q = \overline{p}$.

(iii) For every finite subset $A \subseteq \mathsf{A}(D)$ such that $p \uparrow q$ for all $p, q \in A$, the supremum $\bigsqcup A$ exists.

$\mathbb{T}^{\mathcal{V}}$ from Example 2.2 is an example of an atomic domain with negation.

5.2 Proposition Let D be an atomic domain with negation. Then for all $c \in \mathsf{K}(D)$ we have $\overline{c} = \bigsqcup\{\overline{a} \mid a \in \mathsf{A}(c)\}$.

Proof: Let $c \in \mathsf{K}(D)$. Then $c = \bigsqcup \mathsf{A}(c)$, hence $\mathsf{A}(c)$ is consistent. By (ii) of Definition 5.1, we obtain that every pair of elements from $\{\overline{a} \mid a \in \mathsf{A}(c)\}$ is consistent, and by (iii) the supremum $d = \bigsqcup\{\overline{a} \mid a \in \mathsf{A}(c)\}$ exists. From monotonicity of $^-$, we obtain first $d \sqsubseteq \overline{c}$, and then $\overline{d} \sqsubseteq \overline{\overline{c}} = c$. But, again by monotonicity of $^-$, we know that $\overline{d}$ is an upper bound of $\mathsf{A}(c)$, hence $c \sqsubseteq \overline{d}$, and consequently $c = \overline{d}$ and $\overline{c} = d = \bigsqcup\{\overline{a} \mid a \in \mathsf{A}(c)\}$ as required. ■

The following result, an analogon to the resolution theorem mentioned in the introduction, allows one to replace the search for derivations by search for contradiction.

5.3 Theorem Let D be an atomic domain with negation. Let T be a theory and X be an atomic clause. Then $T \models X$ if and only if $T \cup \{\{\bar{a}\} \mid a \in X\} \vdash^* \{\}$.

Proof: Assume $T \models X$. Then $T \vdash^* X$ and $\{X\} \cup \{\{\bar{a}\} \mid a \in X\} \vdash^* \{\}$ follows easily by repeated application of the resolution rule (r).

Conversely, assume $T \cup \{\{\overline{a}\} \mid a \in X\} \vdash^* \{\}$, i.e. $T \cup \{\{\overline{a}\} \mid a \in X\} \models \{\}$. If $T \models \{\}$ then $T \vdash^* \{\} \vdash^* X$. So assume that $T \not\models \{\}$, i.e. there exists $w \in D$ with $w \models T$. We have to show that $w \models X$ for every such w. Since $w \models T$ but $w \not\models T \cup \{\{\overline{a}\} \mid a \in X\}$, we have that there is $a \in X$ with $\overline{a} \not\uparrow w$. Hence there exists $x \in \mathsf{A}(w)$ with $x \not\uparrow \overline{a}$. From the hypothesis we obtain $x = a$. Hence $a \sqsubseteq w$ and therefore, by the weakening rule, $w \vdash^* X$, i.e. $w \models X$. ∎

On atomic domains with negation, we can therefore establish the following sound and complete proof principle.

5.4 Theorem Let T be a theory and X a clause. Consider $T' = \mathsf{A}(T)$. For every atomic clause $A \in \mathsf{A}(X)$ attempt to show $T' \cup \{\{\overline{a}\} \mid a \in A\} \vdash^* \{\}$ using (axt) and (mas). If this succeeds, then $T \models X$. Conversely, if $T \models X$ then there exists a derivation $T' \cup \{\{\overline{a}\} \mid a \in A\} \vdash^* \{\}$ for each $A \in \mathsf{A}(X)$ using only the above mentioned rules.

Proof: If $T' \cup \{\{\overline{a}\} \mid a \in A\} \vdash^* \{\}$, then by Theorem 4.3 the derivation can be carried out using only the mentioned rules and we obtain $T' \cup \{\{\overline{a}\} \mid a \in A\} \models \{\}$. By Theorem 5.3 we obtain $T' \models A$, so $T' \models A$ for all $A \in \mathsf{A}(X)$. By Theorem 4.2 this yields $T' \models X$ and finally we obtain $T \models X$ by application of Theorem 4.2, noting that $T' = \mathsf{A}(T) \sim T$.

Conversely, if $T \models X$ then we have $T' \models A$ for all $A \in \mathsf{A}(X)$, again by Theorems 4.2 and 4.2. Theorem 5.3 then yields $T' \cup \{\{\overline{a}\} \mid a \in A\} \vdash^* \{\}$ for all $A \in \mathsf{A}(X)$, and finally from Theorem 4.3 we obtain that this derivation can be done using only the designated rules. ∎

5.5 Example We give an abstract example, again using notation from Example 2.2, which shows that reasoningin atomic domains with negation does not lead directly back to resoning in $\mathbb{T}^{\mathcal{V}}$. Consider the subcpo constituted by the elements $\{\bot, p, q, r, \overline{p}, \overline{q}, \overline{r}, pqr, \overline{pqr}, p\overline{q}, p\overline{r}, q\overline{p}, q\overline{r}, r\overline{p}, r\overline{q}\}$, which is an atomic domain with negation. Then e.g. $\{\{p\}, \{q\}\} \models \{r\}$. Indeed, $\{\{p\}, \{q\}, \{\overline{r}\}\} \vdash \{\}$ by (mas) because $\mathsf{mub}\{p, q, \overline{r}\} = \{\}$.

6 Conclusions

We have shown that for certain domains logical consequence in the logic RZ can be reduced to search for contradiction, a result which yields a proof mechanism similar to that underlying the resolution principle used in resolution-based logic programming systems. The result should be understood as foundational for establishing logic programming systems on hierarchical knowledge — like e.g. in formal concept analysis — built on a firm domain-theoretic background. Further research is being undertaken to substantiate this.

References

[1] William C. Rounds and Guo-Qiang Zhang. Clausal logic and logic programming in algebraic domains. *Information and Computation*, 171(2):156–182, 2001.

[2] Samson Abramsky and Achim Jung. Domain theory. In Samson Abramsky, Dov Gabbay, and Thomas S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3. Clarendon, Oxford, 1994.

[3] Dana S. Scott. Domains for denotational semantics. In Magens Nielsen and Erik M. Schmidt, editors, *Automata, Languages and Programming, 9th Colloquium, July 1982, Aarhus, Denmark, Proceedings*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer, Berlin, 1982.

[4] Michael B. Smyth. Powerdomains and predicate transformers: A topological view. In Josep Díaz, editor, *Automata, Languages and Programming, 10th Colloquium, July 1989, Barcelona, Spain, Proceedings*, volume 298 of *Lecture Notes in Computer Science*, pages 662–675. Springer, Berlin, 1989.

[5] Samson Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[6] Guo-Qiang Zhang. *Logic of Domains*. Birkhauser, Boston, 1991.

[7] Pascal Hitzler. *Generalized Metrics and Topology in Logic Programming Semantics*. PhD thesis, Department of Mathematics, National University of Ireland, University College Cork, 2001.

[8] Pascal Hitzler and Anthony K. Seda. Generalized metrics and uniquely determined logic programs. *Theoretical Computer Science*, 305(1–3):187–219, 2003.

[9] Guo-Qiang Zhang and William C. Rounds. Semantics of logic programs and representation of Smyth powerdomains. In Klaus Keimel et al., editors, *Domains and Processes*, pages 151–179. Kluwer, 2001.

[10] J. Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[11] John W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 1988.

[12] Pascal Hitzler. A resolution theorem for algebraic domains. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 2003*, pages 1339–1340. Morgan Kaufmann Publishers, 2003.

[13] Peter T. Johnstone. *Stone Spaces*. Number 3 in Cambridge studies in advanced mathematics. Cambridge University Press, 1982.

[14] Gordon Plotkin. T^ω as a universal domain. *Journal of Computer and System Sciences*, 17:209–236, 1978.

[15] Melvin Fitting. A Kripke-Kleene-semantics for general logic programs. *The Journal of Logic Programming*, 2:295–312, 1985.

[16] Pascal Hitzler and Matthias Wendt. Formal concept analysis and resolution in algebraic domains. In Aldo de Moor and Bernhard Ganter, editors, *Using Conceptual Structures — Contributions to ICCS 2003*, pages 157–170. Shaker Verlag, Aachen, 2003.

[17] Guo-Qiang Zhang. Chu spaces, concept lattices, and domains. In *Proceedings of the Nineteenth Conference on the Mathematical Foundations of Programming Semantics, March 2003, Montreal, Canada*, volume 83 of *Electronic Notes in Theoretical Computer Science*, 2003.

[18] Thierry Coquand and Guo-Qiang Zhang. Sequents, frames, and completeness. In *14th International Workshop on Computer Science Logic, Fischbachau, Germany, August 2000*, volume 1862 of *Lecture Notes in Computer Science*, pages 277–291. Springer, 2000.

Pascal Hitzler (PhD, Dipl.-Math.), born 1971 in Germany, studied Mathematics and Computer Science at the University of Tübingen in Germany, completed with distinction in 1998. He then was a PhD student of Dr. Anthony K. Seda at the National University of Ireland, University College Cork, where he finished his dissertation in Mathematics in 2001. Since then, he is a research assistant of Prof. Dr. Steffen Hölldobler at the Artificial Intelligence Institute at the Department of Computer Science of Dresden University, Germany. His publication record shows publications e.g. in pure mathematics (topology and fixed-point theory), foundations of artificial intelligence (domain-theoretic aspects of knowledge representation and reasoning; logic and connectionism) and in programming language semantics (in particular logic programming). He is also actively involved in running enhancement programmes for skilled high-school students. Homepage: www.wv.inf.tu-dresden.de/∼pascal/