A Resolution Theorem for Algebraic Domains

Pascal Hitzler
Dresden University of Technology
01062 Dresden, Germany
phitzler@inf.tu-dresden.de

Abstract

W.C. Rounds and G.-Q. Zhang have recently proposed to study a form of resolution on algebraic domains [Rounds and Zhang, 2001]. 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.

1 Introduction

Domain Theory [Abramsky and Jung, 1994] is an abstract mathematical theory for programming semantics and has grown into a respected field on the borderline between mathematics and computer science. Studying relationships between domain theory and logic is of acknowledged importance in the field. 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 [Hitzler and Seda, 200x], and the work of Rounds and Zhang on the use of domain logics for disjunctive logic programming and default reasoning [Rounds and Zhang, 2001].

The latter authors, in [Rounds and Zhang, 2001], introduced a form of clausal logic generalized to coherent algebraic domains, motivated by investigating the logical content of the Smyth powerdomain construction. In the following, 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 $T \cup \{\neg X\}$ [Robinson, 1965].

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 analogue of the theorem.

2 The Logic RZ

We assume that the reader is familiar with basic domain-theoretic notions, and we will follow the terminology from [Abramsky and Jung, 1994] and [Rounds and Zhang, 2001]. In the following, $(D, \sqsubseteq)$ will always be assumed to be a coherent algebraic cpo. We will also call these spaces *domains.* 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 A(Z)).

Definition 2.1 *Let D be a domain with set $\mathsf{K}(D)$ of compact elements. A* clause *is a finite subset of $\mathsf{K}(D)$. If X is a clause and $w \in D$, we write $w \models X$ if there exists $x \in X$ with $x \sqsubseteq w$. 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 $X \in T$. 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 $\{E\} \models X$. 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 $T \not\models \{\}$.*

Rounds and Zhang originally set out to characterize logically the notion *of Smyth powerdomain* of coherent algebraic epos. It naturally lead to the clausal logic RZ from Definition 2.1.

We can provide a sound and complete proof theory with simple rules. Consider the following rule, which we call *simplified hyperresolution.*

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

Also consider the following rules, which we call the *reduction rule* and the *extension rule.*

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

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

Theorem 2.2 *The system consisting of (shr), (ext) and (red) is complete.*

For derivations with the rules from above, we use the symbol |- as usual.

3 A Resolution Theorem

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

Definition 3.1 *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 $c = \bigsqcup \mathsf{A}(c)$.*

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

Theorem 3.2 *For any clause X we have $\mathsf{A}(X) \sim \{X\}$.*

In view of Theorem 3.2, 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 $\mathsf{A}(T) = \{\mathsf{A}(X) \mid X \in T\}$.

Theorem 3.3 *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. 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

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

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

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

It also turns out that the rules from Theorem 3.3 are sound. We investigate next a notion of negation on domains.

Definition 3.4 *An atomic domain is called an* atomic domain with negation *// there exists an involutive and Scott-continuous* negation function $^{-} : D \to D$ *with the 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 $q = \bar{p}$.

(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.

Proposition 3.5 *Let D be an atomic domain with negation. Then for all $c \in \mathsf{K}(D)$ we have $\bar{c} = \bigsqcup\{\bar{a} \mid a \in \mathsf{A}(c)\}$.*

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

Theorem 3.6 *Let D be an atomic domain with negation. Let T be a theory[1] and X be an atomic clause. Then $T \models X$ if and only if $T \cup \{\{\bar{a}\} \mid a \in X\} \vdash \{\}$.*

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

Theorem 3.7 *Let T be a theory and X be a clause. Consider $T' = \mathsf{A}(T)$. For every atomic clause $A \in \mathsf{A}(X)$ attempt to show $T' \cup \{\{\bar{a}\} \mid a \in A\} \vdash \{\}$ using* (axt) *and* (mas)*. If this succeeds, then $T \models X$. Conversely, if $T \models X$ then there exists a derivation $T' \cup \{\{\bar{a}\} \mid a \in A\} \vdash \{\}$ for each $A \in \mathsf{A}(X)$ using only the above mentioned rules.*

4 Conclusions

We have seen 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, built on a firm domain-theoretic background. Further research is being undertaken to substantiate this.

References

[Abramsky and Jung, 1994] 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.

[Hitzler and Seda, 200x] Pascal Hitzler and Anthony K. Seda. Generalized metrics and uniquely determined logic programs. *Theoretical Computer Science.* To appear.

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

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