Contexts, Concepts, and Logic of Domains

Pascal Hitzler

Artificial Intelligence Institute, Department of Computer Science
Dresden University of Technology, Dresden, Germany
phitzler@inf.tu-dresden.de, www.wv.inf.tu-dresden.de/~pascal/

Technical Report WV–02–12

Abstract

We relate two formerly independent areas: *Formal concept analysis* and *logic of domains*. We will establish a transformation of contexts into domains, and vice-versa, such that the notion of resolution in domains due to Rounds and Zhang (2001) corresponds to the construction of concepts from contexts, as in formal concept analysis. The results shed light on the use of contexts and domains for knowledge representation and reasoning purposes.

Contents

1 Introduction **1**

2 Formal Contexts and Concepts **3**

3 Clausal Logic and Resolution in Domains **3**

4 From Contexts to Domains and Back **5**

5 Attribute Logic **7**

6 Examples **9**

7 The Infinite Case **11**

8 Conclusions and Further Work **14**

1 Introduction

Domain theory was introduced in the 1970s by Scott as a foundation for programming semantics. It provides an abstract model of computation using order structures and topology, and has grown into a respected field on the borderline between Mathematics and Computer Science [AJ94, SHLG94]. Relationships between domain theory and logic were noted early on by Scott [Sco82], and subsequently developed by many authors, including Smyth [Smy89], Vickers [Vic89], Abramsky [Abr91], and Zhang [Zha91]. 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 logical aspects of knowledge representation and reasoning in artificial intelligence. Two exceptions were applications of quantitative domain theory to the semantic analysis of logic programming paradigms studied by Hitzler and Seda [Sed97, SH99, HS99, Hit01, HS0x], and the work of Rounds and Zhang on the use of domain logic for disjunctive logic programming and default reasoning [ZR97, KRZ98, ZR01, RZ01]. The latter authors developed a notion of clausal logic in coherent algebraic domains, based on considerations concerning the Smyth powerdomain, and extended it to a disjunctive logic programming paradigm [RZ01]. A notion of default negation, in the spirit of answer set programming [MT99] and Reiter's default logic [Rei80] was also added [Hit02].

The notion of *formal concept* evolved out of the philosophical theory of concepts. Wille [Wil82] proposed the main ideas which lead to the development of formal concept analysis as a mathematical field [GW99b]. The underlying philosophical rationale is that a concept is determined by its *extent*, i.e. the collection of objects which fall under this concept, and its *intent*, i.e. the collection of properties or attributes covered by this concept. Thus, a formal concept is usually distilled out of an incidence relation between a set of objects and a set of attributes, see Section 2 for details. The set of all concepts is then a complete lattice under some natural order, called a *concept lattice*.

Implicit in the construction of concept lattices from contexts is an implicational theory of attributes, e.g. the attribute "is a dog" would imply the attribute "is a mammal", to give a simple example. Thus, contexts and concepts determine logical structures, which are investigated e.g. in [GW99a, Gan99, GK99, Wil01]. So formal concept analysis contributes to data mining research, see e.g. [PT02], and likewise, contexts and concepts can be understood as natural ways of representing knowledge which bears some kind of hierarchical or logical structure.

In this paper, we establish a close relationship between the clausal logic on domains due to Rounds and Zhang [RZ01] and the construction of concepts from contexts, as in formal concept analysis [GW99b]. We will show how to obtain a domain from a context, and vice-versa, such that the construction of concepts from a context can be performed via the clausal logic of Rounds and Zhang. Due to the natural capabilities of contexts and concepts for knowledge representation, the result shows the potential of using contexts, concepts, and domain logics for knowledge representation and reasoning.

As such, the paper is part of an ongoing project on the use of domain theory in artificial intelligence, where domains shall be used for knowledge representation, and domain logic for reasoning. The contribution of this paper is on the knowledge representation aspect, more precisely on using domains for representing knowledge which is implicit in formal contexts. Aspects of reasoning, building on the clausal logic of Rounds and Zhang and its extensions, as mentioned above, are being pursued and will be presented elsewhere.

We note that on the other hand, our results may make way for the introduction of domain theory in formal concept analysis, and this issue is also to be taken up elsewhere.

The plan of the paper is as follows. In Section 2 we review the main notions from formal concept analysis which will be needed in the sequel. In Section 3 we present the clausal logic of Rounds and Zhang, in a form which will suffice for the main body of our discussion, which we will restrict to finite contexts. Section 4 establishes the transformation of contexts to domains, and vice-versa, while Section 5 will be devoted to the proof of the main result of the paper, Theorem 5.1, relating clausal logic on domains to the constrution of concepts from contexts. Two illustrating examples in Section 6 will be followed by a brief discussion, in Section 7, of infinite contexts. Conclusions and discussion of further work will close the paper in Section 8.

Acknowledgements. I thank Mike Stange for a primer on Port Royal logic, and Bernhard Ganter, Matthias Wendt, and Guo-Qiang Zhang for inspiring discussions.

2 Formal Contexts and Concepts

We introduce the notions of formal context and concept as used in formal concept analysis. We follow the standard reference [GW99b].

A (*formal*) *context* is a triple (G, M, I) consisting of two sets G and M and a relation $I \subseteq G \times M$. The elements of G are called the *objects* and the elements of M are called the *attributes* of the context. For $g \in G$ and $m \in M$ we write gIm for $(g, m) \in I$, and say that *g has the attribute m*. A context is called *clarified* if $g' = h'$ implies $g = h$ for all $g, h \in G$, and correspondingly, $m' = n'$ implies $m = n$ for all $m, n \in M$. Throughout the paper, we will assume that all contexts are clarified.

For a set $A \subseteq G$ of objects we set $A' = \{m \in M \mid gIm \text{ for all } g \in A\}$, and for a set $B \subseteq M$ of attributes we set $B' = \{g \in G \mid gIm \text{ for all } m \in B\}$. A (*formal*) *concept* of (G, M, I) is a pair (A, B) with $A \subseteq G$ and $B \subseteq M$, such that $A' = B$ and $B' = A$. We call A the *extent* and B the *intent* of the concept (A, B). For singleton sets, i.e. $B = \{b\}$, we simplify notation by writing b' instead of $\{b\}'$.

The set $\mathcal{B}(G, M, I)$ of all concepts of a given context (G, M, I) is a complete lattice with respect to the order defined by $(A_1, B_1) \leq (A_2, B_2)$ if and only if $A_1 \subseteq A_2$, which is equivalent to the condition $B_2 \subseteq B_1$. $\mathcal{B}(G, M, I)$ is called the *concept lattice* of the context (G, M, I).

2.1 Remark For every set $B \subseteq M$ of attributes we have that $B' = B'''$, so that (B', B'') is a concept. Hence, the concept lattice of a context (G, M, I) can be identified with the set $\{B'' \mid B \subseteq M\}$, ordered by superset inclusion.

3 Clausal Logic and Resolution in Domains

We introduce the clausal logic of Rounds and Zhang, together with a corresponding notion of resolution [RZ01]. We restrict our discussion to the finite case until Section 7.

A *partially ordered set* is a pair $(D, \sqsubseteq)$, where D is a nonempty set and $\sqsubseteq$ is a reflexive, antisymmetric, and transitive relation on D. A *bottom element* of a partially ordered set $(D, \sqsubseteq)$ is an element $\bot \in D$ such that $\bot \sqsubseteq d$ for all $d \in D$. For $A \subseteq D$, let $\mathsf{mub}\, A$ be the set of all minimal upper bounds of A, and we note that $\mathsf{mub}\,\emptyset = \{\bot\}$, if it exists. Dually, $\mathsf{mlb}\,A$ denotes the set of all maximal lower bounds of A. Two elements $c, d \in D$ are called *inconsistent*, written $c \not\uparrow d$, if $\mathsf{mub}\{c, d\} = \emptyset$. For $a \in D$, let $\downarrow a = \{b \in D \mid b \sqsubseteq a\}$, and dually $\uparrow a = \{b \in D \mid a \sqsubseteq b\}$. For $A \subseteq D$ define $\downarrow A = \bigcup_{a\in A}(\downarrow a)$ and $\uparrow A = \bigcup_{a\in A}(\uparrow a)$. We call A a *lower set* if $A = \downarrow A$, and we call A an *upper set* if $A = \uparrow A$. For convenience, and with a slight abuse of standard terminology, we define a *poset* to be a finite partially ordered set with bottom element. In the following, let $(D, \sqsubseteq)$ be a poset.

The following definitions can be found in [RZ01], but for the fact that we chose to introduce them for the special case of posets only. For the general case, see the discussion in Section 7.

3.1 Definition A *clause* is a subset of 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. if 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 $w \models X$. A theory T is *closed* if $T \models X$ implies $X \in T$ for all clauses X. It is called *consistent* if $T \not\models \emptyset$ or, equivalently, if there exists w with $w \models T$.

Definition 3.1 can be understood as a model theory for a clausal logic on posets. We can also present a corresponding proof theory. For this purpose, we give three rules for deriving clauses from theories. Let T be a theory. The *weakening rule*

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

should be read as follows. If X is a clause in T, $a \in X$, and $y \sqsubseteq a$, then we can derive the clause $\{y\} \cup (X \setminus \{a\})$. Similarly, the *extension rule*

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

allows to derive the clause $\{y\} \cup X$ from the clause $X \in T$, provided $y \in D$. Finally, the *simplified hyperresolution rule*

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

allows to derive the clause $\mathsf{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$ from T provided X_1 and X_2 are clauses in T, $a_1 \in X_1$, and $a_2 \in X_2$.

We write $T \vdash X$ if the clause X can be derived from the theory T by a finite number of applications of the above rules.

3.2 Theorem Let T be a theory and X be a clause. Then $T \models X$ if and only if $T \vdash X$.

Theorem 3.2 shows that the system consisting of the three rules is sound and complete with respect to the given model theory. Note that the rules given in [RZ01] differ from those above. However, equivalence can be shown, which proves Theorem 3.2, and we will do this later in Section 7 in the more general infinite case.

4 From Contexts to Domains and Back

We give next the transformation of contexts into posets mentioned in the introduction. A context (G, M, I) is transformed into a domain with elements $G \cup M$ as follows.

4.1 Definition Let $K = (G, M, I)$ be a (finite) context. Define the following relation on $D = G \cup M$:

(i) For $m_1, m_2 \in M$ let $m_1 \leq m_2$ if and only if $m_1' \supseteq m_2'$.

(ii) For $g_1, g_2 \in G$ let $g_1 \leq g_2$ if and only if $g_1' \subseteq g_2'$.

(iii) For $g \in G$ and $m \in M$ let $m \leq g$ if and only if $m \in g'$.

Finally, define the binary relation $\sqsubseteq$ on D as the transitive closure of $\leq$.

The order relation $\sqsubseteq$ in Definition 4.1 follows an intuition which is dominant in domain theory, namely that an element which is "higher" in the ordering presents an item carrying "more information" than the item further below. Thus, if g_1 and g_2 are objects, we have $g_1 \sqsubseteq g_2$ if and only if g_2 is "more special" than g_1, i.e. g_2 has more attributes than g_1. Similarly, an attribute m_2 carries "more information" than an attribute m_1, if m_2 is "more special" than m_1, i.e. if every object with attribute m_2 also has attribute m_1. Also, an object g carries "more information" than an attribute m if g has the attribute m (and possibly also others).

In the sequel, we will make the mild but convenient technical assumption that there exists some $m \in M$ with $m' = G$. This assures that there is $m \in M$ with $m \sqsubseteq d$ for all $d \in D$, i.e. m becomes the bottom element with respect to $\sqsubseteq$. Note that is also ensures that $g' \neq \emptyset$ for all $g \in G$.

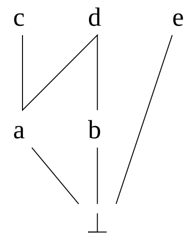
4.2 Proposition (D,$\sqsubseteq$) is a poset, and M is a lower set in D.

Proof: We have just seen that $(D, \sqsubseteq)$ has a bottom element, namely the element $m \in M$ with $m' = G$. The bottom element is uniquely determined since the context is assumed to be clarified. It is clear that $\sqsubseteq$ is reflexive and transitive. For antisymmetry assume that there are a and b in D with $a \sqsubseteq b$ and $b \sqsubseteq a$. Then either $a, b \in M$ or $a, b \in G$ by definition of $\sqsubseteq$. So let $a, b \in M$. Then $a' = b'$ by (i) of Definition 4.1, so that $a = b$ since the context is clarified. The case $a, b \in G$ is similar. Finally, M is obviously a lower set with respect to $\leq$ and therefore also for the transitive closure $\sqsubseteq$. ■

Definition 4.1 assigns to every context (G, M, I) a pair consisting of a poset $(D, \sqsubseteq)$ together with a lower set M, where $D = G \cup M$ and $\sqsubseteq$ is obtained as indicated. We denote the result of the trainsformation by $(D, M, \sqsubseteq)$ or by $\delta(G, M, I)$. More generally, we use the notation $(D, M, \sqsubseteq)$ to denote a poset D, where M is a lower set in D.

We next give the converse construction of a context from a poset.

4.3 Definition Let $(D, \sqsubseteq)$ be a poset and $M \subseteq D$ with $M = \downarrow M$. Let $G = D \setminus M$ and define the context (G, M, I) by gIm if and only if $m \sqsubseteq g$.



	a	b	⊥
c	×		×
d	×	×	×
e			×

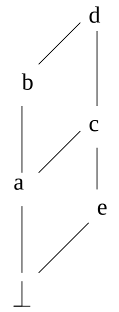


Figure 1: Counterexample for (b) of Proposition 4.4. $(D, \sqsubseteq)$ left, $(D, \sqsubseteq_1)$ right.

Definition 4.3 assigns a context to every pair consisting of a poset $(D, \sqsubseteq)$ and a lower set M. We denote the resulting context by $\kappa(D, M, \sqsubseteq)$.

We remark that Definitions 4.1 and 4.3 are not inverse to each other, as can be seen from the following Proposition. The reason for this asymmetry will be discussed later, after the proof of Theorem 5.1.

4.4 Proposition The following hold.

(a) Let (G, M, I) be a context. Then $\kappa(\delta(G, M, I)) = (G, M, I)$.

(b) Let $(D, \sqsubseteq)$ be a poset and M a lower set such that $\kappa(D, M, \sqsubseteq)$ is clarified. Then $\delta(\kappa(D, M, \sqsubseteq)) = (D, M, \sqsubseteq_1)$ is a poset and $c \sqsubseteq d$ implies $c \sqsubseteq_1 d$. However, $c \sqsubseteq_1 d$ does not necessarily imply $c \sqsubseteq d$.

(c) Under the hypotheses from (b) we have $\delta(\kappa(\delta(\kappa(D, M, \sqsubseteq)))) = \delta(\kappa(D, M, \sqsubseteq))$.

Proof: (a) Let $\delta(G, M, I) = (D, M_1, \sqsubseteq)$ and $\kappa(D, M_1, \sqsubseteq) = (G_1, M_2, I_1)$. Then trivially $G = G_1$, $M = M_1 = M_2$ and $D = G \setminus M$. Now let $g \in G$ and $m \in M$ with gIm. Then $m \sqsubseteq g$ by (iii) of Definition 4.1, and therefore gI_1m by Definition 4.3. Conversely, let $g \in G$ and $m \in M$ with gI_1m. Then $m \sqsubseteq g$ by Definition 4.3, and therefore gIm by (iii) of Definition 4.1.

(b) Let $\kappa(D, M, \sqsubseteq) = (G, M_1, I)$ and $\delta(G, M_1, I) = (D_1, M_2, \sqsubseteq_1)$, which is a poset by Proposition 4.2.

Then trivially $D = D_1$, $M = M_1 = M_2$, and $G = D \setminus M$. Now let $c, d \in D$ with $c \sqsubseteq d$. We distinguish four cases. (Case 1) If $c \in M$ and $d \in G$, then cId by Definition 4.3, and $c \sqsubseteq_1 d$ by (iii) of Definition 4.1. (Case 2) If $c, d \in M$, then $c \sqsubseteq g$ for all $g \in G$ with $d \sqsubseteq g$. Hence $g \in d'$ implies $g \in c'$ for all $g \in G$, and by (i) of Definition 4.1 we obtain $c \sqsubseteq_1 d$. (Case 3) If $c, d \in G$, then $m \sqsubseteq d$ for all $m \in M$ with $m \sqsubseteq c$. Hence $m \in c'$ implies $m \in d'$ for all $m \in M$ and by (ii) of Definition 4.1 we obtain $c \sqsubseteq_1 d$. (Case 4) The remaining case $c \in G$, $d \in M$ is impossible since M is a lower set.

For the last assertion consider $D = \{a, b, c, d, e, \perp\}$ with $a \sqsubseteq c$, $a \sqsubseteq d$, $b \sqsubseteq d$, $\perp$ the bottom element, and $M = \{a, b\}$. Then $a \not\sqsubseteq b$ but $a \sqsubseteq_1 b$, see Figure 1.

(c) $\kappa(D, M, \sqsubseteq)$ is a clarified context, thus $\kappa(\delta(\kappa(D, M, \sqsubseteq))) = \kappa(D, M, \sqsubseteq)$ by (a), which immediately yields the assertion. ∎

Note that $\delta\kappa$ can be understood as a "closure operator" on posets: It casts a poset $(D, M, \sqsubseteq)$ into one which represents a context. A clarification of $\kappa(D, M, \sqsubseteq)$ may have to be performed in this case before it is converted back into a poset.

5 Attribute Logic

We prove the main theorem of this paper, which says that the transformations from Section 4 allow to use the clausal logic of Rounds and Zhang for casting contexts into concepts. Let $(D, \sqsubseteq)$ be a poset and M a lower set. For every $\{b_1, \ldots, b_n\} = B \subseteq M$ define $\overline{B} = \{b \in M \mid \{\{b_1\}, \ldots, \{b_n\}\} \models \{b\}\}$.

5.1 Theorem Let (G, M, I) be a context and $\delta(G, M, I) = (D, M, \sqsubseteq)$. Then for every $B \subseteq M$ we have $\overline{B} = B''$.

The proof of Theorem 5.1 will be prepared by two lemmas.

5.2 Lemma Let (G, M, I) be a context, $\delta(G, M, I) = (D, M, \sqsubseteq)$, and $B \subseteq M$. Then

$$B'' = \left(\bigcap_{a \in \mathsf{mub}\, B} (\downarrow a) \right) \cap M.$$

Proof: From Definition 4.1 we obtain $m' = \uparrow m \cap G$ and $g' = \downarrow g \cap M$ for all $m \in M$ and $g \in G$. By the definitions of extent and intent it follows that

$$B' = \left(\bigcap_{b \in B} (\uparrow b) \right) \cap G = (\uparrow \mathsf{mub}\, B) \cap G$$

and also that

$$A' = \left(\bigcap_{a \in A} (\downarrow a) \right) \cap M = (\downarrow \mathsf{mlb}\, A) \cap M$$

for all $A \subseteq G$. We hence obtain

$$B'' = \left(\bigcap_{a \in (\uparrow \mathsf{mub}\, B) \cap G} (\downarrow a) \right) \cap M$$

and it remains to show that

$$\left(\bigcap_{a \in \mathsf{mub}\, B} (\downarrow a) \right) \cap M = \left(\bigcap_{a \in (\uparrow \mathsf{mub}\, B) \cap G} (\downarrow a) \right) \cap M.$$

Let

$$b \in \bigcap_{a \in \mathsf{mub}\, B} (\downarrow a),$$

i.e. $b \sqsubseteq a$ for all $a \in \mathsf{mub}\, B$. Now let $a_1 \in (\uparrow\mathsf{mub}\, B) \cap G$ be arbitrarily chosen. Then there is $a_2 \in \mathsf{mub}\, B$ with $a_2 \sqsubseteq a_1$ and hence $b \sqsubseteq a_2 \sqsubseteq a_1$. Since a_1 was chosen arbitrarily we obtain $b \sqsubseteq a$ for all $a \in (\uparrow\mathsf{mub}\, B) \cap G$, hence

$$b \in \bigcap_{a \in (\uparrow\mathsf{mub}\, B) \cap G} (\downarrow a),$$

which shows that

$$\left(\bigcap_{a \in \mathsf{mub}\, B} (\downarrow a) \right) \subseteq \left(\bigcap_{a \in (\uparrow\mathsf{mub}\, B) \cap G} (\downarrow a) \right)$$

and hence

$$\left(\bigcap_{a \in \mathsf{mub}\, B} (\downarrow a) \right) \cap M \subseteq \left(\bigcap_{a \in (\uparrow\mathsf{mub}\, B) \cap G} (\downarrow a) \right) \cap M.$$

Conversely, let

$$b \in \left(\bigcap_{a \in (\uparrow\mathsf{mub}\, B) \cap G} (\downarrow a) \right) \cap M$$

and let $\mathsf{mub}\, B = \{m_1, \ldots, m_n, g_1, \ldots, g_m\}$ with $m_i \in M$ for all $i = 1, \ldots, n$ and $g_j \in G$ for all $j = 1, \ldots, m$. It remains to show that (1) $b \sqsubseteq g_j$ for all $j = 1, \ldots, m$ and (2) $b \sqsubseteq m_i$ for all $i = 1, \ldots, n$, because this implies

$$b \in \left(\bigcap_{a \in \mathsf{mub}\, B} (\downarrow a) \right) \cap M.$$

Since $b \sqsubseteq a$ for all $a \in (\uparrow\mathsf{mub}\, B) \cap G$ and $g_j \in (\uparrow\mathsf{mub}\, B) \cap G$ we obtain $b \sqsubseteq g_j$ for all $j = 1, \ldots, m$, which shows (1). For (2), let $m \in \{m_1, \ldots, m_n\}$. By hypothesis we have $b \sqsubseteq g$ for all $g \in \uparrow m \cap G = m'$. Since $b \in M$ we obtain $b' \supseteq m'$ and therefore $b \sqsubseteq m$ by (i) of Definition 4.1. ■

5.3 Lemma Let D be a poset, let $a, a_1, \ldots, a_n \in D$ and note the convention $\bigcap_{b \in \emptyset} (\downarrow b) = D$. Then the following hold.

(a) $\{\{a_1, \ldots, a_n\}\} \models \{a\}$ if and only if $a \in \bigcap_{i=1}^{n} (\downarrow a_i)$.

(b) $\{\{a_1\}, \ldots, \{a_n\}\} \models \{a\}$ if and only if $a \in \bigcap_{b \in \mathsf{mub}\{a_1, \ldots, a_n\}} (\downarrow b)$.

Proof: For the proof, we repeatedly apply Theorem 3.2.

(a) Let $a \in \bigcap_{i=1}^{n} (\downarrow a_i)$. Then $a \sqsubseteq a_1$ and we obtain $\{\{a_1, \ldots, a_n\}\} \vdash \{a, a_2, \ldots, a_n\}$ using the weakening rule. By applying the same rule again to $\{\{a, a_2, \ldots, a_n\}\}$ with $a \sqsubseteq a_2$ we have $\{\{a_1, \ldots, a_n\}\} \vdash \{a, a, a_3 \ldots, a_n\} = \{a, a_3 \ldots, a_n\}$, and by repeated application of the weakening rule we eventually obtain $\{\{a_1, \ldots, a_n\}\} \vdash \{a\}$.

Conversely, let a be such that $\{\{a_1,\dots,a_n\}\} \models \{a\}$. Since we derive $\{a\}$ from the theory $\{\{a_1,\dots,a_n\}\}$ consisting of a single clause, the derivation must be possible using only the weakening and the extension rule. Now assume $a \notin {\downarrow} a_k$ for some $k \in \{1,\dots,n\}$. Then it is an easy proof by induction that every clause derived from $\{a_1,\dots,a_n\}$ must contain either a_k or some b with $b \sqsubseteq a_k$. Since $a \not\sqsubseteq a_k$ it is impossible to derive the clause $\{a\}$.

(b) Let $\mathsf{mub}\{a_1,\dots,a_n\} = \{b_1,\dots,b_m\}$. If $a \in \bigcap_{b\in\mathsf{mub}\{a_1,\dots,a_n\}} (\downarrow b) = \bigcap_{i=1}^{m} (\downarrow b_i)$, then $\{\{b_1,\dots,b_m\}\} \models \{a\}$ by (a). An easy induction argument using (shr) shows that $\{\{a_1\},\dots,\{a_n\}\} \vdash \{b_1,\dots,b_m\}$, so we obtain $\{\{a_1\},\dots,\{a_n\}\} \models \{a\}$.

Conversely, assume $\{\{a_1\},\dots,\{a_n\}\} \models \{a\}$. Then for every $w \in D$ with $w \models \{a_1\},\dots,w \models \{a_n\}$ we have $w \models \{a\}$, i.e. whenever $a_1,\dots,a_n \sqsubseteq w$ then $a \sqsubseteq w$. So let $b \in \mathsf{mub}\{a_1,\dots,a_n\}$. Then $a_1,\dots,a_n \sqsubseteq b$, and hence $a \sqsubseteq b$. Since $b \in \mathsf{mub}\{a_1,\dots,a_n\}$ can be chosen arbitrarily, we obtain $a \in \bigcap_{b\in\mathsf{mub}\{a_1,\dots,a_n\}} ({\downarrow} b)$ as desired. ■

We can now prove Theorem 5.1.

Proof of Theorem 5.1 Let $B = \{b_1,\dots,b_n\}$ and let $b \in \overline{B}$, i.e. $\{\{b_1,\dots,b_n\}\} \models \{b\}$ and $b \in M$. Then by Lemma 5.3 (b) we have $b \in \bigcap_{a\in\mathsf{mub}\,B} ({\downarrow} a)$, and since also $b \in M$ we can apply Lemma 5.2 to obtain $b \in B''$.

Conversely, let $b \in B''$. Then $b \in \bigcap_{a\in\mathsf{mub}\,B} ({\downarrow} a) \cap M$ by Lemma 5.2, hence $b \in M$ and $b \in \bigcap_{a\in\mathsf{mub}\,B} ({\downarrow} a)$, which implies $\{\{b_1\},\dots,\{b_n\}\} \models \{b\}$ by Lemma 5.3. So $b \in \overline{B}$ as desired. ■

5.4 Corollary $\left(\left\{\overline{B} \mid B \subseteq M\right\}, \supseteq\right)$ is isomorphic to the concept lattice of (G, M, I).

Proof: By Theorem 5.1, we have $\overline{B} = B''$, and the assertion follows from Remark 2.1. ■

We return to the remark made earlier in the paragraph preceding Proposition 4.4.

5.5 Remark It does not suffice to replace Definition 4.1 by the simpler definition

for $g \in G$ and $m \in M$ let $m \sqsubseteq g$ if and only if $m \in g'$,

omitting parts (i) and (ii) of Definition 4.1, even if, additionally, a bottom element is added to the resulting order $\sqsubseteq$ to make it a poset. Indeed, Theorem 5.1 does no longer hold in this case: Consider again the poset on the left hand side of Figure 1, which can be understood to represent the context $(\{c,d,e\},\{a,b\},\{(c,a),(d,a),(d,b)\})$, after a bottom element is added. We obtain $a \notin \overline{\{b\}} = \{b,\bot\}$ although $a \in b'' = \{a,b\}$.

6 Examples

We discuss two examples to display our results. The first one is taken from [GW99b, Section 2.2]. Seven triangles are classified according to some properties, as in Table 1. We have added an attribute t to satisfy our technical condition on contexts. The corresponding poset and

Triangle	a	b	c	d	e	f	g	t
1: $(0,0),(6,0),(3,1)$		×	×	×		×		×
2: $(0,0),(1,0),(0,1)$		×	×				×	×
3: $(0,0),(4,0),(1,2)$		×		×	×			×
4: $(0,0),(2,0),(1,\sqrt{3})$	×		×	×	×			×
5: $(0,0),(2,0),(5,1)$		×		×		×		×
6: $(0,0),(2,0),(1,3)$	×	×	×	×				×
7: $(0,0),(2,0),(0,1)$		×					×	×

Table 1: A context for triangles. a: equilateral, b: not equilateral, c: isosceles, d: oblique, e: acute, f: obtues, g: right, t: triangle. The pairs in the first column are coordinates of the vertices of the triangles.

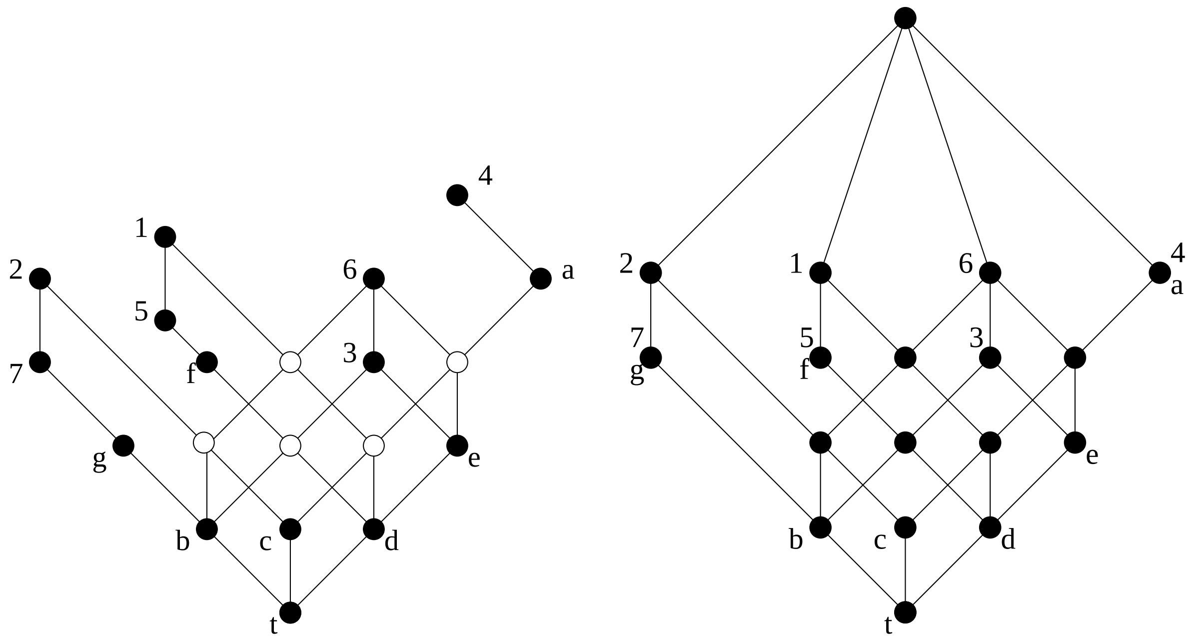


Figure 2: Poset and concept lattice (the latter reversely ordered) for the triangles context. The black dots are elements, while the circles stand for intersections of lines, e.g. on the left hand side we have $b \sqsubseteq f$ and $b \sqsubseteq 3$.

concept lattice are depicted in Figure 2, where the order in the concept lattice is reverse to the usual one.

The labels in Figure 2 are determined as follows. A concept is labelled with an object g if this concept is the least concept (in the usual ordering on concepts) which contains g. A concept is labelled with an attribute m if this concept is the greatest concept (in the usual ordering on concepts) which contains m. We note the similarity between the two pictures in Figure 2, which is not coincidental, as the following theorem shows.

6.1 Theorem Let (G, M, I) be a clarified context, let $\delta(G, M, I) = (D, M, \sqsubseteq)$, and let $(C, \leq)$ be the concept lattice of (G, M, I). We define a mapping $f : D \to C$ as follows. For $m \in M$ let $f(m)$ be the greatest element $(A, B) \in C$ such that $m \in B$. For $g \in G$ let $f(g)$ be the least element $(A, B) \in C$ such that $g \in A$. Then f is an order-homomorphism from $(D, \sqsubseteq)$ to $(C, \geq)$.

Proof: First note that by [GW99b, Proposition 24], and its dual, the mapping f is well-defined.

We show that f is an order-homomorphism, i.e. for all $c, d \in D$ with $c \sqsubseteq d$ we have $f(c) \geq f(d)$. Let $m_1, m_2 \in M$ and $g_1, g_2 \in G$. If $m_1 \sqsubseteq m_2$, then $m_2' \subseteq m_1'$. Since m_2 is contained in $f(m_2)$, we obtain that for all g in $f(m_2)$ we have $g \in m_1'$, and therefore m_1 must also be contained in $f(m_2)$. By definition, $f(m_1)$ is the greatest concept containing m_1, hence $f(m_1) \geq f(m_2)$. If $g_1 \sqsubseteq g_2$, then $g_1' \subseteq g_2'$. Since g_1 is contained in $f(g_1)$, we obtain that for all m in $f(g_1)$ we have $m \in g_2'$, and therefore g_2 must also be contained in $f(g_1)$. By definition, $f(g_2)$ is the least concept containing g_2, hence $f(g_1) \geq f(g_2)$. Finally, let $m_1 \sqsubseteq g_1$. Then $m_1 \in g_1'$, so the concept (g_1'', g_1') contains both g_1 and m_1. But $f(m_1)$ is the greatest concept containing m_1, so $f(m_1) \geq (g_1'', g_1')$. Also, $f(g_1)$ is the least concept containing g_1, so $(g_1'', g_1') \geq f(g_1)$, and we obtain $f(m_1) \geq f(g_1)$. ■

By the triangle example, we see that in general f is not injective. We also note that in the same example the intent B'' of any set of attributes B is obtained as the set $\downarrow B$ with respect to the poset. This, however, is not the case in general, as the next example shows.

Consider the context in Table 2 which classifies animals according to some attributes. The corresponding poset is depicted in Figure 3, and we note that, for example, $\{b, m\}'' = \{b, m, l, a\} \neq \{b, m, a\} = \downarrow\{b, m\}$. As the reader will easily verify, the concept lattice of the animal context is isomorphic to its poset, reversely ordered and with a bottom element added, and the labels in Figure 3, in this case, are the respective images of the mapping f from Theorem 6.1.

7 The Infinite Case

We shortly discuss infinite contexts and partially ordered sets, and deliver the missing proof of Theorem 3.2.

A *complete partial order*, *cpo* for short, is a partially ordered set $(D, \sqsubseteq)$ with a least element, $\perp$, 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

	b	m	f	l	a
s	×		×		×
d		×	×		×
w	×	×		×	×
t			×	×	×

Table 2: A context for animals. s: some dinosour, d: dog, w: whale, t: toad, b: is big, m: is a mammal, f: has feet, l: lives in water, a: is an animal

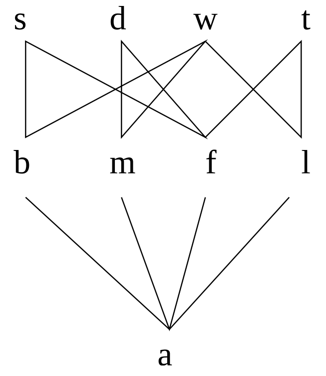


Figure 3: Poset for the animal context.

elements of a cpo D is written as $\mathsf{K}(D)$. An *algebraic cpo* is a cpo such that every $e \in D$ is the directed supremum of all compact elements below it. A *coherent algebraic cpo* is an algebraic cpo such that every finite set of compact elements has a finite set of minimal upper bounds.

Note that in a poset all elements are compact, and that every poset is trivially a coherent algebraic domain. We next carry over the definitions from Section 3 from posets to domains. The reader will easily verify that the new terminology is consistent with the old one.

Define a *clause* in a domain D to be a finite subset of $\mathsf{K}(D)$. The rest of Definition 3.1 remains unchanged. Likewise, the weakening rule and the simplified resolution rule need not be modified. The extension rule becomes

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

with the sole modification that y is required to be compact.

In [RZ01], Rounds and Zhang give the following two rules, called the *binary hyperresolution rule* and the *special rule*[1].

$$\frac{X_1, X_2 \in T; \qquad a_1 \in X_1 \quad a_2 \in X_2; \qquad \mathsf{mub}\{a_1, a_2\} \models Y}{Y \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})} \quad \textsf{(bhr)}$$

$$\frac{\emptyset \in T; \qquad Y \text{ a clause}}{Y} \quad \textsf{(spec)}$$

[1]They also give a third rule which can be disregarded for our discussion.

Due to [RZ01], the system consisting of the binary hyperresolution rule and the special rule is sound and complete with respect to the model theory introduced in Definition 3.1. We will use this result for proving Theorem 3.2.

Proof of Theorem 3.2 For soundness, consider the weakening rule. Then $\mathsf{mub}\{a\} = \{a\} \models \{y\}$ and therefore, by the binary hyperresolution rule, the weakening rule is sound. Now consider the extension rule. If $X = \emptyset$, then the rule is sound by the special rule. If $X \neq \emptyset$, then let $a \in X$ and note that $\mathsf{mub}\{a\} = \{a\} \models \{a, y\}$ for all $y \in \mathsf{K}(D)$, so again by the binary hyperresolution rule, the extension rule is sound. Finally, consider the simplified hyperresolution rule and note that trivially $\mathsf{mub}\{a_1, a_2\} \models \mathsf{mub}\{a_1, a_2\}$, so that by the binary hyperresolution rule, the simplified hyperresolution rule is sound.

For completeness, we note that the special rule can easily be derived from the extension rule. So it suffices to derive the binary hyperresolution rule from (shr), (ext) and (w). Let X_1, X_2 be given with $a_1 \in X_1$, $a_2 \in X_2$, and $a_1 \uparrow a_2$. Furthermore, let Y be a clause with $\mathsf{mub}\{a_1, a_2\} \models Y$ and let $\mathsf{mub}\{a_1, a_2\} = \{b_1, \dots, b_n\}$. Then for every b_i there exists $y_i \in Y$ with $y_i \sqsubseteq b_i$. From X_1 and X_2, using the simplified hyperresolution rule, we can derive $X_3 = \mathsf{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$, and by repeated application of the weakening rule and the extension rule we obtain $X_4 = \{y_1, \dots, y_n\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$. Finally, using the extension rule repeatedly, we can add to X_4 all remaining elements from Y. For $a_1 \not\uparrow a_2$ we have $\mathsf{mub}\{a_1, a_2\} = \emptyset$ and we derive $(X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$ using the simplified hyperresolution rule. Now the extension rule can be applied repeatedly in order to obtain $Y \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})$. This completes the proof. ■

As to the problem of carrying over Theorem 5.1 to the infinite case, we note that not every context which is cast into a partially ordered set using Definition 4.1 results in a coherent algebraic domain, as the following example shows.

7.1 Example Let $G = \{g, h, g_1, g_2, g_3, \dots\}$ and $M = \{m, n, m_1, m_2, m_3, \dots\}$. Define $I \subseteq G \times M$ as $I = \{(g, m), (h, n), (g_i, m), (g_i, n), (g_i, m_i) \mid i \in \mathbb{N}\}$, and let $(D, \sqsubseteq)$ be as in Definition 4.1. Then $\mathsf{mub}\{g, h\} = \{g_1, g_2, g_3, \dots\}$ is infinite. So $(D, \sqsubseteq)$ cannot be a coherent algebraic domain.

Example 7.1 shows that the results in this paper can not carry over to infinite contexts and domains in full generality without modifications. Further investigations are needed in order to understand the infinite case. This is ongoing work of the author and will be presented elsewhere. However, we note that the following theorem holds. Call a context which results in a coherent algebraic domain a *finitary context.*

7.2 Theorem Let (G, M, I) be a finitary context. Then Theorem 5.1 carries over analogously to this case.

Proof: It is easily verified that the proof of Theorem 5.1 carries over with only minor modifications. ■

8 Conclusions and Further Work

We have established results which lay the foundations for a cross-transfer between formal concept analysis and domain logic. Apart from the obvious task of extending Theorem 5.1 to the infinite case, we see two major lines of research emerging from the investigations. The first focuses on applications in formal concept analysis and data mining. How can tools and results from domain theory and domain logics be employed? The second concerns the project mentioned in the introduction, with the aim of using domains for knowledge representation and domain logics for reasoning. Theorem 5.1 establishes a first step towards understanding domains in a concept analysis and data mining framework, thus contributing substantially to the question how domains represent knowledge. It also makes it possible to understand the clausal logic of Rounds and Zhang, and its extensions mentioned in the introduction, as a reasoning framework which acts directly on contexts. Understanding the impact of these observations constitutes a considerable body of work, and is under investigation by the author.

References

[Abr91] Samson Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[AJ94] 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.

[Gan99] Bernhard Ganter. Attribute exploration with background knowledge. *Theoretical Computer Science*, 217:215–233, 1999.

[GK99] Bernhard Ganter and Rüdiger Krauße. Pseudo models and propositional Horn inference, 1999.

[GW99a] Bernhard Ganter and Rudolf Wille. Contextual attribute logic. In William M. Tepfenhart and Walling R. Cyre, editors, *Conceptual Structures: Standards and Practices. Proceedings of the 7th International Conference on Conceptual Structures, ICCS '99, July 1999, Blacksburgh, Virginia, USA*, volume 1640 of *Lecture Notes in Artificial Intelligence*, pages 377–388. Springer, Berlin, 1999.

[GW99b] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis — Mathematical Foundations.* Springer, Berlin, 1999.

[Hit01] Pascal Hitzler. *Generalized Metrics and Topology in Logic Programming Semantics.* PhD thesis, Department of Mathematics, National University of Ireland, University College Cork, 2001.

[Hit02] Pascal Hitzler. Towards nonmonotonic reasoning on hierarchical knowledge. Submitted to the Workshop Logische Programmierung, WLP02, Dresden, 2002.

[HS99] Pascal Hitzler and Anthony K. Seda. Some issues concerning fixed points in computational logic: Quasi-metrics, multivalued mappings and the Knaster-Tarski theorem. In *Proceedings of the 14th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, New York*, volume 24 of *Topology Proceedings*, pages 223–250, 1999.

[HS0x] Pascal Hitzler and Anthony K. Seda. Generalized metrics and uniquely determined logic programs. *Theoretical Computer Science*, 200x. To appear.

[KRZ98] Eric Klavins, William C. Rounds, and Guo-Qiang Zhang. Experimenting with power default reasoning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 1998, Madison, Wisconsin, USA*, pages 846–852. AAAI Press / The MIT Press, 1998.

[MT99] V. Wiktor Marek and Miroslav Truszczyński. Stable models and an alternative logic programming paradigm. In Krzysztof R. Apt, V. Wiktor Marek, Miroslav Truszczyński, and David S. Warren, editors, *The Logic Programming Paradigm: A 25-Year Persepective*, pages 375–398. Springer, Berlin, 1999.

[PT02] John L. Pfaltz and Christopher M. Taylor. Closed set mining of biological data. In Mohammed J. Zahi, Jason T. L. Wang, and Hannu T. T. Toivonen, editors, *Proceedings of the 2nd ACM SIGKDD Workshop on Data Mining in Bioinformatics, July 2002, Edmonton, Alberta, Canada*, 2002. http://www.cs.rpi.edu/∼zaki/BIOKDD02/.

[Rei80] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[RZ01] William C. Rounds and Guo-Qiang Zhang. Clausal logic and logic programming in algebraic domains. *Information and Computation*, 171(2):156–182, 2001.

[Sco82] 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.

[Sed97] Anthony K. Seda. Quasi-metrics and the semantics of logic programs. *Fundamenta Informaticae*, 29(1):97–117, 1997.

[SH99] Anthony K. Seda and Pascal Hitzler. Topology and iterates in computational logic. In *Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997*, volume 22 of *Topology Proceedings*, pages 427–469, 1999.

[SHLG94] Viggo Stoltenberg-Hansen, Ingrid Lindström, and Edward R. Griffor. *Mathematical Theory of Domains*. Cambridge University Press, 1994.

[Smy89] 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.

[Vic89] Steven Vickers. *Topology via Logic.* Cambridge University Press, Cambridge, UK, 1989.

[Wil82] Rudolf Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.

[Wil01] Rudolf Wille. Boolean judgement logic. In Harry Delugach and Gerd Stumme, editors, *Conceptual Structures: Broadening the Base, Proceedings of the 9th International Conference on Conceptual Structures, ICCS 2001, July 2001, Stanford, LA, USA*, volume 2120 of *Lecture Notes in Artificial Intelligence*, pages 115–128. Springer, Berlin, 2001.

[Zha91] Guo-Qiang Zhang. *Logic of Domains.* Birkhauser, Boston, 1991.

[ZR97] Guo-Qiang Zhang and William C. Rounds. Complexity of power default reasoning. In *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science, LICS'97, Warsaw, Poland*, pages 328–339. IEEE Computer Society Press, 1997.

[ZR01] Quo-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.