A Cartesian Closed Category of Approximable Concept Structures

Pascal Hitzler[1]* and Guo-Qiang Zhang[2]**

[1] 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/

[2] Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, Ohio, U.S.A.
gqz@eecs.case.edu; newton@case.edu

Abstract. Infinite contexts and their corresponding lattices are of theoretical and practical interest since they may offer connections with and insights from other mathematical structures which are normally not restricted to the finite cases. In this paper we establish a systematic connection between formal concept analysis and domain theory as a categorical equivalence, enriching the link between the two areas as outlined in [25]. Building on a new notion of *approximable concept* introduced by Zhang and Shen [26], this paper provides an appropriate notion of morphisms on formal contexts and shows that the resulting category is equivalent to (a) the category of complete algebraic lattices and Scott continuous functions, and (b) a category of information systems and approximable mappings. Since the latter categories are cartesian closed, we obtain a cartesian closed category of formal contexts that respects both the context structures as well as the intrinsic notion of approximable concepts at the same time.

1 Introduction

Formal concept analyais (FCA [1]) is a powerful lattice-based tool for symbolic data analysis. In essence, it is based on the extraction of a lattice — called *formal concept lattice* — from a binary relation called *formal context* consisting of a set of objects, a set of attributes, and an incidence relation. The transformation from a two-dimensional incidence table to a lattice structure is a crucial *paradigm shift* from which FCA derives much of its power and versatility as a modeling tool. The concept lattices obtained this way turn out to be exactly the complete lattices, and the particular way in which they structure and represent knowledge is very appealing and natural from the perspective of many scientific disciplines.

* The first author was supported by a Postdoc-fellowship of the German Academic Exchange Service (DAAD); this work was carried out while the author was visiting the second author at the Department of EECS, Case Western Reserve University[2].
** Corresponding author.

K.E. Wolff et al. (Eds.): ICCS 2004, LNAI 3127, pp. 170–185, 2004.

The successful applications of FCA, however, are mainly restricted to finite contexts or finite concept lattices, thus neglecting the full power of the theory. Infinite contexts and their corresponding lattices are of theoretical and practical interest since they may offer connections with and insights from other mathematical structures which are normally not restricted to the finite cases. In this paper we establish a systematic connection between formal concept analysis and domain theory as a categorical equivalence, enriching the link between the two areas as outlined in [25]. Domain theory is a subject of extensive study in theoretical computer science and programming languages. Its basic idea of partial information and successive approximation suggests that for infinite structures to be computationally feasible, items of knowledge or information should either be finitely representable or approximable by such finitely representable items. This idea motivated the introduction of a new notion called *approximable concept*, reported by Zhang and Shen in a separate paper [26]. Approximable concept lattices derived from this new notion are exactly the complete algebraic ones; and every (classical) formal concept is approximable. Furthermore, in cases where the formal contexts are finite, approximable concepts and formal concepts coincide. From a categorical viewpoint, this establishes a relation at the object level as a part of a functor; a main contribution of this paper is the introduction of an appropriate notion of morphisms on formal contexts and the proof that the following three categories are equivalent in the categorical sense [13]:

1. the category of formal contexts and context morphisms **Cxt** introduced here for the first time,
2. the category of complete algebraic lattices and Scott continuous functions, and
3. the category of information systems and approximable mappings **ISys**.

This implies that the category of formal contexts and context morphisms is cartesian closed, and as a result a rich collection of constructions including product and function space is immediately made possible.

Our paper can be viewed as part of a unique research program [8,25,26, 27,19] exploiting the synergies among the following recurring themes in several independently developing and yet somewhat related areas:

- functional dependency $X \to Y$ in databases,
- association rules $X \Rightarrow Y$ in data-mining,
- consequence relation $\wedge X \models \wedge Y$ in logic,
- entailment relation $X \vdash Y$ in information systems and domains,
- intention-extension duality $Y \subseteq (X)''$ in formal concept analysis.

Note that in both logic as well as in domain theory, X is restricted to finite sets while in databases and data-mining both X and Y are restricted to finite cases due to the pragmatic motivations of the areas. In classical formal concept analysis, there is no size constraint on either X or Y even when the formal contexts are infinite. These potential mismatches (or, the alignment of them) turn out to have important consequences, as will be seen from the rest of this paper.

The rest of the paper is structured as follows. In Section 2 we introduce basic notions from formal concept analysis, domain theory, and category theory. In Section 3, results on approximable concepts are provided. In Section 4 we introduce an appropriate notion called *context morphism* on formal contexts and show that formal contexts with context morphisms constitute a category **Cxt**. In Section 5 we recall the category **ISys** of Scott information systems with trivial consistency predicate. This category is known to be equivalent to the category of complete algebraic lattices with Scott continuous functions. We then introduce the functors which will be used in our equivalence proof, which will be carried out in Section 6. Some categorical constructions in Section 7 complete the technical contributions of the paper. The last section gives some concluding remarks.

2 Background

We review some necessary background in FCA, domain theory, and category theory in order to fix notations.

Formal Contexts and Concepts

Our main reference for formal concept analysis (FCA) is [1]. In places, we will follow the notation used in [26], because it is more convenient for our purposes.

Definition 1. *A* formal context *P is a triple $(P_o, P_a, \models)$, where P_o is a set of* objects*, P_a a set of* attributes*, and $\models$ a binary* satisfaction relation $\models \subseteq P_o \times P_a$. *We also define the following mappings.*

$$\alpha_P = \alpha : 2^{P_o} \to 2^{P_a} \text{ with } X \mapsto \{a \mid \forall x \in X,\, x \models a\}$$
$$\omega_P = \omega : 2^{P_a} \to 2^{P_o} \text{ with } Y \mapsto \{o \mid \forall y \in Y,\, o \models y\}$$

A subset $A \subseteq P_a$ is called an (intent-) concept *if $\alpha(\omega(A)) = A$.*

For readers from the traditional FCA community, it should be helpful to note that P_o corresponds to G, P_a corresponds to M, and $\models$ to I using the standard notation of a context (G, M, I). Also note that $(\quad)' = \alpha$ and $(\quad)' = \omega$ since the standard notation ignores the types of the two operators. One can also regard $(\quad)'$ as, informally, an "infix" notation and α, ω as prefix ones.

The following is a central result for FCA, a proof of which can be found in [1]. Recall that a *complete lattice* is a partial order in which all (possibly infinite) suprema (a.k.a. join) and infima (a.k.a. meet) exist.

Theorem 1 (Wille [24]). *The set of all (intent-) concepts of a formal context P, ordered under subset inclusion, is a complete lattice, called the* concept lattice *of P. Furthermore, every complete lattice $(D, \leq)$ is isomorphic to the concept lattice L of the formal context $(D, D, \leq)$, with isomorphism $\iota : D \to L$ given by $d \mapsto \alpha(\omega(\{d\}))$.*

Domain Theory

Domain theory was introduced by Scott in the late 60s for the denotational semantics of programming languages. It provides a theoretically elegant and practically useful mathematical foundation for the design, definition, and implementation of programming languages, and for systems for the specification and verification of programs. The basic idea of domain theory is partial information and successive approximation, captured in *complete partial orders* (cpos). Functions acting on cpos are those which preserve the limits of directed sets – this is the so-called continuity property. If one thinks of directed sets as an approximating schema for infinite objects, then members of a directed set can be thought of as finite approximations. Continuity makes sure that infinite objects can be approximated by finite computations. An important property of continuous functions is that when ordered in appropriate ways, they form a complete partial order again. Thus a continuous function becomes once again an object in a complete partial order. This seamless and uniform treatment of a higher-order object just as an ordinary object is the hallmark of domain theory.

Let $(D, \sqsubseteq)$ be a partial order. A subset X of D is *directed* if it is non-empty and for each pair of elements $a, b \in X$, there is an upper bound $x \in X$ for $\{a, b\}$. A *complete partial order* (cpo) is a partial order $(D, \sqsubseteq)$ with a least element $(\bot)$ and every directed subset X has a least upper bound (or join) $\bigsqcup X$. A complete lattice is a partial order in which any subset has a join (this implies that any subset will also have a meet – greatest lower bound). Compact elements of a cpo $(D, \sqsubseteq)$ are those inaccessible by directed sets: $a \in D$ is *compact* if for any directed set X of D, $a \sqsubseteq \bigsqcup X$ implies that there exists $x \in X$ with $a \sqsubseteq x$. A cpo is *algebraic* if every element is the join of a directed set of compact elements. A set $X \subseteq D$ is *bounded* if it has an upper bound. A cpo is *bounded complete* if every bounded set has a join. Scott domains are, by definition, bounded complete algebraic cpos.

Category Theory

Category theory provides a unified language for managing *conceptual complexity in mathematics*. The importance of category theory to computer science is manifested in its ability in guiding research to the discovery of categorically natural, but otherwise non-obvious missing entities in a conceptual picture.

Our category-theoretical terminology follows [5]. A *category* $\mathbf{C}$ consists of

(i) a class $|\mathbf{C}|$ of *objects* of the category,
(ii) for all $A, B \in |\mathbf{C}|$, a set $\mathbf{C}(A, B)$ of *morphisms* from A to B,
(iii) for all $A, B, C \in |\mathbf{C}|$, a composition operation $\circ : \mathbf{C}(B, C) \times \mathbf{C}(A, B) \to \mathbf{C}(A, C)$,
(iv) for all $A \in |\mathbf{C}|$, an *identity morphism* $\mathrm{id}_A \in \mathbf{C}(A, A)$,

such that for all $f \in \mathbf{C}(A, B)$, $g \in \mathbf{C}(B, C)$, $h \in \mathbf{C}(C, D)$, the associativity axiom $h \circ (g \circ f) = (h \circ g) \circ f$ and the identity axioms $\mathrm{id}_B \circ f = f$ and $g \circ \mathrm{id}_B = g$ are satisfied. As usual, we write $f : A \to B$ for morphisms $f \in \mathbf{C}(A, B)$.

A *functor* $\mathbf{F}$ from a category $\mathbf{A}$ to a category $\mathbf{B}$ consists of

(i) a mapping $|\mathbf{A}| \to |\mathbf{B}|$ of objects, where the image of an object $A \in |\mathbf{A}|$ is dentoted by $\mathbf{F}A$,
(ii) for all $A, A' \in |\mathbf{A}|$, a mapping $\mathbf{A}(A, A') \to \mathbf{B}(\mathbf{F}A, \mathbf{F}A')$, where the image of a morphism $f \in \mathbf{A}(A, A')$ is denoted by $\mathbf{F}f$,

such that for all $A, B, C \in |\mathbf{A}|$ and all $f \in \mathbf{A}(A, B)$ and $g \in \mathbf{A}(B, C)$ we have $\mathbf{F}(f \circ g) = \mathbf{F}f \circ \mathbf{F}g$ and $\mathbf{F}\mathrm{id}_A = \mathrm{id}_{\mathbf{F}A}$.

3 Approximable Concepts

The defining property of a formal concept given by the equality $\alpha(\omega(A)) = A$ is computationally feasible for finite contexts, but lends itself for alternative formulation in the infinite case. From a domain-theoretic perspective, a computationally accessible infinite object is one that can be approximated by partial, finitary objects. If we replace "object" by formal concept, and "finitary objects" by finitely generated concepts (i.e., $\alpha(\omega(A))$ for finite A), then we obtain the following definition, introduced in [26].

Definition 2. *Given a set A, let $\mathsf{Fin}(A)$ denote the set of finite subsets of A. With notation fixed in Definition 1, a subset $A \subseteq P_a$ is called an* approximable (intent-) concept *if for every $X \in \mathsf{Fin}(A)$, we have $\alpha(\omega(X)) \subseteq A$.*

As a consequence, every approximable concept A is the limit (i.e. least upper bound) of a directed set of finitely generated concepts below it:

$$A = \bigcup \{\alpha(\omega(X)) \mid X \in \mathsf{Fin}(A)\}.$$

The notion of approximable concept is a natural one from a logical point of view, in that approximable concepts correspond to *theories*. Informally, a (logical) theory is a set of formulas closed under (a predefined notion) of entailment. A basic notion of entailment can be extracted from a context by letting $X \vdash a$ just when $a \in \alpha(\omega(X))$. This has been observed by many researchers and investigated at a more systematic level in [25] by relating it to information systems [18]. The relation $\vdash$ corresponds to an *association rule* in data mining and an instance of *functional dependence* in databases. If we build theories by taking attributes as atomic propositional formulas and the corresponding $\vdash$ as the entailment relation, then theories coincide with approximable concepts. The distinction is that in classical formal concept analysis, an infinite set X is allowed in the entailment $X \vdash a$ while in approximable concept analysis, only finite Xs are allowed. It is well-known in logic that an infinite conjunction p in the antecedent of an entailment $p \to q$ destroys compactness. For readers interested in discussions along this line, we refer to [23] for general and intuitive examples, and to [10] for related hardcore theory.

We now summarize relevant results from [26]. Recall that a complete lattice is called *algebraic* if each of its element is the supremum of the directed set of compact elements below it. Given a complete algebraic lattice $(D, \leq)$, let $\mathsf{K}(D)$ denote the set of all compact elements of D.

Theorem 2 (Zhang and Shen [26]). *For any formal context P, the set of its approximable (intent-) concepts $\mathcal{A}P$ under set-inclusion forms a complete algebraic lattice. Conversely, every complete algebraic lattice $(D, \leq)$ is order-isomorphic to $\mathcal{A}P$, where $P = (D, \mathsf{K}(D), \leq)$. An isomorphism in this case is given by $K \mapsto \sup K$ for any approximable concept K.*

The supremum just mentioned exists since approximable concepts in this case are exactly the ideals (i.e. downward closed directed subsets) of D.

It is easy to see that with respect to finite contexts, approximable concepts are just the standard ones. In the infinite case, although every standard concept is approximable, not all approximable concepts are concepts in the standard sense. This gives the impression that approximable concepts are more general; but interestingly they are less general in terms of the lattices they represent collectively: the corresponding lattices built from approximable concepts are of a restricted kind – the algebraic, complete lattices – instead of complete lattices in general. This again fits the domain-theoretic paradigm in that a general approximating scheme should be part of a computable mathematical structure. We refer to [26] for an example (necessarily infinite) to illustrate the differences.

4 Cxt: A Category of Formal Contexts

We introduce a new notion of morphism on formal contexts.

Definition 3 (context morphism). *Given formal contexts $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$, a* context morphism *$\rightarrowtriangle_{PQ} = \rightarrowtriangle$ from P to Q is a relation $\rightarrowtriangle \subseteq \mathsf{Fin}(P_a) \times \mathsf{Fin}(Q_a)$, such that the following conditions are satisfied for all $X, X', Y_1, Y_2 \in \mathsf{Fin}(P_a)$ and $Y, Y' \in \mathsf{Fin}(Q_a)$:*

(cm1) $\emptyset \rightarrowtriangle \emptyset$,
(cm2) $X \rightarrowtriangle Y_1$ *and* $X \rightarrowtriangle Y_2$ *imply* $X \rightarrowtriangle Y_1 \cup Y_2$,
(cm3) $X' \subseteq \alpha_P(\omega_P(X))$ *and* $X' \rightarrowtriangle Y'$ *and* $Y \subseteq \alpha_Q(\omega_Q(Y'))$ *imply* $X \rightarrowtriangle Y$.

We give some intuition about this notion of morphism. We think of sets of attributes as carrying knowledge, or information, and morphisms from P to Q relate this knowledge in the sense that some knowledge in P implies some knowledge in Q. So $X \rightarrowtriangle Y$ should be read as: “If at least X is known, then also at least Y is known.” Conditions (cm1) and (cm2) are easily understood from this perspective. Condition (cm3) uses the idea that closure in FCA (i.e. the formation of $\alpha_P(\omega_P(X))$ from some X) can be understood as logical consequence, i.e. $X' \subseteq \alpha_P(\omega_P(X))$ means that X carries more knowledge than X', as remarked in Condition (cm3). Thus it allows us to strenghten on the left-hand side of the relation, and to weaken on the right-hand side.

We show now that we indeed obtain a category.

Proposition 1 (identity context morphism). *With notation as in Definition 3, the relation ι_P defined by*

$$X \iota_P Y \text{ iff } Y \subseteq \alpha(\omega(X))$$

defines a context morphism, which we call the identity context morphism.

Proof. Conditions (cm1) and (cm2) are obviously satisfied. Condition (cm3) follows from monotonicity of $\alpha \circ \omega$ and the fact that $\alpha \circ \omega$ is idempotent. □

Composition of context morphisms is composition of relations, so there is nothing to show in this respect.

Theorem 3. *Formal contexts together with context morphisms constitute a category* **Cxt**.

Proof. We first show that the composition of two context morphisms is a context morphism. So let $\rightarrowtriangle_{PQ}$ and $\rightarrowtriangle_{QR}$ be context morphisms. Then condition (cm1) is easily verified for $(\rightarrowtriangle_{QR} \circ \rightarrowtriangle_{PQ})$. Concerning (cm2), assume $X(\rightarrowtriangle_{QR} \circ \rightarrowtriangle_{PQ})Y_1$ and $X(\rightarrowtriangle_{QR} \circ \rightarrowtriangle_{PQ})Y_2$. Then there exist $Z_1, Z_2 \in \mathsf{Fin}(Q_a)$ with $X \rightarrowtriangle_{PQ} Z_1$, $X \rightarrowtriangle_{PQ} Z_2$, hence $X \rightarrowtriangle_{PQ} Z_1 \cup Z_2$, as well as $Z_i \rightarrowtriangle Y_i$ for $i = 1, 2$. Since $Z_i \subseteq \alpha_Q(\omega_Q(Z_1 \cup Z_2))$ and $Y_i \subseteq \alpha_R(\omega_R(Y_i))$ for $i = 1, 2$, we conclude by (cm3) that $Z_1 \cup Z_2 \rightarrowtriangle_{QR} Y_i$ for $i = 1, 2$, and by (cm2) that $Z_1 \cup Z_2 \rightarrowtriangle_{QR} Y_1 \cup Y_2$ which suffices. For (cm3), assume $X' \subseteq \alpha_P(\omega_P(X))$, $X'(\rightarrowtriangle_{QR} \circ \rightarrowtriangle_{PQ})Y'$, and $Y \subseteq \alpha_R(\omega_R(Y'))$. Then there exists $Z \in \mathsf{Fin}(Q_a)$ with $X' \rightarrowtriangle_{PQ} Z$ and $Z \rightarrowtriangle_{QR} Y'$. Since $Z \subseteq \alpha_Q(\omega_Q(Z))$, we conclude by (cm3) that $X \rightarrowtriangle_{PQ} Z$ and $Z \rightarrowtriangle_{QR} Y$, hence $X(\rightarrowtriangle_{QR} \circ \rightarrowtriangle_{PQ})Y$ by definition of composition, as desired.

The remaining conditions are easily verified: associativity of morphisms follows from the fact that composition of morphisms is composition of relations. The identity axiom follows from (cm3). □

5 Information Systems

We show that the category **Cxt** is equivalent to the cartesian closed category of complete algebraic lattices with Scott continuous functions. Our proof utilizes the fact that the latter category is eqivalent to the category of Scott information systems with trivial consistency predicate and approximable mappings as morphisms. The corresponding definitions are as follows, and can be found in [12, 17,18,22,26,28].

An *information system (with trivial consistency predicate)* $\underline{A}$ is a pair $(A, \vdash_A)$, where A is the *token set* and $\vdash_A \ \mathsf{Fin}(A) \times \mathsf{Fin}(A)$ is the *entailment relation*, and furthermore the conditions

(is1) $a \in X$ implies $X \vdash_A \{a\}$
(is2) $(\forall b \in Y. X \vdash_A \{b\})$ and $Y \vdash_A Z$ imply $X \vdash_A Z$

are satisfied for all $a \in A$ and $X, Y, Z \in \mathsf{Fin}(A)$. An *information state* of an information system is a set $X \subseteq A$ such that $\{a\} \in X$ whenever there is $Y \in \mathsf{Fin}(X)$ with $Y \vdash_A \{a\}$. Information states can be characterized as the images of the operator $\mathsf{state} : 2^A \to 2^A$ defined by

$$\mathsf{state}(X) = \{a \mid \exists Y (Y \in \mathsf{Fin}(X) \text{ and } Y \vdash_A \{a\})\}.$$

The set of all information states of an information system $\underline{A}$ is denoted by $\mathsf{states}(A)$.

Let $\underline{A}$ and $\underline{B}$ be information systems. An *approximable mapping* $\rightsquigarrow_{AB} = \rightsquigarrow$ from $\underline{A}$ to $\underline{B}$ is a relation $\rightsquigarrow \subseteq \mathsf{Fin}(A) \times \mathsf{Fin}(B)$, such that the following conditions are satisfied for all $X, X', Y_1, Y_2 \in \mathsf{Fin}(A)$ and $Y, Y' \in \mathsf{Fin}(B)$.

(am1) $\emptyset \rightsquigarrow \emptyset$
(am2) $X \rightsquigarrow Y_1$ and $X \rightsquigarrow Y_2$ imply $X \rightsquigarrow Y_1 \cup Y_2$
(am3) $X \vdash_A X'$ and $X' \rightsquigarrow Y'$ and $Y' \vdash_B Y$ imply $X \rightsquigarrow Y$

Information systems with trivial consistency predicate together with approximable mappings as morphisms constitute a cartesian closed category, which we denote by **ISys**. The identity morphisms in this case are given by $X \iota_A Y$ iff $X \vdash_A Y$, for any information system $\underline{A}$. Composition of morphisms is composition of relations.

The following definition and theorem are taken from [26].

Definition 4. *For a given formal context $P = (P_o, P_a, \models)$, define a system $\mathcal{IS}(P) = (P_a, \vdash)$ by setting $X \vdash Y$ iff $Y \subseteq \alpha(\omega(X))$.*

Theorem 4. *Given a formal context $P = (P_o, P_a, \models)$, we have that $\mathcal{IS}(P)$ is an information system. Furthermore, a subset $X \subseteq P_a$ is an approximable concept of P if and only if it is a state of the derived information system $\mathcal{IS}(P)$.*

The mapping $\mathcal{IS}$ will later on turn out to be the object part of a functor. The object part of the corresponding left adjoint $\mathcal{CT}$ will be defined next.

Definition 5. *Given an information system $\underline{A} = (A, \vdash)$, let $\mathcal{CT}(A)$ be the formal context* $(\mathsf{states}(A), \mathsf{K}(\mathsf{states}(A)), \subseteq)$*, where* $\mathsf{K}(\mathsf{states}(A))$ *stands for the set of compact elements of* $(\mathsf{states}(A), \subseteq)$*.*

The following is taken from [26].

Theorem 5. *The approximable concepts of $\mathcal{CT}(A)$ coincide with the downward-closed directed sets of compact elements of the complete algebraic lattice*

$$(\mathsf{states}(A), \subseteq).$$

Hence (by ideal completion), $\mathcal{A}(\mathcal{CT}(A))$ is isomorphic to $(\mathsf{states}(A), \subseteq)$ *via the isomorphism $K \mapsto \sup K$, where $\mathcal{A}$ is defined in Theorem 2.*

We describe the action of $\mathcal{IS}$ and $\mathcal{CT}$ on morphisms in order to obtain functors between the respective categories.

Let $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$ be formal contexts, and let $\twoheadrightarrow$ be a context morphism. Let $\mathcal{IS}(P) = (P_a, \vdash_P)$ and $\mathcal{IS}(Q) = (Q_a, \vdash_Q)$. Then define $\mathcal{IS}(\twoheadrightarrow_{PQ}) = \rightsquigarrow \subseteq \mathsf{Fin}(P_a) \times \mathsf{Fin}(Q_a)$ by setting $X \rightsquigarrow Y$ iff $X \twoheadrightarrow Y$.

Theorem 6. *The relation $\rightsquigarrow$ is an approximable mapping and $\mathcal{IS}$ is a functor from* **Cxt** *to* **ISys**.

Proof. Straightforward by inspecting the defining properties of a functor. □

Concerning $\mathcal{CT}$, let $\underline{A} = (A, \vdash_A)$ and $\underline{B} = (B, \vdash_B)$ be information systems and let

$$\mathcal{CT}(A) = P = (\mathsf{states}(A), \mathsf{K}(\mathsf{states}(A)), \subseteq) \qquad \text{and}$$
$$\mathcal{CT}(B) = Q = (\mathsf{states}(B), \mathsf{K}(\mathsf{states}(B)), \subseteq)$$

be corresponding formal contexts as defined in Definition 5. Furthermore, let $\rightsquigarrow$ be an approximable mapping from A to B. Then define

$$\mathcal{CT}(\rightsquigarrow) = \twoheadrightarrow \subseteq \mathsf{Fin}(\mathsf{K}(\mathsf{states}(A))) \times \mathsf{Fin}(\mathsf{K}(\mathsf{states}(B)))$$

by setting $X \twoheadrightarrow Y$ iff for each $Y' \in \mathsf{Fin}(\bigcup Y)$ there exists $X' \in \mathsf{Fin}(\bigcup X)$ with $X' \rightsquigarrow Y'$.

Lemma 1. *The relation $\twoheadrightarrow$ is a context morphism.*

Proof. For (cm1) note that $\mathsf{Fin}(\bigcup \emptyset) = \{\emptyset\}$.

For (cm2) let $X \twoheadrightarrow Y_1$ and $X \twoheadrightarrow Y_2$, i.e. for each $Y_1' \in \mathsf{Fin}(\bigcup Y_1)$ there exists $X' \in \mathsf{Fin}(\bigcup X)$ with $X' \rightsquigarrow Y_1'$, and for each $Y_2' \in \mathsf{Fin}(\bigcup Y_2)$ there exists $X'' \in \mathsf{Fin}(\bigcup X)$ with $X'' \rightsquigarrow Y_2'$. Now let $Y \in \mathsf{Fin}(\bigcup(Y_1 \cup Y_2))$. Then there exist $Y_1' \in \mathsf{Fin}(\bigcup Y_1)$ and $Y_2' \in \mathsf{Fin}(\bigcup Y_2)$ with $Y_1' \cup Y_2' = Y$. So we also have $X' \in \mathsf{Fin}(\bigcup X)$ with $X' \rightsquigarrow Y_1'$ and $X'' \in \mathsf{Fin}(\bigcup X)$ with $X'' \rightsquigarrow Y_2'$. Since $X' \cup X'' \in \mathsf{Fin}(\bigcup X)$ and $X' \cup X'' \vdash_A X'$ and $X' \cup X'' \vdash_A X''$, we obtain by (am3) that $X' \cup X'' \rightsquigarrow Y_i'$ (for $i = 1, 2$), and hence $X' \cup X'' \rightsquigarrow Y_1' \cup Y_2' = Y$. By $X' \cup X'' \in \mathsf{Fin}(\bigcup X)$ we conclude $X \twoheadrightarrow Y_1 \cup Y_2$.

For (cm3), note that for $X, Y \in \mathsf{Fin}(\mathsf{K}(\mathsf{state}(A)))$ we have $X \subseteq \alpha_P(\omega_P(Y))$ if and only if $\bigcup X \subseteq \bigcup Y$. Now assume that $X' \subseteq \alpha_P(\omega_P(X))$ and $X' \twoheadrightarrow Y'$ and $Y \subseteq \alpha_Q(\omega_Q(Y'))$. Then $\bigcup X' \subseteq \bigcup X$ and $\bigcup Y \subseteq \bigcup Y'$ and for each $Y'' \in \mathsf{Fin}(\bigcup Y')$ there exists $X'' \in \mathsf{Fin}(\bigcup X')$ with $X'' \rightsquigarrow Y''$. But then in particular, for each $Y'' \in \mathsf{Fin}(\bigcup Y)$ there exists $X'' \in \mathsf{Fin}(\bigcup X)$ with $X'' \rightsquigarrow Y''$. So $X \twoheadrightarrow Y$. □

Theorem 7. *$\mathcal{CT}$ is a functor from* **ISys** *to* **Cxt**.

Proof. Concerning the identity condition, we have $X\, \mathcal{CT}(\mathrm{id}_A) Y$ iff for each $Y' \in \mathsf{Fin}(\bigcup Y)$ there exists $X' \in \mathsf{Fin}(\bigcup X)$ with $X' \vdash_A Y'$. Or in other words, we have $X\, \mathcal{CT}(\mathrm{id}_A) Y$ iff for each $Y' \in \mathsf{Fin}(\bigcup Y)$ with $\mathsf{state}(Y') \subseteq \mathsf{state}(\bigcup Y)$ there exists

$X' \in \mathsf{Fin}(\bigcup X)$ with $\mathsf{state}(X') \subseteq \mathsf{state}(\bigcup X)$ and $\mathsf{state}(Y') \subseteq (X')$. Since finitely generated states are compact in the complete algebraic lattice $(\mathsf{states}(A), \subseteq)$, this is equivalent to the statement $\mathsf{state}(\bigcup Y) \subseteq \mathsf{state}(\bigcup X)$, or in other words, $Y \subseteq \alpha(\omega(X))$.

It remains to show that $\mathcal{CT}(\leadsto_{BC} \circ \leadsto_{AB}) = \mathcal{CT}(\leadsto_{BC}) \circ \mathcal{CT}(\leadsto_{AB})$. Let first $X\, \mathcal{CT}(\leadsto_{BC}) \circ \mathcal{CT}(\leadsto_{AB}) Y$, i.e. there is Z with $X\, \mathcal{CT}(\leadsto_{BC}) Z$ and $Z\, \mathcal{CT}(\leadsto_{AB}) Y$. This means that for all $Y' \in \mathsf{Fin}(\bigcup Y)$ there exists $Z' \in \mathsf{Fin}(\bigcup Z)$ with $Z' \leadsto_{BC} Y'$ and for all $Z' \in \mathsf{Fin}(\bigcup Z)$ there exists $X' \in \mathsf{Fin}(\bigcup X)$ with $X' \leadsto_{AB} Z'$. Consequently, for all $Y' \in \mathsf{Fin}(\bigcup Y)$ there is $X' \in \mathsf{Fin}(\bigcup X)$ with $X'(\leadsto_{BC} \circ \leadsto_{AB}) Y'$, i.e. $X\, \mathcal{CT}(\leadsto_{BC} \circ \leadsto_{AB}) Y$.

Conversely, let $X\, \mathcal{CT}(\leadsto_{BC} \circ \leadsto_{AB}) Y$, i.e. for all $Y' \in \mathsf{Fin}(\bigcup Y)$ there exists $X' \in \mathsf{Fin}(\bigcup X)$ with $X'(\leadsto_{BC} \circ \leadsto_{AB}) Y'$. Hence, for all $Y' \in \mathsf{Fin}(\bigcup Y)$ there exist $X' \in \mathsf{Fin}(\bigcup X)$ and $Z' \in \mathsf{Fin}(Q_a)$ with $X' \leadsto_{AB} Z'$ and $Z' \leadsto_{BC} Y'$. Let $Y = \{Y_1, \ldots, Y_n\}$. Then each Y_i is a compact state, hence for each i there is $Y_i' \in \mathsf{Fin}(\bigcup Y)$ with $\mathsf{state}(Y_i') = \mathsf{state}(Y_i)$. For each such Y_i' there exist $Z_i' \in \mathsf{Fin}(Q_a)$ and $X_i' \in \mathsf{Fin}(\bigcup X)$ with $X_i' \leadsto_{AB} Z_i'$ and $Z_i' \leadsto_{BC} Y_i'$. Let $Z = \{\mathsf{state}(Z_1'), \ldots, \mathsf{state}(Z_n')\}$. Now, given $Y' \in \mathsf{Fin}(\bigcup Y)$ we obtain $\bigcup Y_i' \vdash_C Y'$. By (am2) and (am3) we have $\bigcup_i Z_i' \leadsto_{BC} \bigcup Y_i'$, and by (am3) we obtain $\bigcup_i Z_i' \leadsto_{BC} Y'$. Since $\bigcup_i Z_i' \in \mathsf{Fin}(\bigcup Z)$, we obtain $Z\, \mathcal{CT}(\leadsto_{BC}) Y$. The same kind of argument along $\leadsto_{AB}$ yields $X\, \mathcal{CT}(\leadsto_{BC}) Z$, which completes the proof. □

6 Categorical Equivalence and Cartesian Closedness

In this section we establish the fact that $\mathcal{IS}$ and $\mathcal{CT}$ provide equivalences between categories.

Recall that a morphism $f \in \mathbf{C}(A, B)$ is called an *isomorphism* if there is a (necessarily unique) morphism $g \in \mathbf{C}(B, A)$ such that $g \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$. The *identity functor* on a category $\mathbf{C}$ maps all objects and morphisms to themselves, and is denoted by $\mathrm{id}_{\mathbf{C}}$. A *natural transformation* $\eta : \mathbf{F} \Rightarrow \mathbf{G}$ between functors $\mathbf{F}, \mathbf{G} : \mathbf{A} \to \mathbf{B}$ is a class of morphisms $\eta_A \in \mathbf{B}(\mathbf{F}A, \mathbf{G}A)_{A \in |\mathbf{A}|}$ such that for every $f \in \mathbf{A}(A, A')$ we have $\eta_{A'} \circ \mathbf{F}f = \mathbf{G}f \circ \eta_A$. A natural transformation is called a *natural isomorphism* if all of its morphisms are isomorphisms. A functor $\mathbf{F}$ from $\mathbf{A}$ to $\mathbf{B}$ is called an *equivalence of categories* iff there is a functor $\mathbf{G}$ from $\mathbf{B}$ to $\mathbf{A}$ and two natural isomorphisms $\mathrm{id}_{\mathbf{B}} \Rightarrow \mathbf{FG}$ and $\mathbf{GF} \Rightarrow \mathrm{id}_{\mathbf{A}}$. $\mathbf{G}$ is then also an equivalence of categories, and is left adjoint to $\mathbf{F}$.

Lemma 2. *There exists a natural transformation $\eta : \mathcal{CT} \circ \mathcal{IS} \Rightarrow id_{\mathbf{Cxt}}$, i.e. a class of context morphism $(\eta_P)_P$ from $\mathcal{CT}(\mathcal{IS}(P))$ to P, where P ranges over all formal contexts, such that for every context morphism $\multimap$ between formal contexts P and Q we have $\eta_Q \circ \mathcal{CT}(\mathcal{IS}(\multimap)) = \multimap \circ \eta_P$. Furthermore, η is a natural isomorphism, i.e. all η_P are isomorphisms – in other words, for each η_P there exists a context morphism $\multimap_P$ from P to $\mathcal{CT}(\mathcal{IS}(P))$ such that $\multimap_P \circ \eta_P = \iota_P$ and $\eta_P \circ \multimap_P = \iota_{\mathcal{CT}(\mathcal{IS}(P))}$.*

Proof. Let $P = (P_o, P_a, \models_P)$ be some formal context. We define η_P by setting $S\eta_P X$, for $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(\mathcal{IS}(P))))$ and $X \in \mathsf{Fin}(P_a)$, whenever $X \in \mathsf{Fin}(\bigcup S)$. It is easily verified that η_P is a context morphism.

Now let $T(\eta_Q \circ \mathcal{CT}(\mathcal{IS}(\twoheadrightarrow)))X$. This is equivalent to saying that there exists some $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(\mathcal{IS}(P))))$ such that $X \in \mathsf{Fin}(\bigcup S)$ and for all $s \in \mathsf{Fin}(\bigcup S)$ there exists some $t \in \mathsf{Fin}(\bigcup T)$ with $t \twoheadrightarrow s$. This implies that there is $t \in \mathsf{Fin}(\bigcup T)$ with $t \twoheadrightarrow X$, which in turn is equivalent to $T(\twoheadrightarrow \circ \eta_P)X$. Conversely, let $T(\twoheadrightarrow \circ \eta_P)X$, which is equivalent to saying that there is $t \in \mathsf{Fin}(\bigcup T)$ with $t \twoheadrightarrow X$. Now with $S = \{\mathsf{state}(X)\} \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(\mathcal{IS}(P))))$ this implies that $X \in \mathsf{Fin}(\bigcup S)$ and (by (cm3)) for all $s \in \mathsf{Fin}(\bigcup S)$ there exists $t \in \mathsf{Fin}(\bigcup T)$ with $t \twoheadrightarrow s$. This is in turn equivalent to $T(\eta_Q \circ \mathcal{CT}(\mathcal{IS}(\twoheadrightarrow)))X$, as noted earlier in this paragraph.

To show that all η_P are isomorphisms, let $\twoheadrightarrow_P$ be the context morphism from P to $\mathcal{CT}(\mathcal{IS}(P))$ which is defined by $X \twoheadrightarrow_P S$, for $X \in \mathsf{Fin}(P_a)$ and $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(\mathcal{IS}(P))))$, whenever $\bigcup S \subseteq \alpha_P(\omega_P(X))$. It is easily verified that $\twoheadrightarrow_P$ is indeed a context morphism. Now $X(\eta_P \circ \twoheadrightarrow_P)Y$ iff there exists S with $X \twoheadrightarrow_P S$ and $S\eta_P Y$, i.e. iff there exists S with $\bigcup S \subseteq \alpha_P(\omega_P(X))$ and $Y \in \mathsf{Fin}(\bigcup S)$. This in turn is equivalent to $Y \in \mathsf{Fin}(\alpha_P(\omega_P(X)))$, i.e. to $X\iota_P Y$ as desired. Likewise, let $T, S \in \mathsf{Fin}(\mathsf{K}(\mathsf{state}(\mathcal{IS}(P))))$. Then $S(\twoheadrightarrow_P \circ \eta_P)T$ iff there exists Y with $S\eta_P Y$ and $Y \twoheadrightarrow_P T$, i.e. iff $Y \in \mathsf{Fin}(\bigcup S)$ and $\bigcup T \subseteq \alpha_P(\omega_P(Y))$. Since S and T are sets of compact states, this in turn is equivalent to $\bigcup T \subseteq \bigcup S$, i.e. to $T\iota_{\mathcal{CT}(\mathcal{IS}(P))}S$. □

Lemma 3. *There exists a natural transformation* $\eta : \mathrm{id}_{\mathbf{ISys}} \Rightarrow \mathcal{IS} \circ \mathcal{CT}$*, i.e. a class of approximable mappings* $(\eta_A)_{\underline{A}}$ *from* $\underline{A}$ *to* $\mathcal{IS}(\mathcal{CT}(A))$*, where* $\underline{A}$ *ranges over all information systems (with trivial consistency predicate), such that for every approximable mapping* $\rightsquigarrow$ *between information systems* $\underline{A}$ *and* $\underline{B}$ *we have* $\eta_B \circ \rightsquigarrow = \mathcal{IS}(\mathcal{CT}(\rightsquigarrow)) \circ \eta_A$*. Furthermore,* η *is a natural isomorphism, i.e. all* η_A *are isomorphisms — or in other words, for each* η_A *there exists a context morphism* $\rightsquigarrow_A$ *from* $\mathcal{IS}(\mathcal{CT}(A)))$ *to* $\underline{A}$ *such that* $\rightsquigarrow_A \circ \eta_A = \iota_A$ *and* $\eta_A \circ \rightsquigarrow_A = \iota_{\mathcal{IS}(\mathcal{CT}(A))}$.

Proof. Let $\underline{A} = (A, \vdash)$ be some information system. We define η_A by setting $X\eta_A S$, for $X \in \mathsf{Fin}(A)$ and $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(A)))$, whenever $X \vdash_A s$ for all $s \in \mathsf{Fin}(\bigcup S)$. It is easily verified that η_A is an approximable mapping.

We have $X(\eta_B \circ \rightsquigarrow)T$ iff there exists $Y \in \mathsf{Fin}(B)$ such that $X \rightsquigarrow Y$ and for all $t \in \mathsf{Fin}(\bigcup T)$ we have $Y \vdash_B t$. This is equivalent to the statement that

$$(\sharp) \text{ for all } t \in \mathsf{Fin}(\textstyle\bigcup T) \text{ we have } X \rightsquigarrow t.$$

On the other hand, we have $X(\mathcal{IS}(\mathcal{CT}(\rightsquigarrow)) \circ \eta_A)T$ if and only if

(∗) there exists $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(A)))$ such that for all $s \in \mathsf{Fin}(\bigcup S)$ we have $X \vdash_A s$ and for all $t \in \mathsf{Fin}(\bigcup T)$ there exists $u \in \mathsf{Fin}(\bigcup S)$ with $u \rightsquigarrow t$.

Statement (♯) implies (∗) via $S = \{\mathsf{state}(X)\}$. Statement (∗) implies that for all $t \in \mathsf{Fin}(\bigcup T)$ there exists $u \in \mathsf{Fin}(\bigcup S)$ with $X \vdash_A u$ and $u \rightsquigarrow t$, which by (am3) implies (♯).

To show that all η_A are isomorphisms, let $\rightsquigarrow_A$ be the approximable mapping from $\mathcal{IS}(\mathcal{CT}(A))$ to $\underline{A}$ defined by $S \rightsquigarrow_A X$, for all $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(A)))$ and $X \in \mathsf{Fin}(A)$, whenever $X \in \mathsf{Fin}(\bigcup S)$. It is easily verified that $\rightsquigarrow_A$ is an approximable mapping. Now $X(\rightsquigarrow_A \circ \eta_A)Y$ if and only if there exists $S \in \mathsf{Fin}(\mathsf{K}(\mathsf{states}(A)))$ with $X\eta_A S$ and $S \rightsquigarrow_A Y$, i.e. $X \vdash_A s$ for all $s \in \mathsf{Fin}(\bigcup S)$ and $Y \in \mathsf{Fin}(\bigcup S)$. This in turn is equivalent to $X \vdash_A Y$, i.e. to $X\iota_A Y$ as desired. Likewise, $S(\eta_A \circ \rightsquigarrow_A)T$ if and only if there exists $X \in \mathsf{Fin}(A)$ with $S \rightsquigarrow_A X$ and $X\eta_A T$. This in turn is equivalent to saying that there exists $X \in \mathsf{Fin}(\bigcup S)$ such that $X \vdash_A t$ for all $t \in \mathsf{Fin}(\bigcup T)$. Since S and T are sets of compact states, this is equivalent to the statement $\bigcup T \subseteq \bigcup S$, i.e. to $S\iota_{\mathcal{IS}(\mathcal{CT}(A))}T$. □

Theorem 8. *Both $\mathcal{CT}$ and $\mathcal{IS}$ are equivalences of categories, i.e. the categories* **Cxt** *and* **ISys** *are equivalent, and* **Cxt** *is cartesian closed.*

Proof. It suffices to show that there are natural isomorphisms $\mathcal{CT} \circ \mathcal{IS} \Rightarrow \mathrm{id}_{\mathbf{Cxt}}$ and $\mathrm{id}_{\mathbf{ISys}} \Rightarrow \mathcal{IS} \circ \mathcal{CT}$, which were provided by Lemmata 2 and 3. The last statement follows from the well-known fact that **ISys** is cartesian closed. □

7 Constructions

We have established our main result in Theorem 8 and shown that **Cxt** is cartesian closed. We now study what the corresponding categorical constructions, i.e. product and function space, look like. Although the existence of these constructions has been justified in previous sections, carrying out the actual constructions in full detail can be an engineering endeavor [22,28].

Terminal Object

The *unit* or *terminal object* in **Cxt** is the context $\mathbf{1} = (\emptyset, \emptyset, \emptyset)$, and for each context $P = (P_o, P_a, \models)$ the unique context morphism $\rightarrow$ from P to $\mathbf{1}$ is given by $X \rightarrow \emptyset$ for all $X \in \mathsf{Fin}(P_a)$.

Product

For a formal context $P = (P_o, P_a, \models_P)$ let $P' = (P'_o, P'_a, \models_{P'})$, where $P'_o = P_o \cup \{g\}$, $P'_a = P_a \cup \{m\}$, and g, m are some elements not in P_o respectively P_a. $\models_{P'}$ coincides with $\models_P$ on $P_a \times P_o$ and $g \models_{P'} a$ for all $a \in P_a \cup \{m\}$ and $h \models_{P'} m$ for all $h \in P_o \cup \{g\}$. Informally, P' is obtained from P by "adding a full row and a full column".

Let $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$ be formal contexts. Define the *product*

$$P \times Q = (P'_o \times Q'_o, P'_a \times Q'_a, \models_{P \times Q})$$

of P and Q by setting $(g_1, g_2) \models_{P \times Q} (m_1, m_2)$ iff $g_1 \models_{P'} m_1$ and $g_2 \models_{Q'} m_2$. Obviously, $P \times Q$ is a formal context.

Theorem 9. *Let $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$ be formal contexts. Then there exist context morphisms $\pi_P : P \times Q \to P$ and $\pi_Q : P \times Q \to Q$ such that for all context morphisms $\rightarrowtriangle_P$ from R to P and $\rightarrowtriangle_Q$ from R to Q there exists exactly one context morphism $\langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle$ from R to $P \times Q$ such that*

$$\pi_P \circ \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle = \rightarrowtriangle_P \qquad \textit{and} \qquad \pi_Q \circ \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle = \rightarrowtriangle_Q .$$

Proof. We define π_P by setting $\{(m_i, m_i') \mid i = 1, \ldots, k\} \pi_P \{n_j \mid j = 1, \ldots, l\}$ iff $\{n_j \mid j = 1, \ldots, l\} \subseteq \alpha_P(\omega_P(\{m_i \mid i = 1, \ldots, k\}))$. This is equivalent to saying that $M \pi_P N$ iff $(\pi_1(M) \cap P_a) \iota_P N$, where $\pi_1(M)$ denotes the projection of the set of pairs M to the set of the first components of its elements. The context morphism π_Q is defined analogously. Define $X \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle Y$ iff $X \rightarrowtriangle_P (\pi_1(Y) \cap P_a)$ and $X \rightarrowtriangle_Q (\pi_2(Y) \cap Q_a)$, where π_2 is the corresponding projection on the second component.

We next show $\pi_P \circ \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle = \rightarrowtriangle_P$. Denoting $\langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle$ by $\leadsto$, we then obtain $X(\pi_P \circ \leadsto) Y$ iff there exists some $Z \in \mathsf{Fin}(P_a' \times Q_a')$ with $X \leadsto Z$ and $Z \pi_P Y$, which is the case iff $(*)$ there exists Z with $X \rightarrowtriangle_P (\pi_1(Z) \cap P_a)$, $X \rightarrowtriangle_Q (\pi_2(Z) \cap Q_a)$ and $(\pi_1(Z) \cap P_a) \iota_P Y$. This in turn implies that there is $Z \in \mathsf{Fin}(P_a' \times Q_a')$ with $X \rightarrowtriangle_P Y$ and $X \rightarrowtriangle_Q (\pi_2(Z) \cap Q_a)$. Such a Z trivially exists: every Z with $\pi_2(Z) = \emptyset$ satisfies the condition. So this condition reduces to $X \rightarrowtriangle_P Y$ as required. Conversely, assume $X \rightarrowtriangle_P Y$. Let $Z = Y \times (Q_a' \setminus Q_a)$. Then this implies condition $(*)$ which was shown above to be equivalent to $X(\pi_P \circ \leadsto) Y$. The equation $\pi_Q \circ \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle = \rightarrowtriangle_Q$ is shown similarly.

For uniqueness assume that there is $\rightarrowtriangle : R \to P \times Q$ which satisfies $\pi_P \circ \rightarrowtriangle = \rightarrowtriangle_P$ and $\pi_Q \circ \rightarrowtriangle = \rightarrowtriangle_Q$. Then we obtain

$$\begin{aligned}
X \rightarrowtriangle Z \quad & \text{iff} \quad X \rightarrowtriangle Z, \quad (\pi_1(Z) \cap P_a)\, \iota_P\, (\pi_1(Z) \cap P_a), \\
& \qquad\qquad \text{and} \quad (\pi_2(Z) \cap Q_a)\, \iota_Q\, (\pi_2(Z) \cap Q_a) \\
& \text{iff} \quad X \rightarrowtriangle Z, \quad Z\, \pi_P\, (\pi_1(Z) \cap P_a), \quad \text{and} \quad Z\, \pi_Q\, (\pi_2(Z) \cap Q_a) \\
& \text{iff} \quad X(\pi_P \circ \rightarrowtriangle)(\pi_1(Z) \cap P_a) \quad \text{and} \quad X(\pi_Q \circ \rightarrowtriangle)(\pi_2(Z) \cap Q_a) \\
& \text{iff} \quad X \rightarrowtriangle_P (\pi_1(Z) \cap P_a) \quad \text{and} \quad X \rightarrowtriangle_Q (\pi_2(Z) \cap Q_a)
\end{aligned}$$

which shows $\rightarrowtriangle = \langle \rightarrowtriangle_P, \rightarrowtriangle_Q \rangle$. □

Function Space

Given formal contexts $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$, we define the function space $P \to Q$ as follows. Consider the set $M = \mathsf{Fin}(P_a) \times \mathsf{Fin}(Q_a)$. We define a Scott information system on M. For all collections $\{(u_i, u_i')\} \in \mathsf{Fin}(M)$ and $\{(v_j', v_j')\} \in \mathsf{Fin}(M)$, let $\{(u_i, u_i')\} \vdash_M \{(v_j, v_j')\}$ iff

$$v_j' \subseteq \alpha_Q \left(\omega_Q \left(\bigcup \{ u_i' \mid u_i \subseteq \alpha_P(\omega_P(v_j)) \} \right) \right)$$

for all j. It is easily verified that $(M, \vdash_M)$ is a Scott information system: In order to show condition (is2), assume that we have $\{(y_i, y_i')\} \vdash_M \{(z, z')\}$ and

$\{(x_j, x'_j)\} \vdash_M (y_i, y'_i)$ for all i. Then $z' \subseteq \alpha_Q\left(\omega_Q\left(\bigcup\{y'_i \mid y_i \subseteq \alpha_P(\omega_P(z))\}\right)\right)$ and $y'_i \subseteq \alpha_Q\left(\omega_Q\left(\bigcup\{x'_j \mid x_j \subseteq \alpha_P(\omega_P(y_i))\}\right)\right)$ for all i. So

$$\begin{aligned}
z' &\subseteq \alpha_Q\left(\omega_Q\left(\bigcup\nolimits_i \left\{\alpha_Q\left(\omega_Q\left(\bigcup\nolimits_j \{x'_j \mid x_j \subseteq \alpha_P(\omega_P(y_i))\}\right)\right) \mid y_i \subseteq \alpha_P(\omega_P(z))\right\}\right)\right) \\
&\subseteq \alpha_Q\left(\omega_Q\left(\bigcup\nolimits_i \left\{\alpha_Q\left(\omega_Q\left(\bigcup\nolimits_j \{x'_j \mid x_j \subseteq \alpha_P(\omega_P(z))\}\right)\right) \mid y_i \subseteq \alpha_P(\omega_P(z))\right\}\right)\right) \\
&\subseteq \alpha_Q\left(\omega_Q\left(\alpha_Q\left(\omega_Q\left(\bigcup\nolimits_j \{x'_j \mid x_j \subseteq \alpha_P(\omega_P(z))\}\right)\right)\right)\right) \\
&= \alpha_Q\left(\omega_Q\left(\bigcup\{x'_j \mid x_j \subseteq \alpha_P(\omega_P(z))\}\right)\right)
\end{aligned}$$

as required. The function space $P \to Q$ is then defined as the context

$$\mathcal{CT}(M) = (\mathsf{states}(M), \mathsf{K}(\mathsf{states}(M)), \subseteq).$$

Theorem 10. *Let $P = (P_o, P_a, \models_P)$ and $Q = (Q_o, Q_a, \models_Q)$ be formal contexts. Then the context morphisms from P to Q are exactly the approximable concepts in $P \to Q$.*

Proof. Let A be an approximable concept in $P \to Q$. Then $A \in \mathsf{states}(M)$ by Theorem 5, and we have to show that A is a context morphism. Conditions (cm1) and (cm2) are easily verified. Condition (cm3) follows from the monotonicity of $\alpha_P \circ \omega_P$ and $\alpha_Q \circ \omega_Q$.

Conversely, let A be a context morphism from P to Q. By Theorem 5, we have to show that $A \in \mathsf{states}(M)$. So let $\{(x_i, x'_i)\} \in \mathsf{Fin}(A)$ and $\{(x_i, x'_i)\} \vdash_M \{(a, a')\}$. We have to show that $(a, a') \in A$. From $\{(x_i, x'_i)\} \vdash_M \{(a, a')\}$ we obtain

$$a' \subseteq \alpha_Q\left(\omega_Q\left(\bigcup\{x'_i \mid x_i \subseteq \alpha_P(\omega_P(a))\}\right)\right).$$

Setting $X = \bigcup\{x_i \mid x_i \subseteq \alpha_P(\omega_P(a))\}$ and $X' = \bigcup\{x'_i \mid x_i \subseteq \alpha_P(\omega_P(a))\}$, we obtain $a' \subseteq \alpha_Q(\omega_Q(X'))$ and $X \subseteq \alpha_P(\omega_P(a))$. By (cm2) and (cm3) we also have $(X, X') \in A$, and so by (cm3) again we have $(a, a') \in A$. □

8 Concluding Remarks

We have proposed a notion of morphism for formal contexts which results in a cartesian closed category. Our work is in the spirit of both formal concept analysis and domain theory, and makes way for the cross-transfer of methods and results between these areas. Our work builds on a domain-theoretic perspective on formal concepts, which results in complete algebraic lattices instead of complete lattices as the corresponding concept hierarchies.

Technically, our contribution uses Scott information systems in order to capture the logical content of formal contexts. The general connection between information systems and formal contexts was explicitly outlined in [25]. Such a connection breaks down with classical formal concepts for a certain "discontinuous" class of infinite contexts. This lead to the work reported in [26] in which the notion of approximable concept was introduced and the connection between information systems and formal contexts was established in full generality with the corresponding structure being the complete algebraic lattices. Work along a similar line for the finite case was spelled out in [8] in a different guise using the logic RZ due to [16]. The latter work has also led to the proposal of a non-monotonic reasoning paradigm on (possibly infinite) formal contexts [9], which is in the spirit of the recent evolving answer set programming paradigm [6,20]. Also worth noting are potential connections between our work and that of Lamarche [11] and Plotkin [14].

The work reported here can be viewed as part of a unique research program [8,25,26,27,19] exploiting the synergies among some recurring themes in several independently developing and yet somewhat related areas, such as databases, data-mining, domain theory, logic, and formal concept analysis. Additional interesting connections could be profitably explored with ontological engineering and semantic web. For example, in [27], formal concept analysis has been applied as a formal method for automated web-menu design, where the top layers of the concept lattice naturally provide a menu hierarchy for the navigation of a website. In ontological engineering (e.g. [7,21]), although lattices have been proposed as mathematical structures for representing ontology, FCA provides a scientific and algorithmic basis for it, as well as an understanding that lattice structures are both necessary and sufficient for expressing ontological hierarchies.

Acknowledgment. The first author thanks Markus Krötzsch for some first aid in category theory. We thank the anonymous referees for valuable feedback.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis — Mathematical Foundations. Springer, Berlin (1999).
2. Barr, M.: *-Autonomous categories and linear logic. *Mathematical Structures in Computer Science* 1 (2), 1991.
3. Barwise, J., Seligman, J.: *Information Flow: the logic of distributed systems.* Cambridge University Press, 1997.
4. Birkhoff, G.: *Lattice Theory.* AMS Colloquium Publications, Vol. 25, 1940.
5. Borceux, F.: Handbook of Categorical Algebra 1: Basic Category Theory. Volume 53 of Encyclopedia of Mathematics and its Applictions. Cambridge University Press (1994).
6. Eiter, T., Leone, N., Mateis, C., Pfeifer, G., Scarcello, F.: A deductive system for nonmonotonic reasoning. In Dix, J., Furbach, U., Nerode, A., eds.: Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'97. Volume 1265 of Lecture Notes in Artificial Intelligence., Springer, Berlin (1997).
7. Guarino, N., Poli, R. (eds.): *Formal Ontology in Conceptual Analysis and Knowledge Representation.* Kluwer Academic Publishers, 1993.
8. Hitzler, P., Wendt, M.: Formal concept analysis and resolution in algebraic domains. In de Moor, A., Ganter, B., eds.: Using Conceptual Structures — Contributions to ICCS 2003, Shaker Verlag, Aachen (2003) 157–170.

9. Hitzler, P.: Default reasoning over domains and concept hierarchies. Technical Report WV–03–14, Knowledge Representation and Reasoning Group, Department of Computer Science, Dresden University of Technology (2003).
10. Johnstone, P.: *Stone Spaces*, Cambridge University Press, 1982.
11. Lamarche, F.: From Chu spaces to cpos. *Theory and Formal Methods of Computing 94*, Imperial College Press, 283–305, 1994.
12. Larsen, K., Winskel, G.: Using information systems to solve recursive domain equations. *Information and Computation*, Vol. 91(2):232–258, 1991.
13. MacLane, S.: *Categories for the Working Mathematician,* Springer-Verlag (1971).
14. Plotkin, G.: Notes on the Chu construction and recursion, manuscript, 1993.
15. Pratt, V.: Chu spaces from the representational viewpoint. *Ann. Pure Appl. Logic* 96 no. 1-3, 319–333, 1999.
16. Rounds, W.C., Zhang, G.-Q.: Clausal logic and logic programming in algebraic domains. Information and Computation **171** (2001) 156–182.
17. Saraswat, V.: The category of constraint systems is Cartesian-closed. In: Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, California, June 1992, IEEE Computer Society Press (1992) 341–345.
18. Scott, D.S.: Domains for denotational semantics. In Nielsen, M., Schmidt, E.M., eds.: Automata, Languages and Programming, 9th Colloquium, July 1982, Aarhus, Denmark, Proceedings. Volume 140 of Lecture Notes in Computer Science., Springer, Berlin (1982) 577–613.
19. Shen, G., Ye, T., Sun, J. and Zhang, G.-Q.: Concept lattices, clustering, and visualization using LaTeX, manuscript, 2004.
20. Simons, P., Niemelä, I., Soininen, T.: Extending and implementing the stable model semantics. Artificial Intelligence (200x), To appear.
21. Sowa, J.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., 2000.
22. Spreen, D.: A note on strongly finite sequent structures. In Zhang, Lawson, Liu, Luo (eds.), *Domain Theory, Logic and Computation* (Semantic Structures in Computation, Vol. 3), Kluwer Academic Publishers, Dordrecht, 2004.
23. Vickers, S.: *Topology via Logic*, Cambridge University Press, 1989.
24. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered sets*, Reidel, Dordrecht-Boston(1982), 445–470.
25. Zhang, G.-Q.: Chu spaces, formal concepts, and domains. Proceedings of the 19th Conference on the Mathematical Foundations of Programming Semantics, Montreal, Canada, March 19-23, 2003, Electronic Notes in Computer Science Vol. 83, 16 pages, 2003.
26. Zhang, G.-Q., Shen, G.: Approximable concepts, Chu spaces, and information systems. In de Paiva and Pratt (Guest Editors), *Special Issue on Chu Spaces and Applications*, *Theory and Applications of Categories*, accepted.
27. Zhang, G.-Q., Shen, G., Staiger, J., Troy, A., Sun, J.: *FcAWN* – concept analysis as a formal method for automated web-menu design. Submitted.
28. Zhang, G.-Q.: *Logic of Domains*, Birkhauser, Boston, 1991.