Electronic Notes in Theoretical Computer Science 83 (2004)
URL: http://www.elsevier.nl/locate/entcs/volume83.html 17 pages

Chu Spaces, Concept Lattices, and Domains

Guo-Qiang Zhang [1]

Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106, U. S. A.

Abstract

This paper serves to bring three independent but important areas of computer science to a common meeting point: Formal Concept Analysis (FCA), Chu Spaces, and Domain Theory (DT). Each area is given a perspective or reformulation that is conducive to the flow of ideas and to the exploration of cross-disciplinary connections. Among other results, we show that the notion of states in Scott's information system corresponds precisely to that of formal concepts in FCA with respect to all finite Chu spaces, and the entailment relation corresponds to "association rules".

1 Introduction

This paper serves as the meeting point of three "parallel worlds": Chu spaces, Domain Theory, and Formal Concept Analysis. It brings the three independent areas together and establishes fundamental connections among them, leaving open opportunities for the exploration of cross-disciplinary influences.

We begin with an overview of each of the three areas, followed by an account of the background of each area from a unified perspective. We then move to basic connections among them and point to topics of immediate interest and opportunities for further development, including applications in data-mining and knowledge discovery.

Due to its interdisciplinary nature, the paper is written in a way that does not assume specific background knowledge for each area.

1.1 Domain theory

Domain theory (DT) was introduced by Scott in the late 60s for the denotational semantics of programming languages. It provides the mathematical foundation for the design, definition, and implementation of programming languages, and for systems for the specification and verification of programs.

[1] Emailgqz@eecs.cwru.edu

The fundamental idea of domain theory is partial information and successive approximation. The notion of partial information is captured by a *complete partial order* (cpo). 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 the 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 partial order. The beauty of domain theory is that a higher-order object is treated just as another ordinary object.

For further information on domain theory, see, for example, [1,2,9,20,31,32].

1.2 Formal concept analysis

FCA is an order-theoretic method for the mathematical analysis of scientific data, pioneered by German scientists Wille and others [10] in mid 80's. The novel idea of FCA is the clustering of attributes based on the algebraic principle of Galois connection, forming a partially ordered set called *concept lattice*. The clustering determines which collection of attributes forms a coherent entity called *a concept*, by the philosophical criteria of unity between *extension and intension*. The extension of a concept consists of all objects belonging to the concept, while the intension of a concept consists of attributes common to all these objects. One can then take this as the defining property of a concept: *a collection of attributes which agrees with the intension of its extension*.

Over the past twenty years, FCA has become a powerful tool for clustering, data analysis, information retrieval, knowledge discovery, and ontological engineering, used by over two hundred scientific projects so far. This fruitful development of applied FCA lies in the fact that FCA is susceptible to many interpretations, so that connections can be made with different areas and used by researchers from different disciplines.

1.3 Chu spaces

Category theory (CT) has provided a unified language for managing conceptual complexity in mathematics and computer science. Chu spaces, having its birth place in category theory, was brought to light in computer science through the work of Barr and Seely [3,4,29] as constructive models of linear logic. Pratt's extensive work [21,22,23,24,25,26] broadened the scope of their applications to areas such as models for concurrency and philosophy of logic, information, and computation.

There are substantial culture differences among the three areas. FCA, for example, focuses on internal properties of and algorithms for concept structures almost exclusively on an individual basis, while CT mandates that concept structures should be looked at collectively as a whole, with appropriate morphisms relating one individual structure to another. It can be seen as a *universal object-oriented language*. On the other hand, DT carries an intrinsic higher-order view incorporating the notions of partial information and successive approximation. In a precise sense, FCA and Chu spaces started with the same objects but went to different directions out of their own independent motivations. This paper brings them together again through domain-theoretic methods – Scott's information systems [28], in particular.

Related work

Lamarche [13] provides an order-theoretic model for linear logic grounded on Chu spaces. Hitzler [11] gives an account of how the clausal logic of Rounds and Zhang [27] can be viewed as a way for constructing concepts, in a limited setting. These can be seen as special cases of what is manifested by this paper.

2 Preliminaries

This section reviews terminologies and backgrounds for the three areas mentioned earlier, with the goal to bring them to some common bases.

2.1 Cpos

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 exits $x \in X$ 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 bounded complete algebraic cpos.
Notation. The upper set $\uparrow X$ of a set X is defined to be $\{y \mid \exists x \in X, x \sqsubseteq y\}$. A set is *upward-closed* if $X = \uparrow X$. Similarly, a set is *down-closed* if $X = \downarrow X$.

2.2 Closure systems and closure operators

For any set A, let $\mathcal{P}(A)$ denote the powerset of A. A subset $\mathcal{C}$ of the powerset $\mathcal{P}(A)$ is called a *closure system* on A if $\mathcal{C}$ is closed under arbitrary intersections, i.e., for every $X \subseteq \mathcal{C}$, $\bigcap X \in \mathcal{C}$. Note that, by convention, this implies that the whole space A is always a member of a closure system $\mathcal{C}$, by instantiating X as the empty set in the definition.

A *closure operator* on A is a function $\varphi : \mathcal{P}(A) \to \mathcal{P}(A)$ which is inflationary ($X \subseteq \varphi(X)$), monotonic ($X \subseteq Y \Rightarrow \varphi(X) \subseteq \varphi(Y)$), and idempotent ($\varphi(\varphi(X)) = \varphi(X)$).

Proposition 2.1 *Define a* closed set *with respect to a closure operator*

$$\varphi : \mathcal{P}(A) \to \mathcal{P}(A)$$

to be a fixed point of φ. Then closed sets of φ are precisely sets of the form $\varphi(X)$. The collection of closed sets $\{\varphi(X) \mid X \in \mathcal{P}(A)\}$ forms a closure system on A.

The defining property of closure systems provides arbitrary meets. The join of a subset X can then be obtained as the meet of the set of upper bounds of X, A included. These observations lead to the following basic property about closure systems.

Proposition 2.2 *For any closure system $\mathcal{C}$ on A, the partial order $(\mathcal{C}, \subseteq)$ is a complete lattice with A being the top element.*

2.3 Galois connections

Let P, Q be sets. A pair of functions

$$s : \mathcal{P}(P) \to \mathcal{P}(Q) \quad \text{and} \quad t : \mathcal{P}(Q) \to \mathcal{P}(P)$$

is called a *Galois connection*[2] if for each $X \in \mathcal{P}(P)$ and $Y \in \mathcal{P}(Q)$,

$$s(X) \supseteq Y \quad \text{if and only if} \quad X \subseteq t(Y).$$

The next well-known fact shows that closure operators can be derived from Galois connections in a natural way.

Proposition 2.3 *For any Galois connection s, t with $s : \mathcal{P}(P) \to \mathcal{P}(Q)$ and $t : \mathcal{P}(Q) \to \mathcal{P}(P)$, $s \circ t$ is a closure operator on Q, and $t \circ s$ is a closure operator on P.*

The proof of this proposition uses some basic properties about Galois connections, summarized in the following lemma (for more details see [15]).

Lemma 2.4 *Let the pair (s, t) with $s : \mathcal{P}(P) \to \mathcal{P}(Q)$ and $t : \mathcal{P}(Q) \to \mathcal{P}(P)$ be a Galois connection. The following are true:*

- *$s \circ t$ and $t \circ s$ are inflationary.*

[2] Galois connections appear in the literature in two equivalent versions. The original version uses order-reversing maps [7], and the second version, more popular in computer science, uses order-preserving maps. Since set-containment is more primitive, the notion of Galois connection used in this paper is more concrete, serving our purpose well. Note that we are in fact *neutral* with respect to the issue of order-preserving vs. order-reversing: set-inclusion removes the potential overhead for keeping track of the direction of order.

- *s and t are anti-monotonic, i.e., if $X \subseteq Y$ then $s(Y) \subseteq s(X)$, and similarly for t.*
- *$s \circ t$ and $t \circ s$ are monotonic.*
- *$s \circ t \circ s = s$ and $t \circ s \circ t = t$ and, therefore, $s \circ t$ and $t \circ s$ are idempotent.*

The next definition serves to fix notations only.

Definition 2.5 Any function $f : A \to B$ can be lifted to the powerset level in two canonical ways:

$$f^+ : \mathcal{P}(A) \to \mathcal{P}(B) \quad \text{with} \quad X \longmapsto \{f(a) \mid a \in X\},$$
$$f^- : \mathcal{P}(B) \to \mathcal{P}(A) \quad \text{with} \quad Y \longmapsto \{a \mid f(a) \in Y\}.$$

f^- is the standard *inverse image* operation, and f^+ is the forward image operation. The notation is chosen here by symmetry. Note that $f^+ \circ f^-$ is less than the identity function on $\mathcal{P}(B)$ and $f^- \circ f^+$ dominates identity with respect to coordinatewise inclusion (they form a Galois connection in the general sense [7]).

Proposition 2.6 *For any function $f : A \to B$, we have*

- *$(f^+ \circ f^-)Y \subseteq Y$ for any $Y \in \mathcal{P}(B)$;*
- *$(f^- \circ f^+)X \supseteq X$ for any $X \in \mathcal{P}(A)$;*
- *$f^+ \circ f^-$ is the identity function if and only if f is onto;*
- *$f^- \circ f^+$ is the identity function if and only if f is one-to-one.*

3 Chu spaces and formal concept lattices

We will consider a special form of Chu spaces in this paper. Pratt [25] provides arguments for the use of sets in place of the enriching category V.

Definition 3.1 A Chu space P is a triple $(P_o, \models_P, P_a)$ where P_o is a set of objects and P_a is a set of attributes. The satisfaction relation $\models_P$ is a subset of $P_o \times P_a$. A mapping from a Chu space $(P_o, \models_P, P_a)$ to a Chu space $(Q_o, \models_Q, Q_a)$ is a pair of functions (f_a, f_o), with $f_a : P_a \to Q_a$ and $f_o : Q_o \to P_o$ such that for any $x \in P_a$ and $y \in Q_o$, $f_o(y) \models_P x$ iff $y \models_Q f_a(x)$.

Example 3.2 The most common construction in data-mining is to extend a relation (context) by a row, adding one new object with the observed attributes, but the attribute set remain unchanged. This construction induces a Chu-mapping, from the enlarged space to the initial space, with f_a being the identity function on attributes, and f_o the injection of objects.

Example 3.3 Galois connections can also be viewed as Chu space mappings. Suppose a pair of functions

$$s : \mathcal{P}(P) \to \mathcal{P}(Q) \quad \text{and} \quad t : \mathcal{P}(Q) \to \mathcal{P}(P)$$

forms a Galois connection. Then (s,t) is a mapping from the Chu space $(\mathcal{P}(P), \supseteq, \mathcal{P}(P))$ to the Chu space $(\mathcal{P}(Q), \subseteq, \mathcal{P}(Q))$, because for each $X \in \mathcal{P}(P)$ and $Y \in \mathcal{P}(Q)$, we have $t(Y) \supseteq X$ if and only if $Y \subseteq s(X)$.

A Chu space is called a *context* in FCA, but "Chu" carries with it the notion of morphisms, to form a category. On the other hand, FCA provides the notion of *concepts*, intrinsic to a Chu space.

Definition 3.4 With respect to a Chu space $P = (P_o, \models_P, P_a)$, two functions can be defined:

$$\alpha : \mathcal{P}(P_o) \to \mathcal{P}(P_a) \quad \text{with} \quad X \longmapsto \{a \mid \forall x \in X\ x \models_P a\},$$
$$\omega : \mathcal{P}(P_a) \to \mathcal{P}(P_o) \quad \text{with} \quad Y \longmapsto \{o \mid \forall y \in Y\ o \models_P y\}.$$

A subset $A \subseteq P_a$ is called a (formal) concept (of attributes) if it is a fixed point of $\alpha \circ \omega$, i.e., $\alpha(\omega(A)) = A$. Dually, a subset $X \subseteq P_o$ is called a (formal) concept (of objects) if it is a fixed point of $\omega \circ \alpha$.

The functions α and ω are dependent on P and we will use subscripts to avoid confusion. Readers familiar with FCA will notice that our notation differs from the standard notation in FCA which collapses both α_P and ω_P to a single $(\ \ )'$ without the possibility of using subscripts. The elaboration of notation to a less context-sensitive one makes it more expressive and accurate (try to restate some of the results in this paper using $(\ \ )'$ only!). The following is a fundamental theorem in formal concept analysis.

Theorem 3.5 (Wille) *With respect to a Chu space $P = (P_o, \models_P, P_a)$, the pair (α, ω) forms a Galois connection. As a consequence, we have*

- *The set of attribute (object) concepts of P forms a closure system.*
- *The attribute (object) concepts of P under set inclusion form a complete lattice.*
- *The lattice of attribute concepts and the lattice of object concepts are anti-isomorphic to each other.*

Notation. From now on, we write $\mathcal{L}P$ for the complete lattice of formal concepts associated with a Chu space P.

Example 3.6 Here is an example which shows that Chu-mappings do not necessarily preserve concepts.

P	a	b
1	×	×
2	×	

Q	a	b
1	×	×
2	×	×

Define $f : P_a \to Q_a$ to be the identity map, and $g : Q_o \to P_o$ to be the constant map $\lambda x.1$. One can readily check that the pair (f, g) so defined gives a Chu-mapping from P to Q. Note that while $\{a\}$ is a concept of P, $f^+(\{a\}) = \{a\}$ is not a concept of Q.

The following result, first pointed out in [13], serves as a starting point to understand how Chu-mappings interact with concept lattices.

Proposition 3.7 *Let (f, g) with $f : P_a \to Q_a$ and $g : Q_o \to P_o$ be a Chu mapping from $(P_o, \models_P, P_a)$ to $(Q_o, \models_Q, Q_a)$. We have*

$$\alpha_P \circ g^+ = f^- \circ \alpha_Q \quad \text{and} \quad \omega_Q \circ f^+ = g^- \circ \omega_P.$$

Notation. We use $\sqsubseteq$ to denote the extensional order of functions on sets with respect to inclusion.

Proposition 3.8 *With respect to any Chu-mapping (f, g) from $(P_o, \models_P, P_a)$ to $(Q_o, \models_Q, Q_a)$ with $f : P_a \to Q_a$ and $g : Q_o \to P_o$, the following statements are true:*

- $g^+ \circ g^- \circ \omega_P \sqsubseteq \omega_P$,
- $\alpha_P \circ g^+ \circ g^- \circ \omega_P \sqsupseteq \alpha_P \circ \omega_P$,
- $\alpha_Q \circ \omega_Q \circ f^+ \sqsupseteq f^+ \circ \alpha_P \circ \omega_P$,
- $\alpha_P \circ \omega_P \circ f^- \sqsubseteq f^- \circ \alpha_Q \circ \omega_Q$,
- $\omega_P \circ \alpha_P \circ g^+ \sqsubseteq g^+ \circ \omega_Q \circ \alpha_Q$,
- $\omega_Q \circ \alpha_Q \circ g^- \sqsupseteq g^- \circ \omega_P \circ \alpha_P$.

The next two propositions identify some conditions under which concepts are preserved under Chu-mappings.

Proposition 3.9 *Let (f, g) with $f : P_a \to Q_a$ and $g : Q_o \to P_o$ be a mapping from $(P_o, \models_P, P_a)$ to $(Q_o, \models_Q, Q_a)$, as defined on Chu spaces. If both f and g are onto (surjective) then we have:*

- *f^+ maps (attribute) concepts over P to (attribute) concepts over Q;*
- *f^- maps (attribute) concepts over Q to (attribute) concepts over P;*
- *g^+ maps (object) concepts over Q to (object) concepts over P;*
- *g^- maps (object) concepts over P to (object) concepts to Q.*

Proposition 3.10 *Let (f, g) with $f : P_a \to Q_a$ and $g : Q_o \to P_o$ be a mapping from $(P_o, \models_P, P_a)$ to $(Q_o, \models_Q, Q_a)$, as defined on Chu spaces. The following statements are true.*

- *If f is injective (one-to-one) and g is surjective (onto), then X is a concept if f^+X is a concept.*
- *If g is injective and f is surjective, then Y is a concept if g^+Y is a concept.*
- *If both f and g are surjective, then B is a concept if f^-B is a concept, and X is a concept if g^-X is a concept.*

4 Concept lattices and information systems

The converse of Theorem 3.5 is also true: every complete lattice is isomorphic to a concept lattice of a Chu space. This is a standard result in FCA; we provide a proof in terms of the formulation used in this paper to make it self-contained. The proof uses the idea of "open sets as properties". These open sets are called *Alexandrov open* [12].

Theorem 4.1 (Representation Theorem) *For every complete lattice D, there is a Chu space P such that D is order-isomorphic to $\mathcal{L}P$.*

Proof. Suppose $(D, \sqsubseteq)$ is a complete lattice. Define the Chu space $P = (P_o, \models, P_a)$, where $P_o = P_a = D$, and $x \models b$ iff $b \sqsubseteq x$. We want to show that $\mathcal{L}P$ is order-isomorphic to $(D, \sqsubseteq)$.

First note that for any $X \subseteq P_a$, we have

$$\begin{aligned} \omega X &= \{o \mid \forall x \in X, o \models x\} \\ &= \{o \mid \forall x \in X, x \sqsubseteq o\} \\ &= \{o \mid \bigsqcup X \sqsubseteq o\} \\ &= \uparrow(\bigsqcup X). \end{aligned}$$

On the other hand,

$$\begin{aligned} \alpha Y &= \{a \mid \forall y \in Y, y \models a\} \\ &= \{a \mid \forall y \in Y, a \sqsubseteq y\} \\ &= \{a \mid \forall y \in Y, a \in \downarrow y\} \\ &= \bigcap \{\downarrow y \mid y \in Y\}. \end{aligned}$$

Therefore, X is a concept iff $(\alpha \circ \omega)X \subseteq X$, or

$$\bigcap \{\downarrow y \mid y \in \uparrow(\bigsqcup X)\} \subseteq X.$$

Since $\bigcap \{\downarrow y \mid y \in \uparrow(\bigsqcup X)\} = \downarrow \bigsqcup X$, this is equivalent to saying that $\downarrow \bigsqcup X \subseteq X$. Hence, $X \subseteq P_a$ is a concept iff $X = \downarrow \bigsqcup X$. In other words, concepts of $(D, \sqsupseteq, D)$ are precisely the down-closed subsets of D generated by a single element.

Since for each $x, y \in D$, $x \sqsubseteq y$ iff $\downarrow x \subseteq \downarrow y$, the mapping $x \longmapsto \downarrow x$ provides an order-isomorphism between D and $\mathcal{L}P$. □

A few special cases may be worth noting. First, D, being the down-closure of the top element, is always a concept. On the other hand, the empty set does not qualify as a concept because $(\alpha \circ \omega)\emptyset = \{\bot\}$.

We now move to Scott's notion of *information systems* [28] which provide a concrete representation of Scott domains, with a logical flavor.

An information system consists of a set A of tokens, a subset Con of the set of finite subsets of A, denoted as $\mathrm{Fin}(A)$, and a relation $\vdash$ between Con and A. The subset Con on A is often called the consistency predicate, and the relation $\vdash$ is called the entailment relation. Both the consistency predicate and the entailment relation satisfy some routine axioms, made precise in the following definition.

Definition 4.2 An information system $\underline{A}$ is a triple $(A, \mathit{Con}, \vdash)$, where

- A is the token set,
- Con is the consistency predicate ($\mathit{Con} \subseteq \mathrm{Fin}(A)$ and $\emptyset \in \mathit{Con}$),
- $\vdash$ is the entailment relation ($\vdash \subseteq \mathit{Con} \times A$).

Moreover, the consistency predicate and entailment relation satisfy the following properties:

- $X \subseteq Y \;\&\; Y \in \mathit{Con} \Rightarrow X \in \mathit{Con}$,
- $a \in A \Rightarrow \{\, a \,\} \in \mathit{Con}$,
- $X \vdash a \;\&\; X \in \mathit{Con} \Rightarrow X \cup \{\, a \,\} \in \mathit{Con}$,
- $a \in X \;\&\; X \in \mathit{Con} \Rightarrow X \vdash a$,
- $(\forall b \in Y.\, X \vdash b) \;\&\; Y \vdash c \Rightarrow X \vdash c$.

Although monotonicity for $\vdash$ is not explicitly given, it is a derivable property. The notion of consistency can be easily extended to arbitrary token sets by enforcing compactness, i.e., a set is consistent if every finite subset of it is consistent. By overloading notation, we write $X \in \mathit{Con}$ when every finite subset of X is consistent; hence the consistency of finite sets is a more primitive notion.

An information system $(A, \mathit{Con}, \vdash)$ induces an operator $F : \mathit{Con} \to \mathit{Con}$, given as

$$F(X) := \{a \mid \exists Y (Y \subseteq^{\mathrm{fin}} X \;\&\; Y \vdash a)\}.$$

(Here, $\subseteq^{\mathrm{fin}}$ stands for "finite subset of", and X need not be finite.) It follows from the properties of an information system that F is a closure operator in a *generalized sense*: it is inflationary, monotonic, and idempotent. However, this is not strictly a closure operator because F is defined on Con, instead of $\mathcal{P}(A)$.

The *information states*, or simply *states*, of an information system are sets of the form $F(X)$ with $X \in \mathit{Con}$, where Con is understood in the generalized sense to include infinite sets through the compactness condition. Information states consist of consistent, deductively closed (under $\vdash$) sets of tokens. The importance of information systems lies in the fact that they provide a logical representation of Scott domains.

Theorem 4.3 (Scott) *For any information system $\underline{A}$, the collection of its* *information states under set inclusion forms a Scott domain. Conversely, every Scott domain is order-isomorphic to the partial order of information states of some information system.*

A Chu space determines an information system in the following way.

Definition 4.4 For a given Chu space $P = (P_o, \models_P, P_a)$, define a system $(A_P, Con_P, \vdash_P)$ with $A_P = P_a$ as follows. Write $x \models_P X$ if $x \models_P b$ for each $b \in X$. For a finite set X of attributes and an attribute a, define $X \vdash_P a$ if

$$\forall x \in P_o\ (x \models_P X \Rightarrow x \models_P a).$$

The consistency predicate Con_P is the trivial one: every subset of P_a is consistent.

Lemma 4.5 *For a given Chu space $P = (P_o, \models_P, P_a)$ and the derived relation $\vdash_P$ given in the previous definition, we have*

$$X \vdash_P a \text{ if and only if } a \in \alpha_P \circ \omega_P(X).$$

Proof.

$$\begin{aligned} X \vdash_P a &\Longleftrightarrow \forall x \in P_o\ (x \models_P X \Rightarrow x \models_P a) \\ &\Longleftrightarrow \omega_P(X) \subseteq \omega_P(\{a\}) \\ &\Longleftrightarrow \alpha_P \circ \omega_P(\{a\}) \subseteq \alpha_P \circ \omega_P(X) \\ &\Longleftrightarrow a \in \alpha_P \circ \omega_P(X) \end{aligned}$$

□

Proposition 4.6 *Given a Chu space $P = (P_o, \models_P, P_a)$, the triple*

$$(A_P, Con_P, \vdash_P)$$

is an information system.

The proof goes by checking each axiom of an information system, which is straightforward. The interesting question is whether a set of attributes is a concept if and only if it is an information state.

Theorem 4.7 *Given a Chu space $P = (P_o, \models_P, P_a)$ with P_a a finite set, $X \subseteq P_a$ is a concept if and only if it is a state of the derived information system $(A_P, Con_P, \vdash_P)$.*

Proof. Suppose X is a concept. We show that it is *deductively closed*, i.e., for each a and each $Y \subseteq X$, $Y \vdash_P a$ implies $a \in X$. Since X is a concept, we have $\alpha_P \circ \omega_P(X) = X$. Suppose $Y \subseteq X$ and $Y \vdash_P a$. By Lemma 4.5, we have $a \in \alpha_P \circ \omega_P(Y)$. Since $\alpha_P \circ \omega_P(Y) \subseteq \alpha_P \circ \omega_P(X)$ and $\alpha_P \circ \omega_P(X) = X$, we get $a \in X$.

For the other direction, suppose T is a deductively closed set, i.e., for any $a \in P_a$, $T \vdash_P a$ implies $a \in T$. By Lemma 4.5 again, $T \vdash_P a$ means $a \in \alpha_P \circ$

$\omega_P(T)$. Therefore, the deductive closeness of T implies that $\alpha_P \circ \omega_P(T) \subseteq T$, and T is a concept. $\square$

The finiteness of P_a is needed for the second part of the proof. Every concept is a deductively closed set, but an infinite deductively closed set is not necessarily a concept, as our example below shows.

Note that finiteness is a severe restriction theoretically. This restriction represents a conceptual mismatch, rather than a technical shortcoming.

Note also that, by the earlier Representation Theorem 4.1, any information system $(A, Con, \vdash)$ with a trivial consistency predicate determines a Chu space which determines an isomorphic concept lattice. A simpler and more direct construction exists. The required Chu space can be defined as follows. We can take information states x as objects, and tokens $a \in A$ as attributes, and let $x \models a$ iff a is a member of x. With respect to this Chu space, the extension of a set of attributes B is the set $[\![B]\!] := \{x \mid B \subseteq x\}$, where x is an information state; the intension of $[\![B]\!]$ is the set $\{a \mid \forall x \in [\![B]\!],\ a \in x\}$. The requirement that B matches the intension of the extension of B can be stated precisely as $B = \{a \mid \forall x \in [\![B]\!],\ a \in x\}$, which amounts to the statement that B is an information state of the original information system $(A, Con, \vdash)$, observe that the information state generated by B is precisely the intension of the extension of B.

From another perspective, the correspondence between *information states* and *formal concepts* breaks down in the infinite case because information systems represent Scott domains, which are *algebraic*; on the other hand, concept lattices, though bounded complete, need not be algebraic, as the following example shows.

Example 4.8 We follow the construction given in the proof of Theorem 4.1 to construct a Chu space whose concepts form a lattice isomorphic to the one pictured below. This lattice is not algebraic even if one turns it up side down.

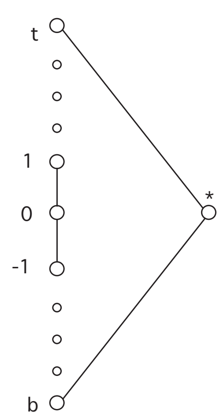

Fig. 1. A complete lattice which is not algebraic

According to the proof of Theorem 4.1, we obtain the Chu space given by the following table

P	$\uparrow t$	$\uparrow b$	$\uparrow *$	$\uparrow 0$	$\uparrow 1$	$\uparrow 2$	$\cdots$	$\uparrow -1$	$\uparrow -2$	$\uparrow -3$	$\cdots$
t	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\cdots$	$\times$	$\times$	$\times$	$\cdots$
b		$\times$					$\cdots$				$\cdots$
$*$		$\times$	$\times$				$\cdots$				$\cdots$
0		$\times$		$\times$			$\cdots$	$\times$	$\times$	$\times$	$\cdots$
1		$\times$		$\times$	$\times$		$\cdots$	$\times$	$\times$	$\times$	$\cdots$
2		$\times$		$\times$	$\times$	$\times$		$\times$	$\times$	$\times$	$\cdots$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
-1		$\times$						$\times$	$\times$	$\times$	$\cdots$
-2		$\times$							$\times$	$\times$	$\cdots$
-3		$\times$								$\times$	$\cdots$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

With respect to its corresponding information system, note that for any finite subset X of $\{\uparrow i \mid i \geq 0\} \subseteq P_a$, we do not have $X \vdash *$ since $*$ is not a member in $\alpha \circ \omega(X)$ (by Lemma 4.5). However, $*$ *is* a member of $\alpha \circ \omega\{\uparrow i \mid i \geq 0\}$ and hence it is a member of the concept generated by $\{\uparrow i \mid i \geq 0\}$. In other words, we need "infinitary" implication $\{\uparrow i \mid i \geq 0\} \vdash *$ to capture a concept, but this cannot be captured by finitary entailment relations. In a sense, compactness fails here.

We can also see this from the proof of Theorem 4.1. While $*$ is a member of

$$\downarrow \bigcap \{\uparrow i \mid i \geq 0\} = \downarrow t,$$

$t \notin \downarrow(\bigcap X)$ for any finite subset X of $\{\uparrow i \mid i \geq 0\}$.

One can further observe that the composition $\alpha \circ \omega$, though monotonic, is not continuous.

We mention without proof a few conditions under which the concept lattice is algebraic.

Proposition 4.9 *For any context P, its corresponding concept lattice $\mathcal{L}P$ is algebraic if and only if $\alpha_P \circ \omega_P$ is continuous.*

The function $\alpha_P \circ \omega_P$ is continuous if there are no infinitely-checked columns.

Proposition 4.10 *For any context P, $\alpha_P \circ \omega_P$ is continuous if for each $a \in P_a$, $\omega_P\{a\}$ is a finite set.*

Proposition 4.11 *For any context P, $\alpha \circ \omega$ is continuous if the set $\{\omega Y \mid Y \text{ finite}\}$ is well-founded.*

Proposition 4.12 *For any context P, $\alpha \circ \omega$ is continuous iff for any $b \in P_a$ and any $u \subseteq P_a$, $b \in (\alpha \circ \omega)u$ implies $b \in (\alpha \circ \omega)X$ for some finite subset X of u.*

5 Towards data-mining applications

Many data sets are, or can be put into the form of, Chu spaces. In this section we discuss some of the implications of our earlier results in the area of data-mining and knowledge discovery. Pfaltz and his collaborators [18,19] have done some interesting work on the minimal representation in concept lattices.

The first observation, although simple, is helpful. It is a well-known fact in formal concept anaylsis.

Proposition 5.1 *Let $P = (P_o, \models_P, P_a)$ be a Chu space. For each object $x \in P_o$, the set of its attributes $\alpha_P\{x\}$ is a concept.*

This gives immediate structural information about concept lattices: the set of attributes collected from each row always forms a concept and, moreover, any intersection of a subset of these concepts forms another concept.

Proposition 5.2 *Let $P = (P_o, \models_P, P_a)$ be a Chu space, and let Chu space $Q = (Q_o, \models_Q, Q_a)$ be a structure obtained from P by adding a row without changing the attribute set, i.e.,*

- *$Q_o = P_o \cup \{n\}$ where n is the "new" object,*
- *$P_a = Q_a$,*
- *$\models_P = \models_Q$ when restricted to $P_o \times P_a$.*

Then the function pair (f, g) with f the identity function on Q_a and g the injection $P_o \to Q_o$ with $x \mapsto x$ is a Chu mapping from Q to P.

This is a more precise statement of Example 3.1. The upshot of it is that since f is surjective and g is injective, we have, by Theorem 3.3, item 2, Y is a concept if $g^+(Y)$ is a concept, for any $Y \subseteq P_o$. We can state this more precisely in terms of attribute concepts, as follows.

Proposition 5.3 *Let $P = (P_o, \models_P, P_a)$ and $Q = (Q_o, \models_Q, Q_a)$ be Chu spaces as given in the previous proposition: Q extends P by a row. Then every attribute concept A of P is also an attribute concept of Q.*

Of course, transitivity allows us to generalize this result to the case when Q is an extension of P by adding rows, and this provides the foundation for an iterative, "on-line" construction of concept lattices when the attribute set is fixed (which is usually the case), but new data keep coming in, not all at once.

5.1 A closure-system centric view

The closure-system point of view is equivalent to that of Galois connections for concept lattices. However, closure systems sometimes provide a more straightforward theoretic basis for data-mining algorithms.

Lemma 5.4 *Let A be a set. Then the set of all closure systems over A forms a (meta) closure system over $\mathcal{P}(A)$.*

For closure systems $\mathcal{C}_1$ and $\mathcal{C}_2$ over A, let $\mathcal{C} := \mathcal{C}_1 \cap \mathcal{C}_2$. One can check that $\mathcal{C}$ is again a closure system over A. In general, intersection preserves closure systems, and the intersection of an empty collection of closure systems over A is the largest closure system $\mathcal{P}(A)$ over A.

As an immediate consequence of this lemma, any subset of the powerset $\mathcal{P}(A)$ generates a closure system.

Lemma 5.5 *Every subset of $\mathcal{P}(A)$ generates a closure system over A, which is the smallest closure system containing the starting subset.*

We can then view concept lattices as a generated closure system.

Proposition 5.6 *Let $P = (P_o, \models_P, P_a)$ be a Chu space. Then its concept lattice $\mathcal{L}P$ is isomorphic to the closure system generated by the set $\{\alpha_P\{x\} \mid x \in P_o\}$. Dually, $\mathcal{L}P$ anti-isomorphic to the closure system generated by the set $\{\omega_P\{p\} \mid p \in P_a\}$.*

This brings flexibility for procedures for constructing concept lattices. For example, one can partition P_o into $A \cup B = P_o$, find the closure system generated by $\{\alpha_P\{x\} \mid x \in A\}$ and $\{\alpha_P\{x\} \mid x \in B\}$, respectively, and then find the closure system generated by the union of the two closure systems. This view provides an easy-to-understand, straightforward way to justify the correctness of many FCA related algorithms in the literature (for which correctness proofs are not always provided).

6 Conclusions and future work

This paper has brought three relatively independent areas together: Chu spaces, Formal Concepts, and Domains. The formulation and results provided here serve as a basis for many opportunities for further development, in a number of directions. For example:

The notion of morphisms on general concept lattices needs to be further explored, keeping Chu-mapping as a yard-stick. Continuous functions on finite lattices may be the starting point, because there is a standard way to define continuous functions on Scott's information systems, as *approximable mappings*.

Since the notion of *contexts* in FCA is the same as a special form of Chu spaces, and Chu spaces come with well-defined and robust notions of morphisms, it would be desirable to provide a categorical account of the construction $\mathcal{L}$ from Chu spaces to complete lattices. For example, can $\mathcal{L}$ be seen as a part of a coreflection between a certain category of Chu spaces and a certain category of complete lattices? Work reported in [14,16] may be useful for understanding this categorical aspect.

Attributes in real-world data rarely come without some preliminary structural information. For example, one attribute many be in *conflict* with another, for scientific or logical reasons (case in point: "four-legged animal" and "two-legged animal"). In general, objects having all the attributes are not interesting, even if they exit. It would be beneficial to explore a notion of formal concepts with the constraint of a consistency predicate, in the spirit of information systems. This will bring concept structures to cpos without necessarily a top element, such as bounded complete cpos, if not Scott domains.

A rich collection of constructions on Chu spaces exist, for the purpose of modeling linear logic. It would be interesting to see which of those constructions can be useful for exploring the structure of data, and how and if concepts are preserved with respect to these constructions.

In general, logical systems for reasoning about concept structures may be profitably developed using a similar approach as "logic of domains", or semantics-based proof systems [1,8,12,30,32]. The information-flow theory of Barwise [5] has already taken a step in this direction.

Last and maybe most, there is an explosion of research on ontology and ontological engineering over the last couple of years, sparked by the Semantic Web initiative [6]. There seems to be a great deal of potential in exploring FCA, information systems, and Chu constructions in this context, especially with respect to the understanding of ontological structures and automated learning of ontology from the Web.

Acknowledgment. Thanks go to Pascal Hitzler and John Pfaltz for stimulating discussions related to formal concept analysis, and to Austin Melton for pointers to Galois connections.

References

[1] S. Abramsky and A. Jung, *Domain theory.* In: Handbook of Logic in Computer Science (S. Abramsky, D. Gabbay, and T. S. E. Maibaum, eds.), Vol. 3, Clarendon Press, 1994.

[2] R. Amadio and P.-L. Curien, *Domains and Lambda-Calculi.* Cambirdge University Press, 1998.

[3] M. Barr. *-Autonomous categories, with an appendix by Po Hsiang Chu. *Lecture Notes in Mathematics*, Vol. 752, Springer-Verlag, 1979.

[4] M. Barr. *-Autonomous categories and linear logic. *Mathematical Structures in Computer Science*, Vol. 1, pp. 159-178, 1991.

[5] J. Barwise and J. Seligman. *Information Flow: the logic of distributed systems.* Cambridge Tracts in TCS, Vol. 44, 1997.

[6] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. *Scientific American*, Vol. 284, pp. 34-43, 2001.

[7] G. Birkhoff. Lattice Theory. AMS Colloquium Publications, Vol. 25, 1940.

[8] T. Coquand and G. Q. Zhang, Sequents, frames, and completeness, *Computer Science Logic 2000* (P. Clote and H. Schwichtenberg, eds.), *Lecture Notes in Computer Science*, Vol. 1862, pp. 277-291, 2000.

[9] C. Gunter and D. Scott. *Semantic domains*. In Jan van Leeuwen, editor, Handbook of Theoretical Computer Science, Vol. B, pp. 633-674, Elsevier, 1990.

[10] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.

[11] P. Hitzler. Contexts, concepts, and logic of domains. Technical Report WV-02-12, AI Institute, Dresden University of Technology.

[12] P. Johnstone. *Stone Spaces.* Cambridge University Press, 1982.

[13] F. Lamarche. From Chu spaces to cpos. *Theory and Formal Methods of Computing 94*, Imperial College Press, pp. 283-305, 1994.

[14] J. McDill, A. Melton, and G. Strecker. A category of Galois connections. *Lecture Notes in Computer Science*, Vol. 283, pp. 290-300, 1987.

[15] A. Melton, D. Schmidt, and G. Strecker. Galois connections and computer science applications. *Lecture Notes in Computer Science*, Vol. 240, pp. 299-312, 1985.

[16] A. Melton, B. Schröder, and G. Strecker. Lagois connections – a counterpart to Galois connections. *Theoretical Computer Science*, Vol. 136, pp. 79-107, 1994.

[17] S. MacLane. *Categories for the working mathematician.* Springer-Verlag, 1971.

[18] J. Pfaltz and C. Taylor. Scientific knowledge discovery through iterative transformation of concept lattices. *SIAM Workshop on Discrete Math. and Data Mining*, Arlington, VA, U.S.A., 2002.

[19] J. Pfaltz and C. Taylor. Closed set data mining of biological data. *BIOKDD 2002 (2nd Workshop on Data Mining in Bioinformatics)*, Edmonton, July, 2002.

[20] G. Plotkin. Notes on Domains (Pisa Notes), University of Edinburgh, 1983. Available at:
`http://www.dcs.ed.ac.uk/home/gdp/publications/`

[21] V. Pratt. Higher dimensional automata revisited. *Math. Structures Comput. Sci.* Vol. 10, pp. 525-548, 2000.

[22] V. Pratt. Chu spaces from the representational viewpoint. *Ann. Pure Appl. Logic*, Vol. 96, pp. 319-333, 1999.

[23] V. Pratt. Towards full completeness of the linear logic of Chu spaces. *Mathematical foundations of programming semantics* (Pittsburgh, PA, 1997), *Electronic Notes in Theoretical Computer Science*, Vol. 7, 18 pp., 1997.

[24] V. Pratt. The Stone gamut: a coordinatization of mathematics. *Proceedings of 10th Annual Symposium on Logic in Computer Science*, pp. 444-454, 1995.

[25] V. Pratt. Chu spaces and their interpretation as concurrent objects. *Lecture Notes in Comput. Sci.* Vol. 1000, pp. 392-405, 1995.

[26] V. Pratt. Chu spaces as a semantic bridge between linear logic and mathematics. *Theoretical Computer Science*, Vol. 294, pp. 439-471, 2003.

[27] W. Rounds and G.-Q. Zhang. Clausal logic and logic programming in algebraic domains. *Information and Computation*, Vol. 171, pp. 183-200, 2001.

[28] D. S. Scott. Domains for denotational semantics, *Lecture Notes in Computer Science*, Vol. 140, pp. 577-613, 1982.

[29] R.A.G. Seely. Linear logic, *-autonomous categories and cofree algebras. In *Categories in Computer Science and Logic*, *Contemporary Mathematics*, Vol. 92, pp. 371-382, 1989.

[30] S. Vickers. *Topology via Logic*, Cambridge University Press, 1989.

[31] G. Winskel. *The Formal Semantics of Programming Languages.* MIT Press, 1993.

[32] G.-Q. Zhang. *Logic of Domains*, Birkhauser, Boston, 1991.