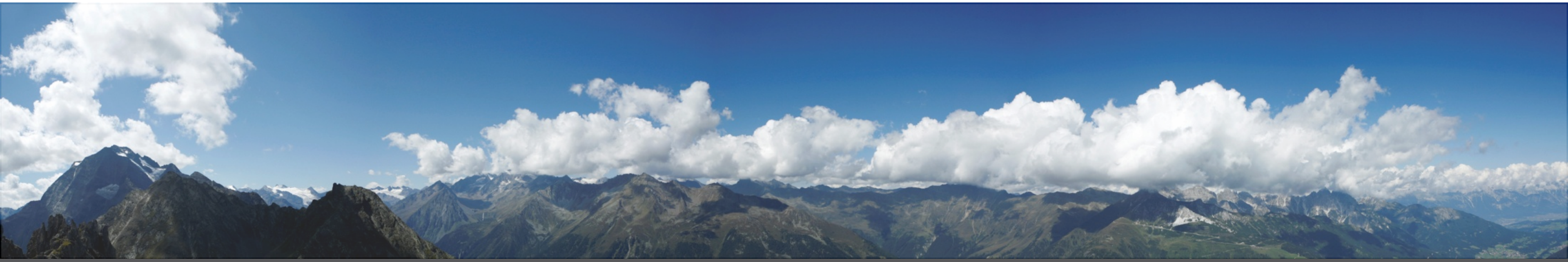
Intelligent Systems

Lecture XII – xx 2009
Formal Concept Analysis

Dieter Fensel and Federico Facca



Where are we?

STI · INNSBRUCK

#	Date	Title
1		Introduction
2		Propositional Logic
3		Predicate Logic
4		Theorem Proving, Logic Programming, and Description Logics
5		Search Methods
6		CommonKADS
7		Problem Solving Methods
8		Planning
9		Agents
10		Rule Learning
11		Inductive Logic Programming
12		**Formal Concept Analysis**
13		Neural Networks
14		Semantic Web and Exam Preparation

Outline

- Motivation
- Technical Solution
  - Introduction and Definitions
  - Attribute Exploration
  - Many-Valued Context
- Illustration by a Larger Example
- Extension
- Summary

From data to their conceptualization

MOTIVATION

Introduction

STI · INNSBRUCK

- In many applications we deal with huge amount of data
  - E.g. insurance company records
- Data by itself are not useful to support decisions
  - E.g. can I make an insurance contract to this new customer or it is too risky?
- Thus we need a set of methods to generate from a data set a "summary" that represent a conceptualization of the data set
  - E.g. what are similarities among different customers of an insurance company that divide them in different risk classes?
  - Age >25, City=Innsbruck => Low Risk
- This is a common task that is needed in several domains to support data analysis
  - Analysis of children suffering from diabetes
  - Marketing analysis of store departments or supermarkets
- Formal Concept Analysis is a technique that enables resolution of such problems

Formal Concept Analysis

TECHNICAL SOLUTION

What is a concept?

STI · INNSBRUCK

- What drives us to call an object a “bird”?
- Every object having certain attributes is called “bird”:
  - A bird has feathers
  - A bird has two legs
  - A bird has a bill
  - …
- All objects having these attributes are called “birds”:
  - Duck, goose, owl and parrot are birds
  - Penguins are birds, too
  - …

What is a concept?

- This description of the concept "bird" is based on sets of

- Objects, attributes and a relation form a formal concept

What is a concept?

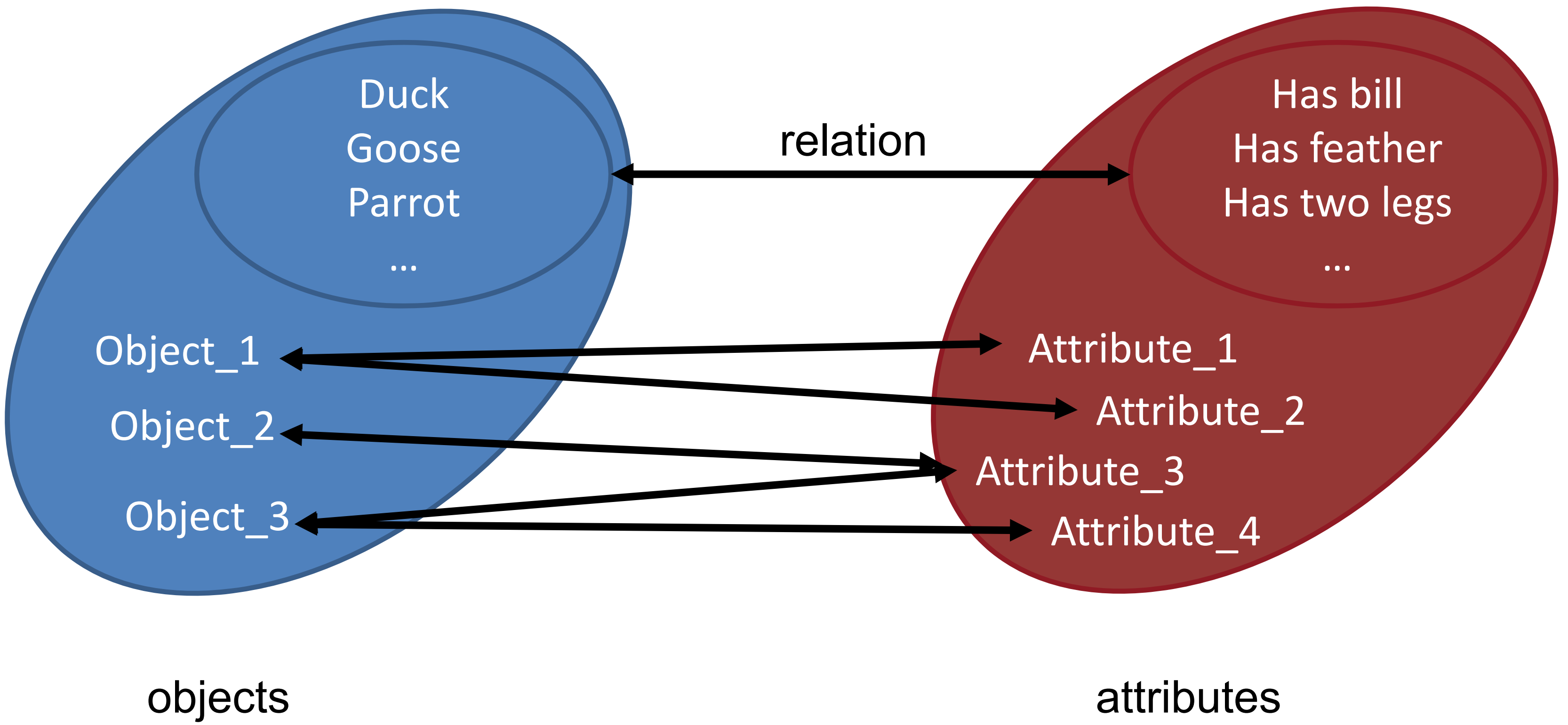
- So, a formal concept is constituted by two parts

- having a certain relation:
  - every object belonging to this concept has all the attributes in B
  - every attribute belonging to this concept is shared by all objects in A
- A is called the concept's extent, B is called the concept's intent

The Universe of Discourse

STI · INNSBRUCK

- A repertoire of objects and attributes (which might or might not be related) constitutes the „context“ of our considerations



Formal Concept Analysis

- **Formal Concept Analysis** is a method used for investigating and processing explicitely given information, in order to allow for meaningful and comprehensive interpretation
  - An **analysis** of data
  - Structures of formal abstractions of **concepts** of human thought
  - **Formal** emphasizes that the concepts are mathematical objects, rather than concepts of mind

Formal Concept Analysis

STI · INNSBRUCK

- Formal Concept Analysis takes as input a matrix specifying a set of objects and the properties thereof, called attributes, and finds both all the “natural” clusters of attributes and all the “natural” clusters of objects in the input data, where
  - a “natural” object cluster is the set of all objects that share a common subset of attributes, and
  - a “natural” property cluster is the set of all attributes shared by one of the natural object clusters
- Natural property clusters correspond one-for-one with natural object clusters, and a concept is a pair containing both a natural property cluster and its corresponding natural object cluster
- The family of these concepts obeys the mathematical axioms defining a lattice, and is called a concept lattice

Definition: Formal Context

STI · INNSBRUCK

- **Context:** A triple (G, M, I) is a (formal) context if
  - G is a set of **objects** (Gegenstand)
  - M is a set of **attributes** (Merkmal)
  - I is a binary relation between G and M called **incidence**

- **Incidence** := $\mathrm{I} \subseteq G \times M$
  - if $g \in G$, $m \in M$ in $(g,m) \in \mathrm{I}$, then we know that "*object g has attribute m*„ and we write $g\mathrm{I}m$.

Definition: Formal Concept

STI · INNSBRUCK

- A pair (A,B) is a **formal concept** of (G,M,I) if and only if
  - $A \subseteq G$
  - $B \subseteq M$
  - A‘ = B, and A = B‘

- Note that at this point the **incidence relationship is closed**; i.e. all objects of the concept carry all its attributes and that there is no other object in G carrying all attributes of the concept

- A is called the **extent** (Umfang) of the concept (A,B), while
- B is called the **intent** (Inhalt) of the concept (A,B)

Definition: Concept Lattice

STI · INNSBRUCK

- The concepts of a given context are naturally ordered by a subconcept-superconcept relation:
  - $(A1,B1) \leq (A2,B2) :\Leftrightarrow A1 \subseteq A2$ ($\Leftrightarrow B2 \subseteq B1$)

- The ordered set of all formal concepts in (G,M,I) is denoted by $\mathfrak{B}(G,M,I)$ and is called **concept lattice** (Begriffsverband)

- A concept lattice consists of the set of concepts of a formal context and the subconcept-superconcept relation between the concepts

Generating Formal Concepts

- Using the derivation operators we can derive formal concepts from our formal context with the following routine:

1. Pick a set of objects A
2. Derive the attributes A'
3. Derive (A')'
4. (A'',A') is a formal concept

- A dual approach can be taken starting with an attribute

Example (1)

STI · INNSBRUCK

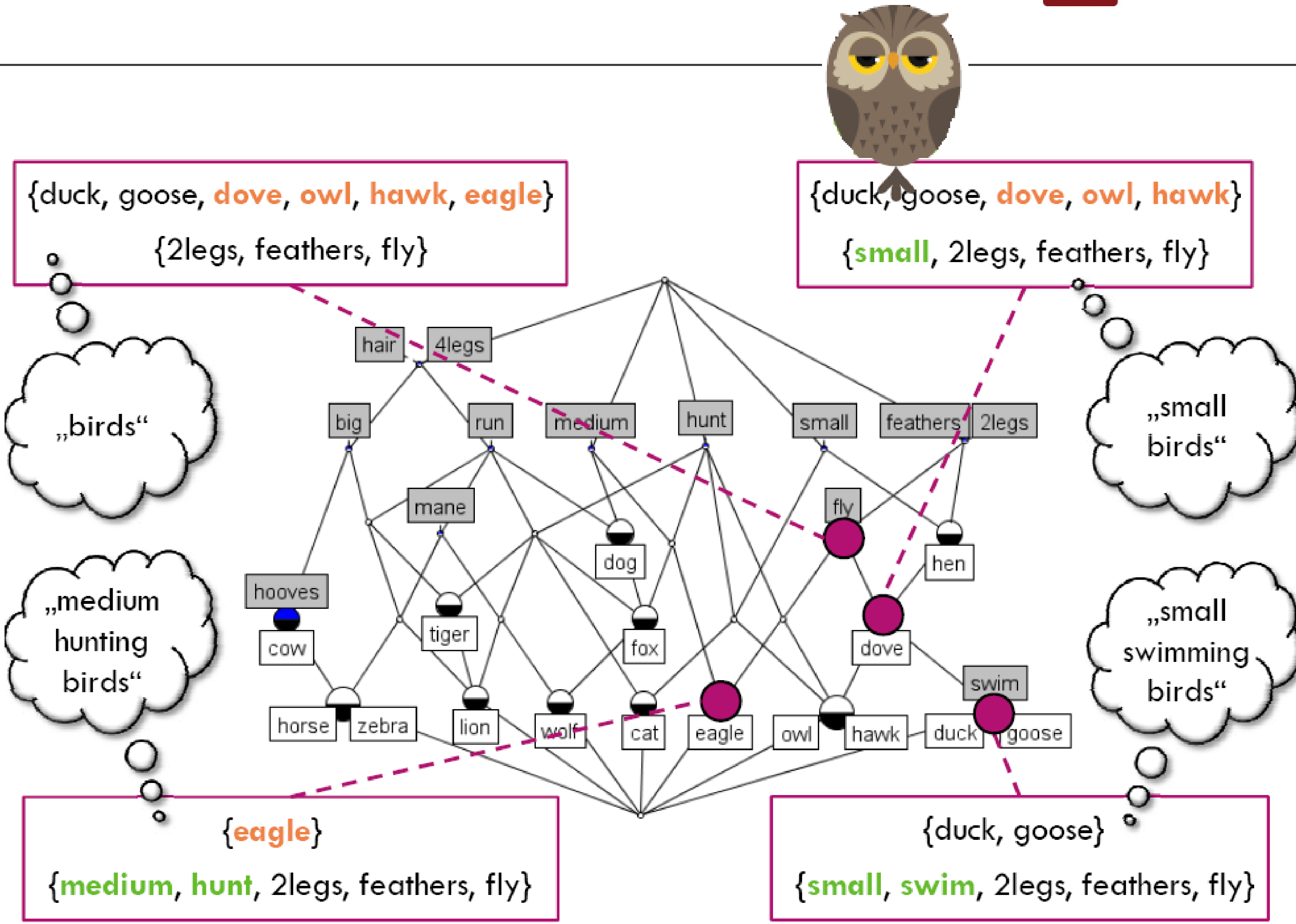
	small	medium	big	twolegs	fourlegs	feathers	hair	fly	hunt	run	swim	mane	hooves
dove	x	.	.	x	.	x	.	x	.	.	.	.	.
hen	x	.	.	x	.	x	.	.	.	.	.	.	.
duck	x	.	.	x	.	x	.	x	.	.	x	.	.
goose	x	.	.	x	.	x	.	x	.	.	x	.	.
owl	x	.	.	x	.	x	.	x	x	.	.	.	.
hawk	x	.	.	x	.	x	.	x	x	.	.	.	.
eagle	.	x	.	x	.	x	.	x	x	.	.	.	.
fox	.	x	.	.	x	.	x	.	x	x	.	.	.
dog	.	x	.	.	x	.	x	.	.	x	.	.	.
wolf	.	x	.	.	x	.	x	.	x	x	.	x	.
cat	x	.	.	.	x	.	x	.	x	x	.	.	.
tiger	.	.	x	.	x	.	x	.	x	x	.	.	.
lion	.	.	x	.	x	.	x	.	x	x	.	x	.
horse	.	.	x	.	x	.	x	.	.	x	.	x	x
zebra	.	.	x	.	x	.	x	.	.	x	.	x	x
cow	.	.	x	.	x	.	x	.	.	.	.	.	x

1. Pick any set of objects A, e.g. A={duck}.
2. Derive the attributes A'={small, two legs, feathers, fly, swim}
3. Derive (A')'={small, two legs, feathers, fly, swim}'={duck, goose}
4. (A'',A')=({duck, goose},{small, two legs, feathers, fly, swim}) is a *formal concept.*

[Bastian Wormuth and Peter Becker, Introduction to Formal Concept Analysis, 2nd International Conference of Formal Concept Analysis]

Example (2)

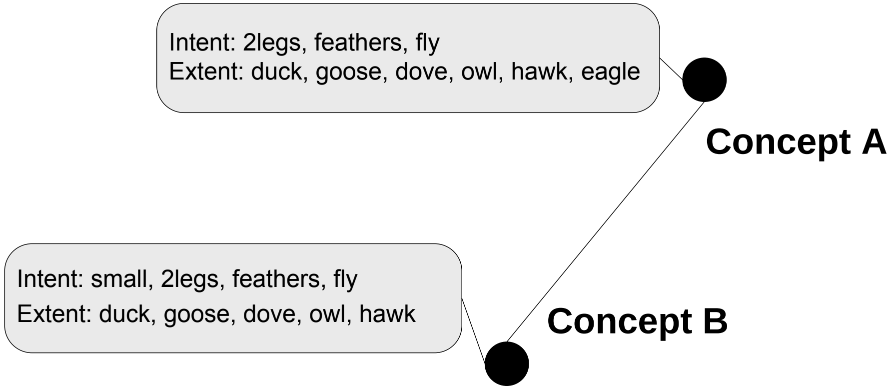
STI · INNSBRUCK



Extent and Intent in a Lattice

- The extent of a formal concept is given by all formal objects on the paths which lead ***down*** from the given concept node
  - The extent of an arbitrary concept is then found in the **principle ideal** generated by that concept
- The intent of a formal concept is given by all the formal attributes on the paths which lead ***up*** from the given concept node
  - The intent of an arbitrary concept is then found in the **principle filter** generated by that concept

Subconcepts in the Concept Lattice



- The Concept B is a subconcept of Concept A because
  - The extent of Concept B is a subset of the extent of Concept A
  - The intent of Concept B is a superset of the intent of Concept A
- All edges in the line diagram of a concept lattice represent this subconcept-superconcept relationship

Reduction of Context

- A context (G,M,I) is called **clarified** if for g, h$\in$G and g'=h' it always follows that g=h and correspondingly, m'=n' implies m=n for all m,n$\in$M; i.e. a context is reduced, if both derivatives are injective.

- A clarified context (G,M,I) is called **row-reduced**, if every object concept is ∨-irreducible and **column-reduced**, if every attribute concept is ∧-irreducible. A context, which is both row-reduced and column-reduced is **reduced**.

- Reducing a context does not change the concept lattice!

- Always reducable are complete rows (objects g with g'=M) and complete columns (attributes m with m'=G).

Implication

STI · INNSBRUCK

- An **implication** $A \rightarrow B$ (between sets $A,B \in M$ of attributes) holds in a formal context if and only if $B \subseteq A$"
  - i.e. if every object that has all attributes in A also has all attributes in B
  - e.g. if X has feather and has bill then is a bird
- The implication determines the concept lattice up to isomorphism and therefore offers an additional interpretation of the lattice structure
- Implications can be used for a step-wise construction of conceptual knowledge; **attribute exploration** is a knowledge acquisition method that is used to acquire knowledge from a domain expert by asking successive questions

ATTRIBUTE EXPLORATION

Attribute Exploration

STI · INNSBRUCK

- Attribute exploration has proven to be a successful method for efficiently capturing knowledge, if it is only “known" to a domain expert
- Attribute exploration asks the expert questions of the form “*is it true that objects having attributes* $mi_1,\ldots,mi_k$ *also have the attributes* $mj_1,\ldots,mj_l$*?*"
- The expert either confirms the question, in which case a new implication of the application domain is found, or rejects it
  - If the expert rejects the question, a counterexample is given, i.e., an object that has all the attributes $mi_1,\ldots,mi_k$ but lacks at least one of $mj_1,\ldots,mj_l$
- The counterexample is then added to the application domain as a new object, and the next question is asked

Attribute Exploration (2)

- Attribute exploration is an attractive method for capturing expert knowledge in that it guarantees to make best use of the expert's answers, and to ask the minimum possible number of questions that suffices to acquire complete knowledge about the application domain

- A derivation of attribute exploration is the so-called **concept exploration** that aims at exploring sublattices of larger data sets in order to determine implications

Lexographic Ordering

STI · INNSBRUCK

- The **lexographic ordering** of sets of objects is given by:
  - for $A,B \subseteq G$, $i \in G = \{1, ..., n\}$ to order the objects
  - $A \prec_i B$ iff $i \in B - A$ and $A \cap \{1,...,i-1\} = B \cap \{1,...,i-1\}$
  - $A \prec B$ iff $\exists i$ such that $A \prec_i B$
- The $\prec$ denotes a lexographic ordering and thus, every two distinct sets $A, B \subseteq G$ are comparable
- $B \subseteq G$ can also be represented in terms of a characteristics vector:
  - $G = \{1,2,3,4,5\}$, $B=\{1,2,5\}$
  - **characteristics vector** of B = 11001

Construction of Concept Lattice

- This algorithm works on a finite context (G,M,I) with a lexographic ordering
- The lexographically smallest extent is ∅;
  - for i=1 we have {1,...,i-1} = ∅
- For an arbitrary X⊆G, one can find the lexographically next concept extent by checking all elements y ∈ G – X (beginning with the lexographically largest) until $X \prec_i X \oplus i$ for the first time
- $X \oplus i = ((X \cap \{1,...,i-1\}) \cup \{i\})''$
- $X \oplus i$ is the lexographically next extent

Construction of Concept Lattice (2)

- Note that due to the duality between the sets G and M, the same algorithm is analogously applicable by exchanging the set G by M

Example

STI · INNSBRUCK

- Consider a context with G = {1,2,3} and M = {a,b,c} and incidence I = {(1,{a,b}), (2,{a}), (3,{b,c})}:

1. ∅“ = {a,b,c}‘ = ∅

   ⇒ 1. extent: ∅

2. ∅ ⊕ 3 = ((∅ ∩ {1,2}) ∪ {3})“ = {3}“ = {b,c}‘ = {3} and

   $\emptyset <_3$ {3}, as 3 ∈ {3} - ∅ and ∅ ∩ {1,2} = {3} ∩ {1,2}

   ⇒ 2. extent: {3}

3. {3} ⊕ 2 = {2}“ = {a}‘ = {1,2} and {3} $\nless_2$ {1,2}

4. {3} ⊕ 1 = {1}“ = {a,b}‘ = {1} and {3} $<_1$ {1}

   ⇒ 3. extent: {1}

5. {1} ⊕ 3 = {1,3} and {1} $<_3$ {1,3}

   ⇒ 4. extent: {1,3}

From „Formale Begriffsanalyse“ by Martin Sonntag (Mai 2000)

Example (2)

6. $\{1,3\} \oplus 2 = \{1,2\}$ and $\{1,3\} \prec_2 \{1,2\}$

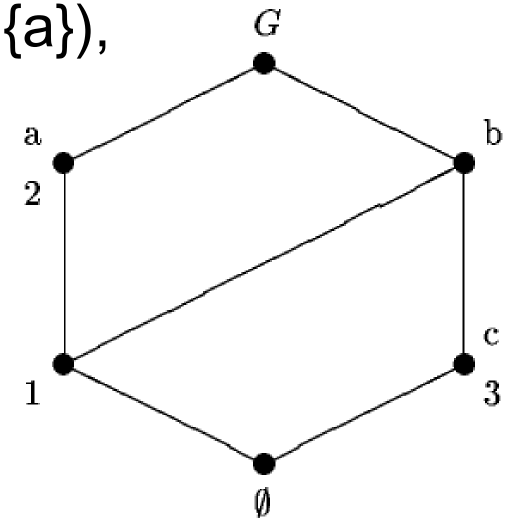
   $\Rightarrow$ 5. extent: $\{1,2\}$

7. $\{1,2\} \oplus 3 = \{1,2,3\}$ and $\{1,2\} \not\prec_3 \{1,2,3\}$

   $\Rightarrow$ 6. extent: $\{1,2,3\}$

8. As G = {1,2,3} there are no further extents.

- $\mathfrak{B}(G,M,I) = \{(\emptyset,M),(\{1\},\{a,b\}),(\{3\},\{b,c\}),(\{1,2\},\{a\}),(\{1,3\},\{b\}),(G,\emptyset)\}$



MANY-VALUED CONTEXT

Many-Valued Context

- In common language settings, objects are not described by simply having an attribute or not
- Attributes can have different values or notions
- Examples of such attributes are for example
  - Fruit has color: red, green, brown...
  - STI Int‘l member has location: Innsbruck, Karlsruhe, Berlin, Madrid...
  - Person has gender: male, female

- Such attributes are referred to as being multi-valued

Many-Valued Context (2)

STI · INNSBRUCK

- An advantage of working with many-valued contexts is the resulting modularity
- Concept lattices can grow exponentially in the size of the context
- The idea of using conceptual scaling for this problem is to only consider few attributes at a time
- Combinations of attributes that are of interest can be put together and can be analysed separately
- The combination of the various many-value contexts are evantually recombined to solve the initial problem

Many-Valued Context (3)

- A many-value context $\mathcal{M}$ is defined as $\mathcal{M}$ = (G, M, W, $\mathcal{I}$) with
  - G a set of objects
  - M a set of attributes
  - W a set of **values** (Wert)
  - $\mathcal{I}$ is a ternary relation ($\mathcal{I} \subseteq G \times M \times W$) between G, M, and W that satisfies $(g,m,v) \in \mathcal{I}, (g,m,w) \in \mathcal{I} \Rightarrow v = w$
- The meaning of $(g,m,w) \in \mathcal{I}$ is: „m has for g the value w“
- If n = |W|, the many-valued context $\mathcal{M}$ is called a n-ary context

Conceptual Scaling

- In order to apply the aforepresented formal concept analysis methods, a many-valued context must be transformed into a one-valued context, as it was treated so far
- This unfolding into a lattice is referred to as **conceptual scaling**
- The word 'scaling' is understood in the sense of 'embedding something in a certain (usually well-known) structure', called a scale: for example, embedding some objects according to the values of measurements of their temperature into a temperature scale
- Note that the outcome of the conceptual scaling procedure is not unique, but contains several degrees of freedom that allow different interpretations

Conceptual Scaling (2)

STI · INNSBRUCK

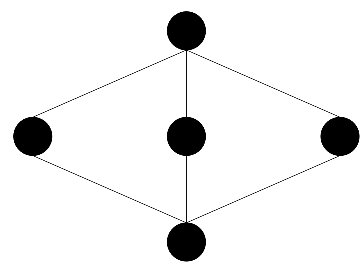
- The simplest version of conceptual scaling is called **plain scaling**
- In plain scaling, a scale is associated to each many-valued attribute m, and m is replaced by the set of its scale attributes.
- A **scale** for a many-valued context is defined as $S_m=(G_m,M_m,I_m)$ for each $m \in M$
- Note that S_m is a one-valued context

- The open questions that remain to be addressed:
  - How to choose the right S_m?
  - How to combine the different S_m in order to derive the desired one-valued context?

Conceptual Scaling (3)

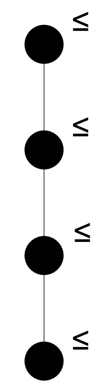
- Typically there is not a single best layout for a conceptual scale
- In principle conceptual scaling could be standardized, however, it mostly relies on the human interpretation
- Still, there are some „rules“ to keep the scales neutral
- Well known standard scales are:
  - Nominal scale

=	1	2	3	4	5
1	×				
2		×			
3			×		
4				×	
5					×



  - Ordinal scale

≤	1	2	3	4	5
1	×	×	×	×	×
2		×	×	×	×
3			×	×	×
4				×	×
5					×



Example

- The structure of data derived from a survey often represents rank orders (“agree strongly”, “agree”, “neutral”, “disagree”, “disagree strongly”)

=	agree strongly	agree	neutral	disagree	disagree strongly
agree strongly	X				
agree		X			
neutral			X		
disagree				X	
disagree strongly					X

- The lattice for such a rank order can be drawn without considering the actual results from the survey
- After drawing a conceptual scale (nominal scale here), formal objects can be mapped to their positions on the scale
- The same scale could thus be used for different surveys to compare their results

ILLUSTRATION BY A LARGER EXAMPLE

From Data to Concept Representation

- Data are often represented in a table form

K0	sex	age
ADAM	M	21
BETTY	F	50
CHRIS	/	66
DORA	F	88
EVA	F	17
FRED	M	/
GEORGE	M	90
HARRY	M	50

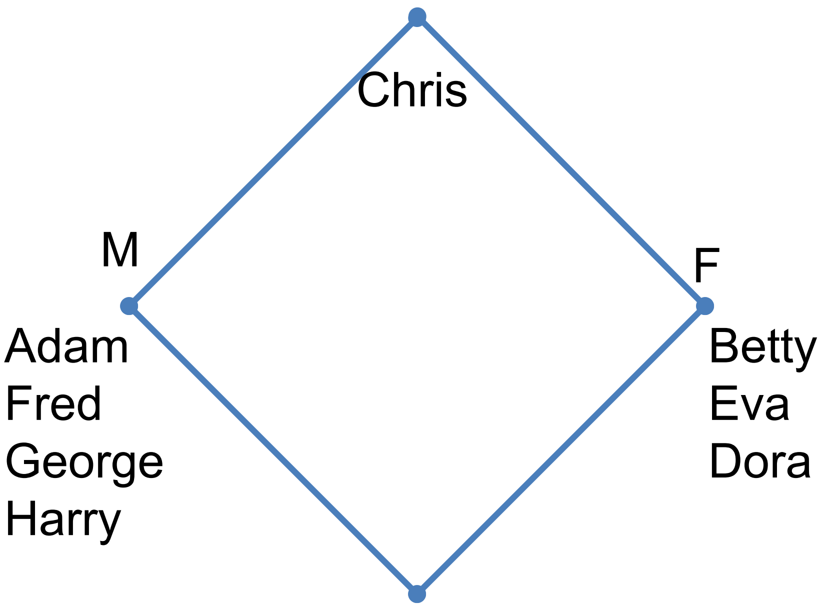
From Data to Concept Representation

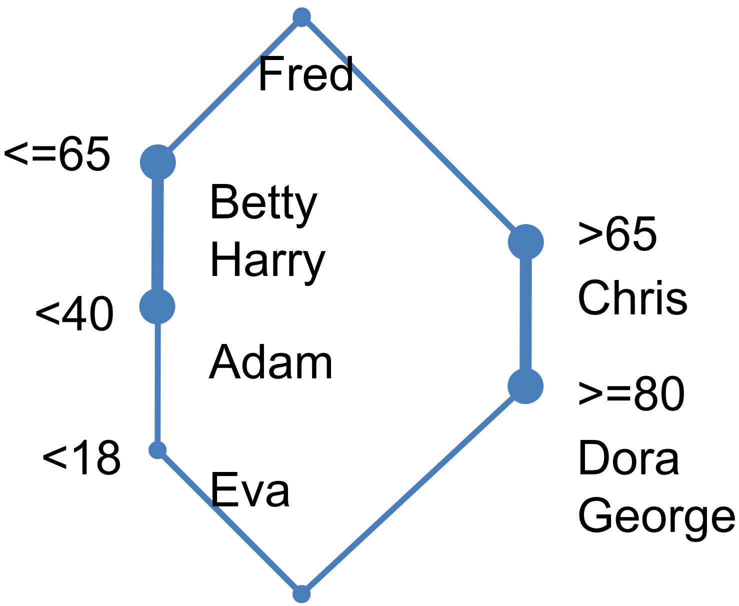
- The previous many-value context can be transformed to a formal context

K	sex		age				
	M	F	<18	<40	<=65	>65	>=80
ADAM	X			X	X		
BETTY		X			X		
CHRIS						X	
DORA		X				X	X
EVA		X	X	X	X		
FRED	X						
GEORGE	X					X	X
HARRY	X				X		

From Data to Concept Representation

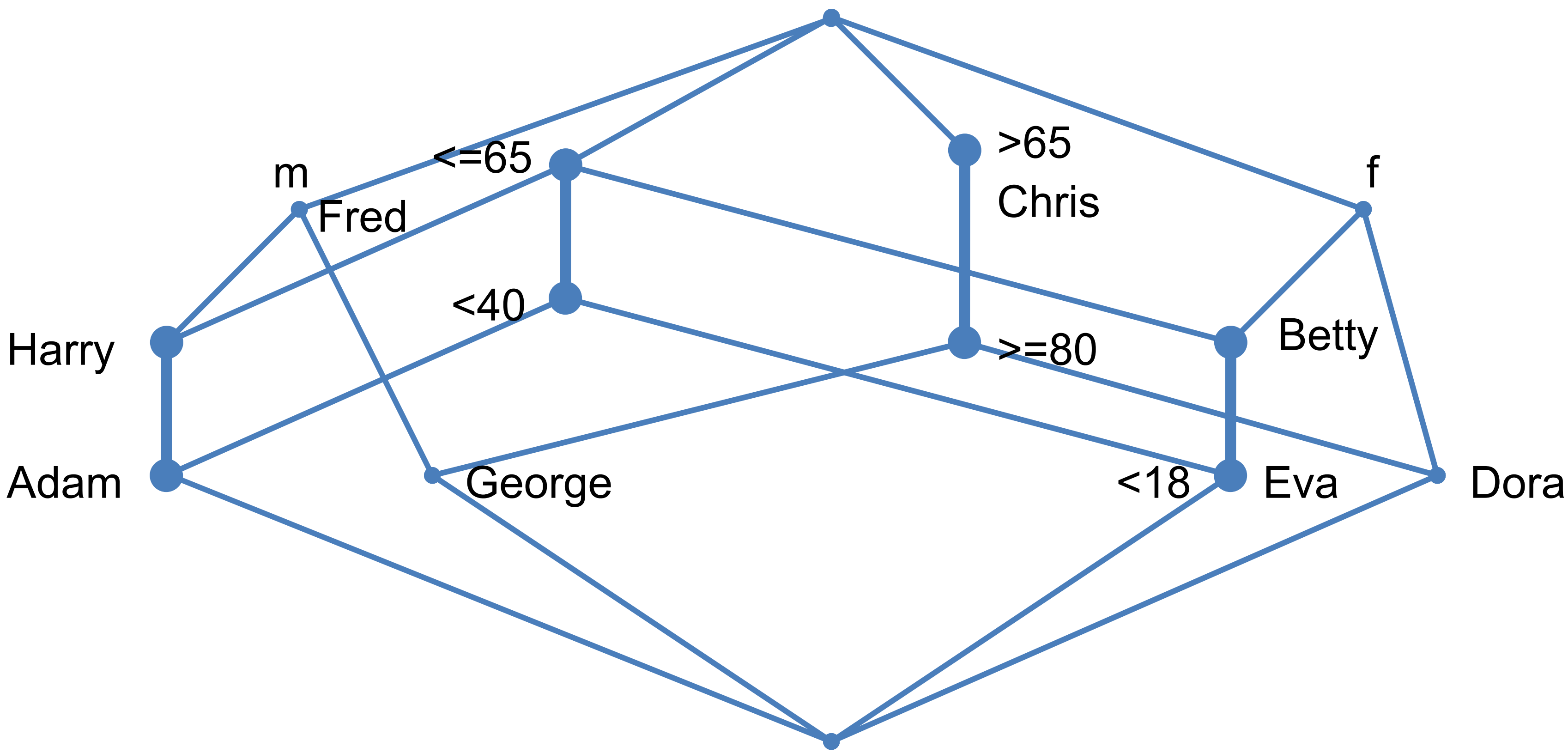
- By selecting a single view (i.e. an attribute space) we can create the following lattices





From Data to Concept Representation

- Diagrams can be combined to create a single lattice representing all the data space



EXTENSIONS

Extensions

STI · INNSBRUCK

- Formal Concept Analysis is a powerful instrument for knowledge representation , acquisition and inference, hence it can be applied as ontology techniques starting from a data set
  - E.g. create a taxonomy of accommodation facilities according to their attributes
- Formal Concept Analysis is the starting point for some data mining tasks such as Frequent Itemset Mining and Association Rules Mining
  - Frequent Itemset Mining defines, given a set of data and the lattice, the frequency of appearance of the nodes of the lattice. Frequency can be adopted to simplify and approximate the lattice (nodes with low frequency are removed)
  - Association Rules Minining defines, given Frequent Itemset, all the implications derived from them (based on the lattice) according to a given minimum support measure (the number of times for which an implication is valid over the data set)

SUMMARY

Summary

- Formal Concept Analysis is a method used for investigating and processing explicitely given information, in order to allow for meaningful and comprehensive interpretation
- A concept is given by a pair of objects and attributes within a formal context (G,M,I)
- The ordered set of all formal concepts in (G,M,I) is denoted by $\mathfrak{B}$(G,M,I) and is called concept lattice
- Within a concept lattices it is possible to derive concept hierarchies, to determine super-concept or sub-concepts
- Note again that there is not unique relationship between a context and a concept lattice, and thus we were looking at context reduction with the goal of minimizing the context without changing the lattice

Summary (2)

- Attribute exploration is a tool of formal concept analysis that supports the acquisition of knowledge
- For a specified context this interactive procedure determines a minimal list of valid implications between attributes of this context together with a list of objects which are counter-examples for all implications not valid in the context
- Finally, we talked about many-value contexts that take attributes into account that have multiple values rather than true or false

Summary (3)

- Conceptual scaling is a technique to transform many-value contexts into one-valued contexts
- Conceptual scaling is also a means to manage the potentially exponential growth of concept lattices
- Nested Line Diagrams are a graphical tool to represent multi-dimenstional many-value contexts

REFERENCES

References

- Bernhard Ganter, Gerd Stumme, Rudolf Wille (Hg.): *Formal Concept Analysis: Foundations and Applications*. Springer, 2005, ISBN 3-540-27891-5.
- Bernhard Ganter, Rudolf Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999, ISBN 3-540-62771-5.
- Bernhard Ganter, Rudolf Wille: Applied Lattice Theory: Formal Concept Analysis. In “General Lattice Theory”, Birkhauser, 1998, ISBN 0-817-65239-6.
- Uta Priss: Formal Concept Analysis in Information Science. Annual Review of Information Science and Technology 40, 2006, pp. 521-543.
- Rudolf Wille: *Introduction to Formal Concept Analysis*. TH Darmstadt (FB Mathematik), 1996.

Next Lecture

#	Date	Title
1		Introduction
2		Propositional Logic
3		Predicate Logic
4		Theorem Proving, Logic Programming, and Description Logics
5		Search Methods
6		CommonKADS
7		Problem Solving Methods
8		Planning
9		Agents
10		Rule Learning
11		Inductive Logic Programming
12		Formal Concept Analysis
13		**Neural Networks**
14		Semantic Web and Exam Preparation

Questions?