Explaining Data with Formal Concept Analysis

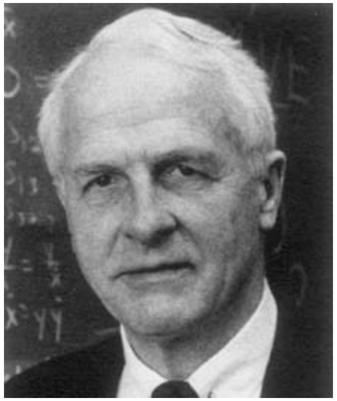
Sebastian Rudolph

Computational Logic ∴ Group

TECHNISCHE UNIVERSITÄT DRESDEN

Formal Concept Analysis?

- Branch of Applied Mathematics
- Based on **Lattice Theory** developed by Garrett Birkhoff and others in the 1930s



- Employs algebra in order to formalize notions of **concept** and **conceptual hierarchy**
- Term **Formal Concept Analysis** (short: FCA) introduced by Rudolf Wille in the 1980s.

Why Formal Concept Analysis?

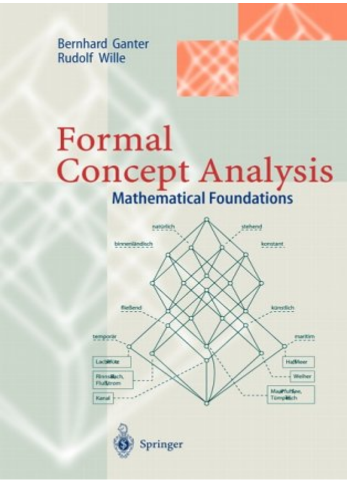
- The method of Formal Concept Analysis offers an algebraic approach to **data analysis** and **knowledge processing**.
- Strengths of FCA are
  - … a solid mathematical and philosophical foundation,
  - … more than 1000 research publications,
  - … experience of several hundred application projects,
  - … an expressive and intuitive graphical representation,
  - and a good algorithmic basis.
- Due to its elementary yet powerful formal theory, FCA can express other methods, and therefore has the potential to unify the methodology of **data analysis**.

FCA – Further Information

- Conferences
  - **International Conference on Formal Concept Analysis** (ICFCA)
  - **International Conference on Conceptual Structures** (ICCS)
  - **Concept Lattices and Applications** (CLA)
- Monograph
  - Bernhard Ganter & Rudolf Wille.

    „Formal Concept Analysis. Mathematical Foundations“

    Springer Verlag, 1999



- FCA Website by Uta Priss: http://www.upriss.org.uk/fca/fca.html

Acknowledgements

- Slides of this tutorial are partially based on material from …
  - Johanna Völker,
  - Peter Becker,
  - Bernhard Ganter,
  - Gerd Stumme, and
  - Bastian Wormuth.

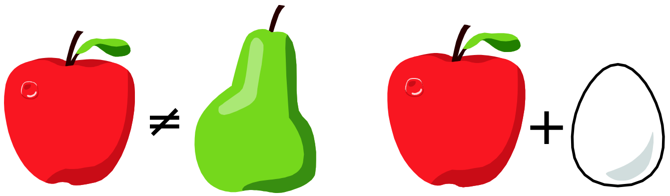
FOUNDATIONS OF FORMAL CONCEPT ANALYSIS

20.09.2019

Sebastian Rudolph

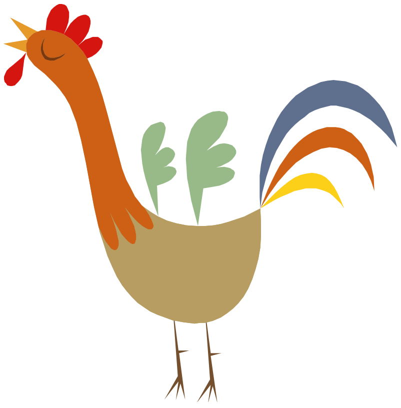
Introduction

- Formal Concept Analysis (FCA) is a…
  - “**mathematization**” of the philosophical understanding of concepts
  - human-centered method to **structure** and **analyze** data
  - method to **visualize** data and its inherent structures, implications and dependencies



What is a Concept?

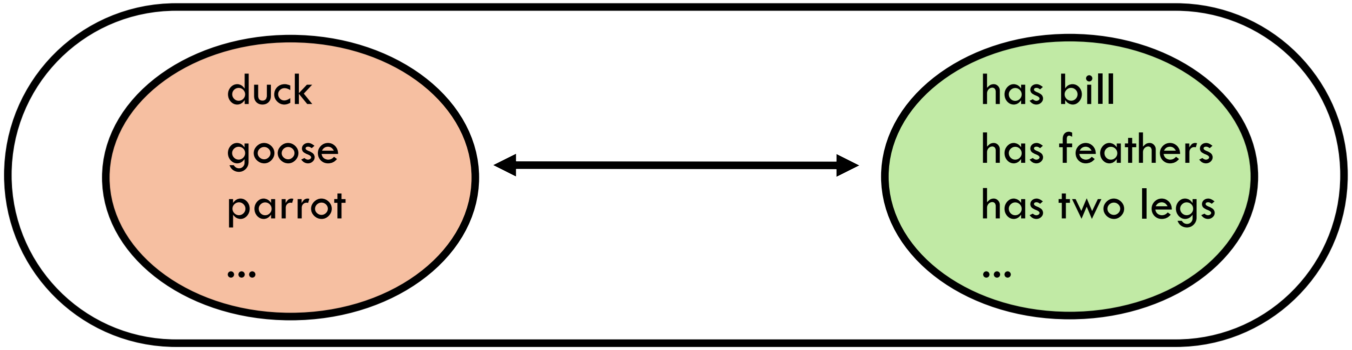
- Consider the concept “bird”. What drives us to call something a “bird” ?
- Every object with certain **attributes** is called “bird”:
  - A bird has feathers.
  - A bird has two legs.
  - A bird has a bill. …
- All **objects** having these attributes are called “birds”:
  - Duck, goose, owl and parrot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



What is a Concept?

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

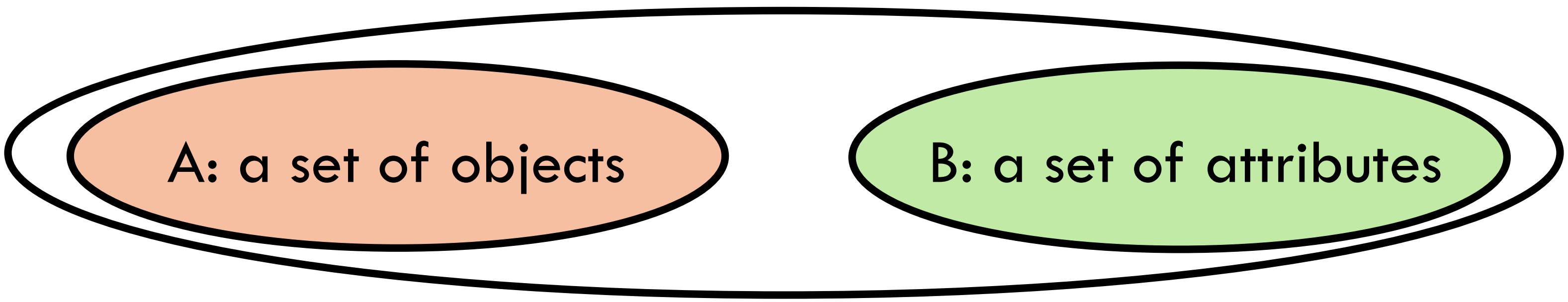
objects related to attributes.



→ Objects, attributes and a relation form a **formal concept**.

What is a Concept?

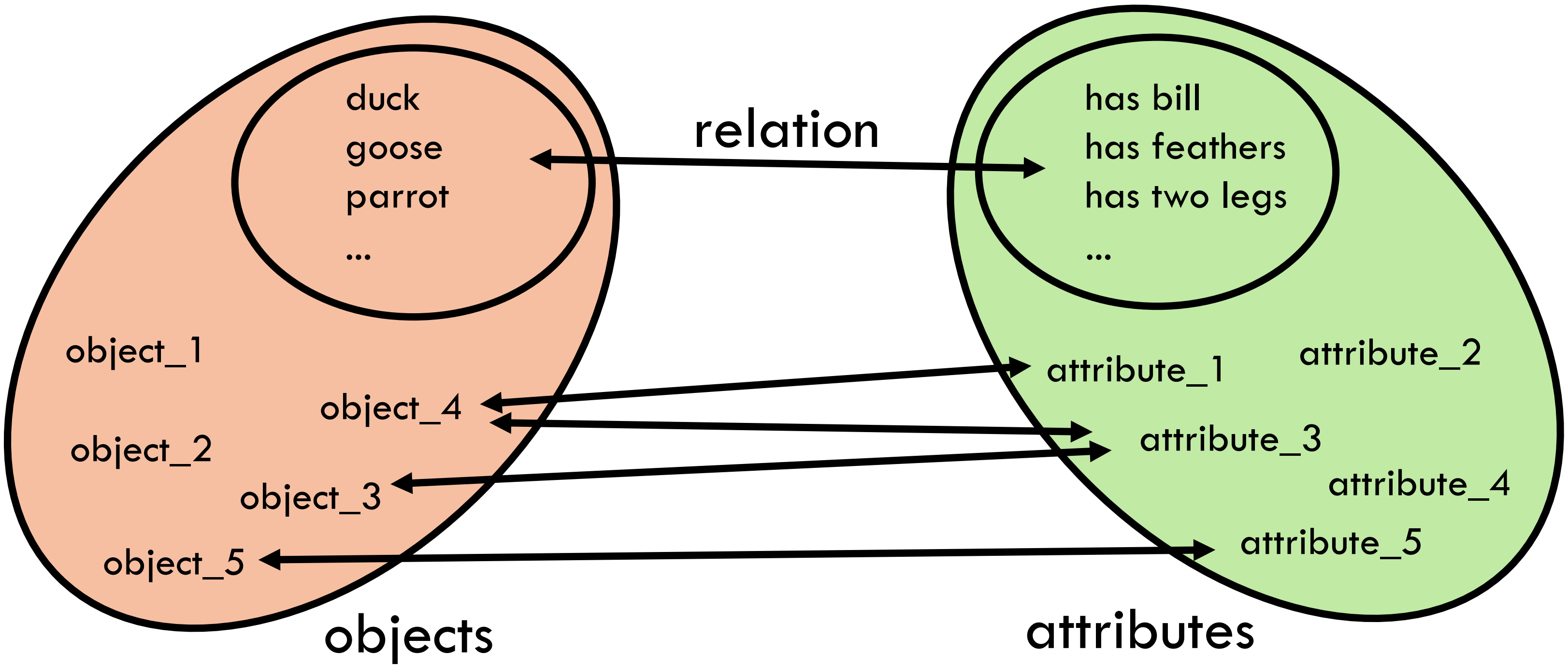
- So, a **formal concept** is constituted by two parts



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

The Universe of Discourse

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



The Formal Context

$\mathbb{K}$	small	medium	big	2legs	4legs	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							
owl	x			x		x							
hawk	x			x		x							
eagle		x		x		x							
fox		x			x								
dog		x			x								
wolf		x			x								
cat	x				x		x		x	x			
tiger			x		x		x		x	x			
lion			x		x								
horse			x		x								x
zebra					x								
cow					x		x						

set of **attributes** (M)

crosses indicate **incidence relation** (I) between G and M

$$I \subseteq G \times M$$

for $g \in G$ and $m \in M$, $(g,m) \in I$ means *object g has attribute m*

set of **objects** (G)

(G,M,I) is called **formal context**

Definition of Formal Concepts

- For the mathematical definition of **formal concepts** we introduce the **derivation operator** “'”.

For a set of objects A, A' is defined as:

A' = {all **attributes** in M common to the objects of A}
$= \{ m \in M \mid \forall g \in A : (g,m) \in I\}$

For a set of attributes B, B' is defined as:

B' = {**objects** in G having all attributes of B}
$= \{ g \in G \mid \forall m \in B : (g,m) \in I\}$

Applying the Derivation Operator

$\mathbb{K}$	small	medium	big	2legs	4legs	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

Applying the Derivation Operator

$\mathbb{K}$	small	medium	big	2legs	4legs	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

Applying the Derivation Operator

$\mathbb{K}$	small	medium	big	2legs	4legs	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

Properties of the Derivation Operator

- $X \subseteq Y \Rightarrow Y' \subseteq X'$

  The more objects we consider, the fewer attributes they have in common.

  The more attributes we require, the fewer objects we find having all of them.

- $X \subseteq X''$
- $X' = X'''$
- therefore, $''$ is a closure operator (on G or M):
  - monotone: $X \subseteq Y \Rightarrow X'' \subseteq Y''$
  - extensive: $X \subseteq X''$
  - idempotent: $(X'')'' = X''$

Definition of Formal Concepts

- We are looking for pairs (A,B) of objects A and attributes B that satisfy the conditions
  $$A' = B \text{ and } B' = A$$
  and we call these pairs **formal concepts**.
- Alternative, equivalent definition:
  - $A \times B \subseteq I$ (i.e., $(g,m) \in I$ for all $g \in A$ and $m \in B$) and
  - A and B are subset-maximal with that property.
- In words: maximal cross-filled rectangles in the context (possibly after swopping rows and columns).

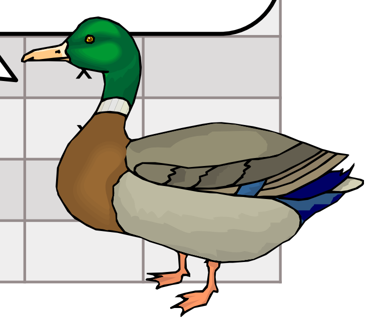
Calculating Formal Concepts

- Using the derivation operator we can derive **formal concepts** from our formal context with the following procedure:
  - Pick an object set A.
  - Derive the attributes A'.
  - Derive (A')'.
  - (A'',A') is a **formal concept**.
- The same routine could be applied starting with an attribute set B: (B',B'') is a formal concept as well.

Calculating Formal Concepts

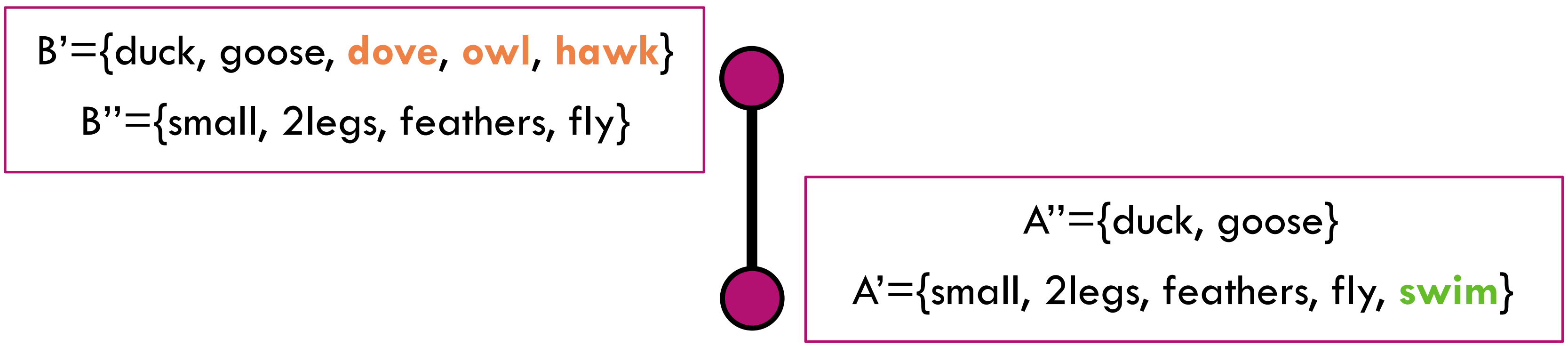
$\mathbb{K}$	small	medium	big	2legs	4legs	feathers	hair	fly	hunt	run	swim	mane	hooves
dove	x			x		x		x					
hen	x			x		x							
duck	x			x		x		x			x		
goose	x			x		x		x			x		
owl	x			x		x		x	x				
hawk	x			x		x		x	x				
lion			x		x		x		x	x		x	
horse			x		x		x			x		x	
zebra			x		x		x			x			
cow			x		x		x						

1. Pick a set of objects: A = {duck}
2. Derive attributes: A‘ = {small, 2legs, feathers, fly, swim}
3. Derive objects: (A‘)‘ = {small, 2legs, feathers, fly, swim}‘ = {duck, goose}
4. Formal concept: (A“,A‘) = ({duck, goose}, {small, 2legs, feathers, fly, swim})



Ordering Concepts

The **formal concept** (**A''**,**A'**)=({duck, goose}, {small, 2legs, feathers, fly, swim}) is represented in the line diagram as a node:



Consider another **formal concept** (**B'**,**B''**)=({duck, goose, dove, owl, hawk},{small, 2legs, feathers, fly}).

The **formal concept** (B',B'') is a superconcept of (A'',A') and (A'',A') is a subconcept of (B',B''), because A'' is a subset of B'.

So (B',B'') is drawn above (A'',A') and connected to it by a line.

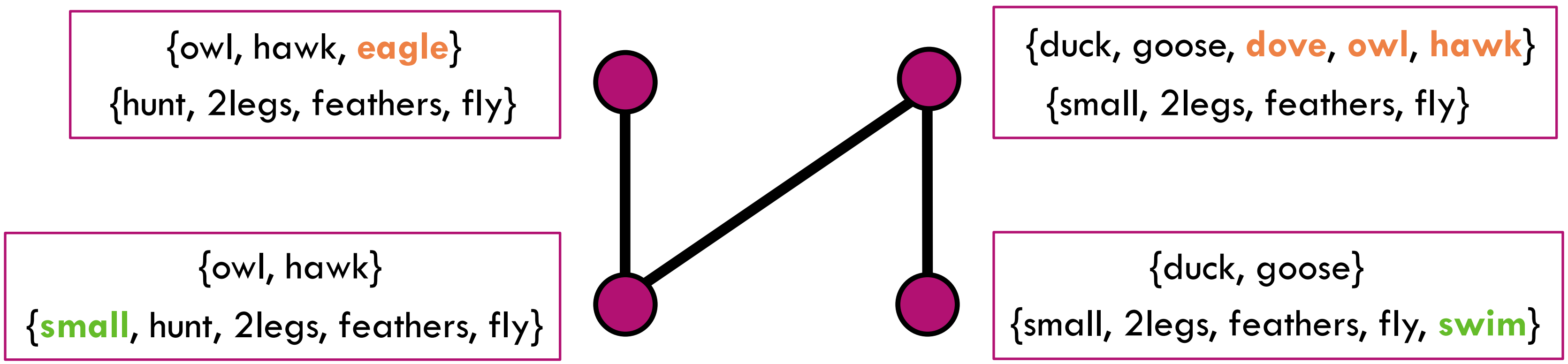
Ordering Concepts

We extend the diagram by adding more **formal concepts**

({owl, hawk}, {feathers, 2legs, small, fly, hunt})

({owl, hawk, eagle}, {feathers, 2legs, fly, hunt})

... and **subconcept relations**:



... and so on.

Several methods exist to derive all **formal concepts**:
Cut over extents, Ganter's algorithm etc.

The Concept Lattice

- The subconcept–superconcept relation defines an order $\leqslant$ on the set $\mathfrak{B}$ of all **formal concepts** of a formal context
- For two concepts (A_1, A_2) and (B_1, B_2), this order is defined by:

$$(A_1,A_2) \leqslant (B_1,B_2) \Leftrightarrow A_1 \subseteq B_1 \; (\Leftrightarrow B_2 \subseteq A_2)$$

- (A_1,A_2) is smaller than (B_1,B_2) if A_1 is subset of B_1 (**objects**) and B_2 is subset of A_2 (**attributes**). Hence, $(\mathfrak{B},\leqslant)$ is an ordered set.
- The set $\mathfrak{B}$ of **formal concepts** has another property:
  - For each family of formal concepts of a **formal context** there exists always a unique greatest subconcept and a unique smallest superconcept.
- The ordered set $\underline{\mathfrak{B}}=(\mathfrak{B},\leqslant)$ plus the last property forms a mathematical structure: the **concept lattice.**

Concept Lattice – Formal Concepts

{duck, goose, dove, owl, hawk, eagle}

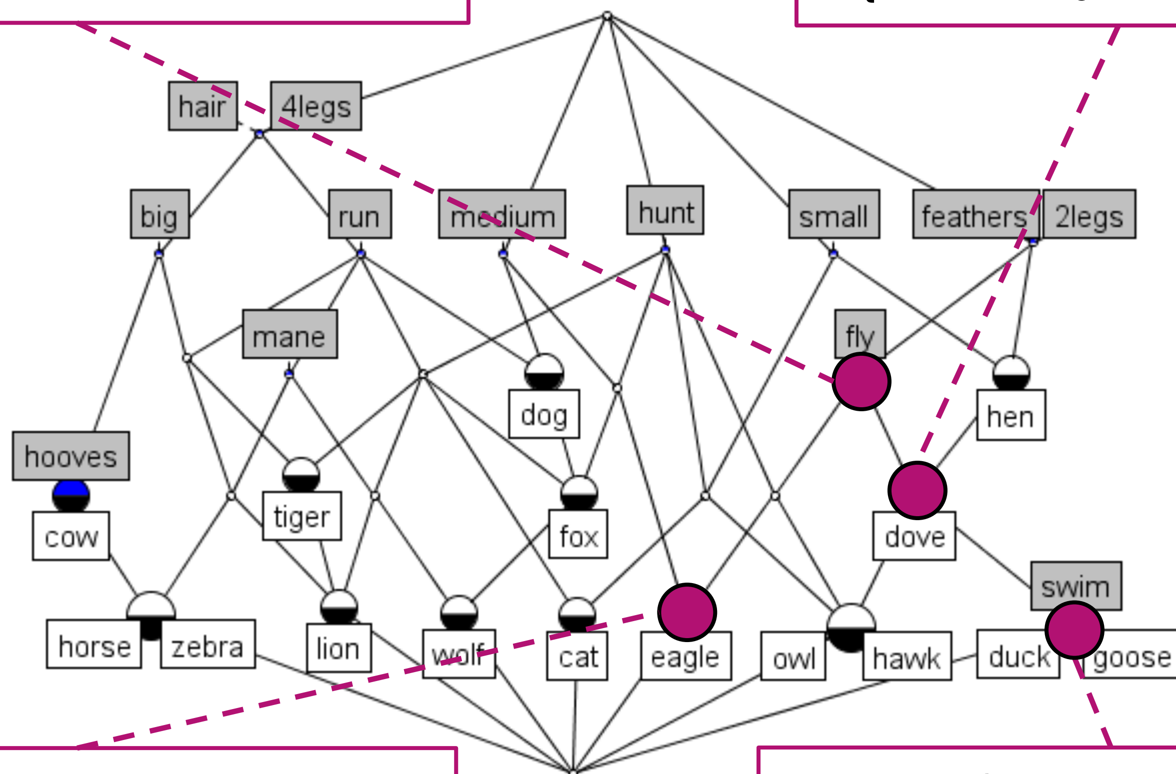
{2legs, feathers, fly}

{duck, goose, dove, owl, hawk}

{small, 2legs, feathers, fly}

„flying birds“

„small flying birds“



„medium hunting birds“

„small swimming birds“

{eagle}

{medium, hunt, 2legs, feathers, fly}

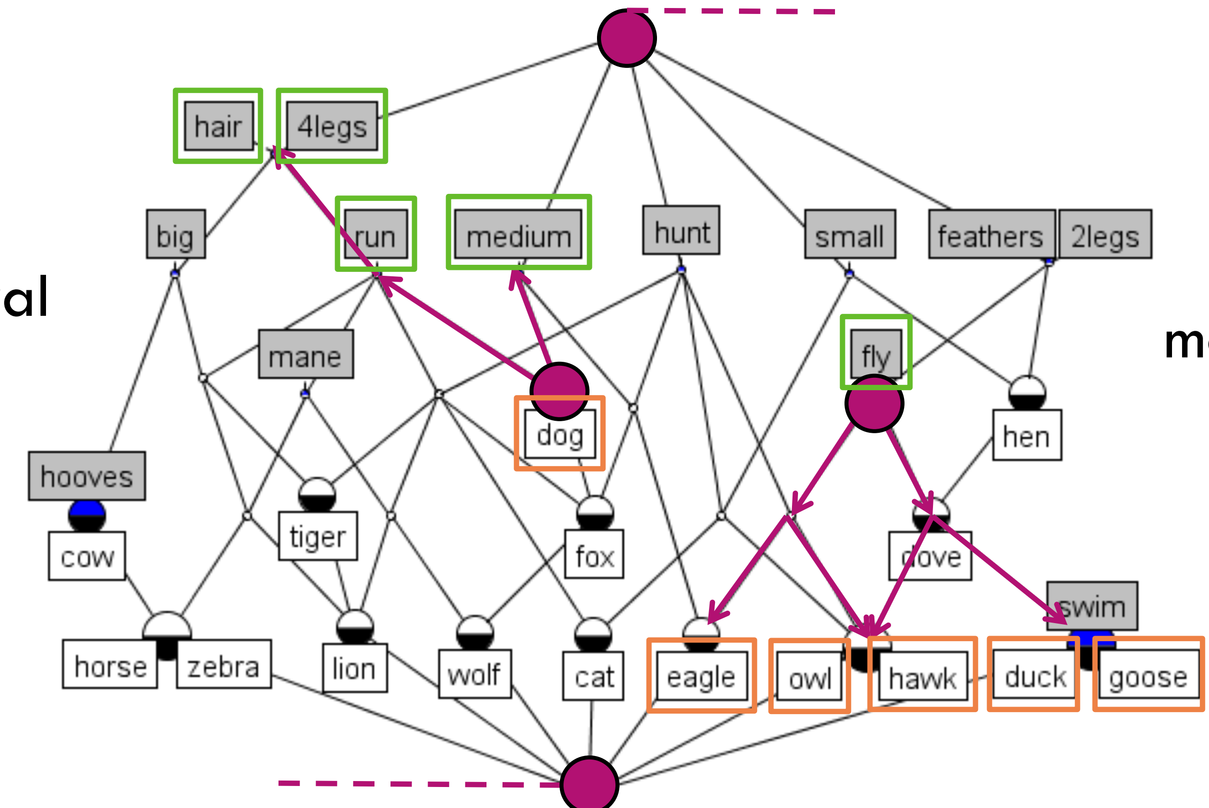
{duck, goose}

{small, swim, 2legs, feathers, fly}

Concept Lattice – Top and Bottom

no **attributes**
all **objects**

more general
concepts



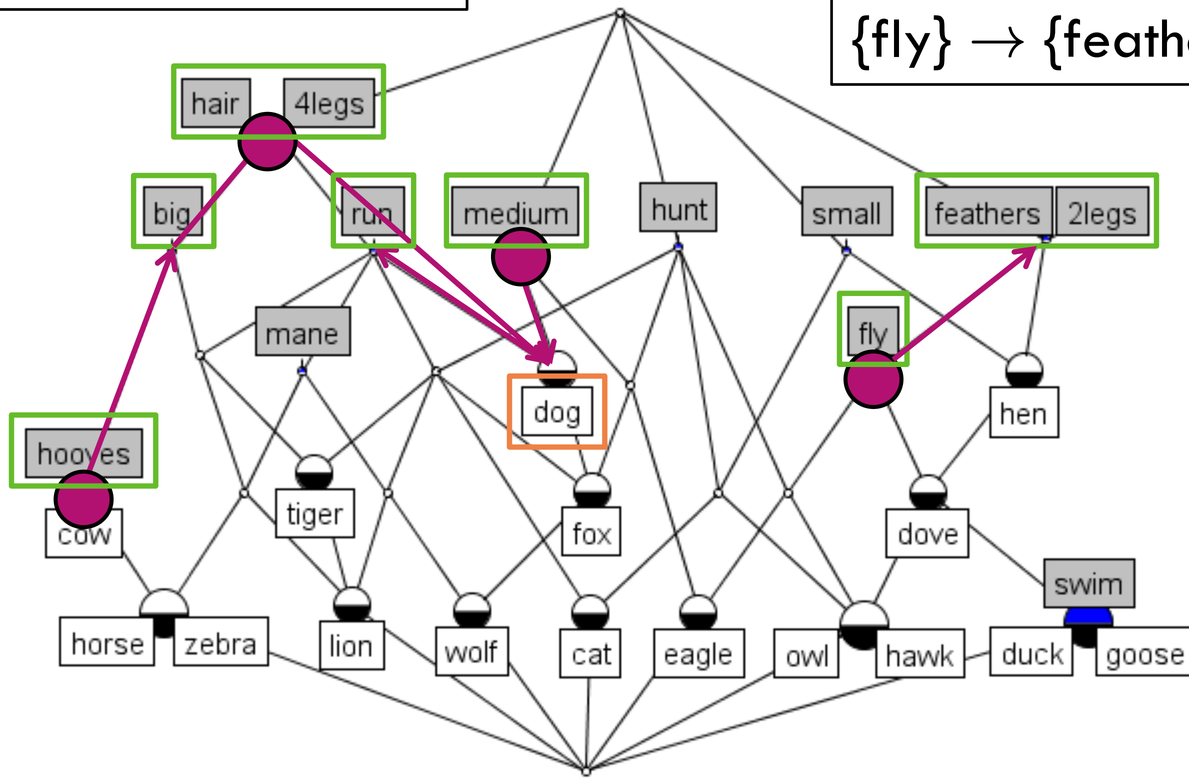
more specific
concepts

all **attributes**
no **objects**

Concept Lattice – Implications

$\{hooves\} \rightarrow \{big, 4legs, hair\}$

$\{fly\} \rightarrow \{feathers, 2legs\}$



How to determine <u>all</u> the **implications** for a given lattice or context?

Computing the Concept Lattice

- How to compute all formal concepts?
- Variant 1: brute-force enumeration

Theorem

Each concept of a context (G, M, I) has the form (X'', X') for some subset $X \subseteq G$ and (Y', Y'') for some subset $Y \subseteq M$. Conversely, all such pairs are concepts.

Algorithm

Determine for every subset Y of M the pair (Y', Y'').

Inefficient! (Too) many concepts are generated multiple times.

Computing the Concept Lattice

- How to compute all formal concepts?
- Variant 2: intersection method

- Suitable for manual computation (Wille 1982)
- Best worst-case time complexity (Nourine, Raynoud 1999)
- Based on the following

Theorem

Every extent is the intersection of attribute extents. (I.e., the closure system of all extents is generated by the attribute extents.)

Which intersections of attribute extents should we take?

Computing the Concept Lattice

- How to compute all formal concepts?
- Variant 2: intersection method

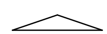
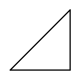
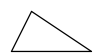
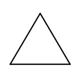
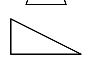
How to determine all formal concepts of a formal context:

1. For each attribute $m \in M$ compute the attribute extent $\{m\}'$.
2. For any two sets in this list, compute their intersection. If it is not yet contained in the list, add it.
3. Repeat until no new extents are generated.
4. If G is not yet contained in the list, add it.
5. For every extent A in the list compute the corresponding intent A'.

Computing the Concept Lattice

- How to compute all formal concepts?
- Variant 2: intersection method

On the blackboard: "triangle" example

Triangles				
abbreviation	coordinates			diagram
T1	(0,0)	(6,0)	(3,1)	
T2	(0,0)	(1,0)	(1,1)	
T3	(0,0)	(4,0)	(1,2)	
T4	(0,0)	(2,0)	$(1,\sqrt{3})$	
T5	(0,0)	(2,0)	(5,1)	
T6	(0,0)	(2,0)	(1,3)	
T7	(0,0)	(2,0)	(0,1)	

Attributes	
symbol	property
a	equilateral
b	isoceles
c	acute angled
d	obtuse angled
e	right angled

	a	b	c	d	e
$T1$		$\times$		$\times$	
$T2$		$\times$			$\times$
$T3$			$\times$		
$T4$	$\times$	$\times$	$\times$		
$T5$				$\times$	
$T6$		$\times$	$\times$		
$T7$					$\times$

Drawing the Concept Lattice

- Given a list of concepts, how to manually draw the lattice diagram?

How to draw a concept lattice by hand:

1. Draw a small circle for the extent G at the top.
2. For every extent (starting with the one's with the most elements) draw a small circle and connect it with the lowest circle(s) whose extent contains the current extent.
3. Every attribute is written slightly above the circle of its attribute extent.
4. Every object is written slightly below the circle that is exactly below the circles that are labeled with the attributes of the object.

Drawing the Concept Lattice

- Given a formal context and a drawn concept lattice diagram, how to check the latter is correct?

Theorem

The concept lattice $\mathfrak{B}(G,M,I)$ is a complete lattice in which infimum and supremum are given by

$$\bigwedge_{t\in T}(A_t,B_t)=\left(\bigcap_{t\in T}A_t,\left(\bigcup_{t\in T}B_t\right)''\right) \text{ and } \bigvee_{t\in T}(A_t,B_t)=\left(\left(\bigcup_{t\in T}A_t\right)'',\bigcap_{t\in T}B_t\right)$$

A complete lattice $(V,\leqslant)$ is isomorphic to $\underline{\mathfrak{B}}(G,M,I)$ if and only if there are mappings $\tilde{\gamma}:G\to V$ and $\tilde{\mu}:M\to V$ such that

- $\tilde{\gamma}(G)$ *is supremum-dense in* $(V,\leqslant)$,
- $\tilde{\mu}(M)$ *is infimum-dense in* $(V,\leqslant)$, *and*
- gIm *is equivalent to* $\tilde{\gamma}(g)\leqslant\tilde{\mu}(m)$ *for all* $g\in G$ *and all* $m\in M$.

In particular, $(V,\leqslant)\cong\underline{\mathfrak{B}}(V,V,\leqslant)$.

Drawing the Concept Lattice

- Given a formal context and a drawn concept lattice diagram, how to check the latter is correct?

How you can check the drawn diagram:

1. Is it really a lattice? (that's often skipped)
2. Is every concept with exactly one upper neighbor labeled with at least one attribute?
3. Is every concept with exactly one lower neighbor labeled with at least one object?
4. Is for every $g \in G$ and $m \in M$ the label of the object g below the label of the attribute m iff $(g, m) \in I$ holds?

Summary

- Formal contexts
  - **Objects**, **attributes** and **incidence relation**
- … and **formal concepts**
  - Extent and intent
  - Subconcept relations
- Concept lattices
  - How to interpret a concept lattice
  - Generalization and specialization
  - Implications
- How to draw and verify concept lattices
- Next: **implications and attribute exploration**

IMPLICATIONS AND ATTRIBUTE EXPLORATION

20.09.2019

Sebastian Rudolph

Attribute Implications (aka propositional Horn clauses)

- For A,B $\subseteq$ M, the *implication* A $\rightarrow$ B *holds* in $\mathbb{K}$, if every object having all attributes from A also has all attributes from B.
- Formally: A $\subseteq$ {g}' implies B $\subseteq$ {g}' for all g $\in$ G
- Examples:
  - {wet} $\rightarrow$ {fluid}
  - {fluid, dry} $\rightarrow$ {warm}
  - {dry, wet} $\rightarrow$ {cold} (!)

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

How to „Datamine“ Implications?

- We want to extract the „implicational“ knowledge from a formal context.
- Very naive approach: enumerate all $(2^{2^{|M|}})$ implications and check against context.
  - Takes way too long.
  - Generated implication set is extremely redundant.
- Examples:
  - $\{\text{fluid, dry}\} \to \{\text{fluid}\}$
  - $\{\text{wet}\} \to \{\text{fluid}\}$ vs. $\{\text{wet, cold}\} \to \{\text{fluid}\}$

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

How to „Datamine“ Implications?

- Observations:
  - For any attribute set A, the implication $A \to A''$ holds in $\mathbb{K}$
  - If $A \to B$ holds in $\mathbb{K}$ then $B \subseteq A''$
- Hence the implications of the form $A \to A''$ provide enough information about all implications of the context.
- Still rather naive approach: enumerate all ($2^{|M|}$) attribute sets A and generate implication $A \to A''$
  - Still takes way too long
  - Generated implication set is still extremely redundant

What does Redundancy Mean?

- Boils down to question of logical entailment of implications: When does an implication $A \rightarrow B$ follow from a set $\mathfrak{I}$ of implications?
- Two equivalent definitions:
  - Semantically: $A \rightarrow B$ holds in every formal context wherein every implication from $\mathfrak{I}$ holds.
  - Syntactically: $A \rightarrow B$ can be derived from $\mathfrak{I}$ using the three *Armstrong Rules*:

$$\frac{}{X \rightarrow X} \qquad \frac{X \rightarrow Y}{X \cup Z \rightarrow Y} \qquad \frac{X \rightarrow Y \qquad Y \cup Z \rightarrow W}{X \cup Z \rightarrow W}$$

Implication Bases

- Given a formal context $\mathbb{K}$, a set of implications $\mathfrak{I}$ is called *implication base* of $\mathbb{K}$, if …
  - every implication $A \rightarrow B$ from $\mathfrak{I}$ holds in $\mathbb{K}$,
  - every implication $A \rightarrow B$ holding in $\mathbb{K}$ can be derived from $\mathfrak{I}$, and
  - none of the implications from $\mathfrak{I}$ can be derived from the other implications contained in $\mathfrak{I}$
- Question: which $A \rightarrow A''$ to choose to make up an implication base?

The Stem Base

- Question: which $A \rightarrow A''$ to choose to make up an implication base?
- Answer: take all the *pseudo-intents* of $\mathbb{K}$.
- Attribute set P is called pseudo-intent, if
  - P is not an intent (i.e. $P \neq P''$), but
  - if P contains another pseudo-intent Q, then it also contains Q''
- Definition recursive (but OK at least for finite M)
- Set $\{P \rightarrow P'' \mid P \text{ pseudo-intent}\}$ is called *stem base*

How to Compute the Stem Base

- We order attributes in a row:
  - e.g. a,b,c,d,e,f
- Based on that order, attribute sets are encoded as bit-vectors of length $|M|$
  - e.g. $\{a,c,d\}$ becomes $[1,0,1,1,0,0]$
- Implications are pairs of bit-vectors
  - e.g. $\{a\} \to \{a,e,f\}$ becomes $([1,0,0,0,0,0], [1,0,0,0,1,1])$
- Implications can be „applied" to attribute sets
  - $(\{a\} \to \{a,e,f\})$ applied to $\{a,c,d\}$ yields $\{a,c,d,e,f\}$
    $([1,0,0,0,0,0], [1,0,0,0,1,1])\ [1,0,1,1,0,0] = [1,0,1,1,1,1]$
- Implication sets can be applied to attribute sets:
  - $\{\{b,d\}\to\{c\},\{a\}\to\{d\}\}$ applied to $\{a,b\}$ yields $\{a,b,c,d\}$
  - write $\Im(A)$ for the result of applying implication set $\Im$ to attribute set A
- A+i defined as: take A, set ith bit to 1 and all subsequent bits to 0
  - e.g. $[0,1,0,0,1,1]+3 = [0,1,1,0,0,0]$

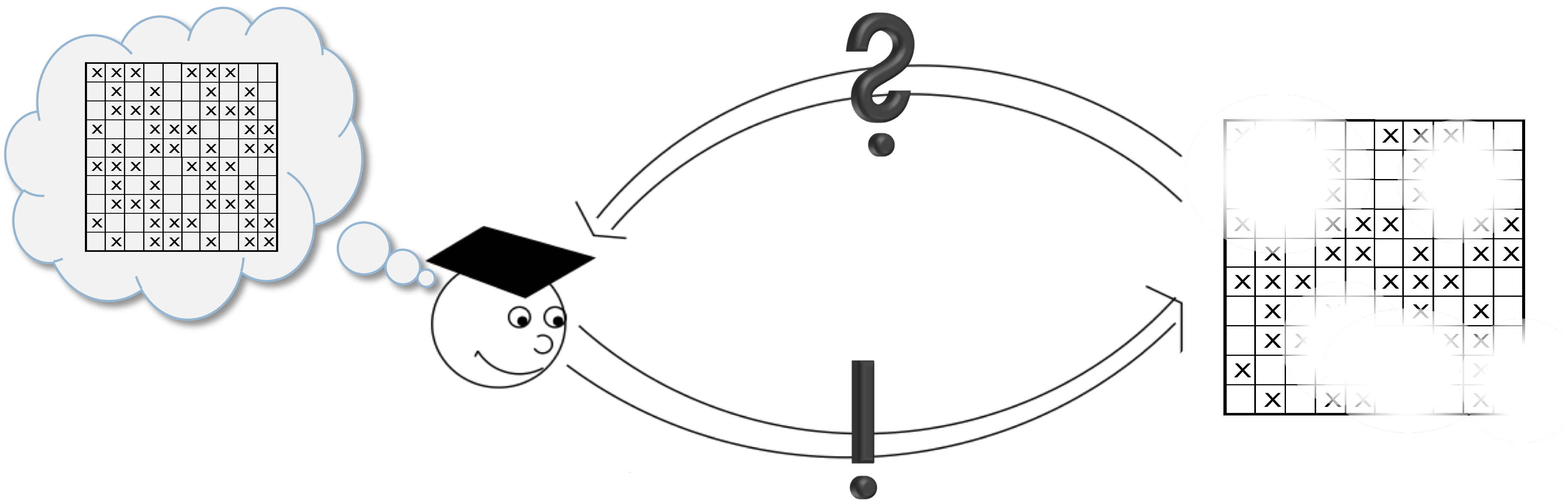
How to Compute the Stem Base

- Input formal context $\mathbb{K}$
- Create list $\mathfrak{I}$ of implications, initially empty
  Let A = [0,0,…,0]   *(bit representation of empty set)*
- Repeat
  - Add A $\rightarrow$ A'' to $\mathfrak{I}$ in case A $\neq$ A''
  - Starting from i = |M|+1, decrement i until
    - i=0 or
    - The ith bit of A is 0 and
      applying $\mathfrak{I}$ to A+i produces 1s only at positions greater than i
  - If i=0 output $\mathfrak{I}$ and exit
  - Let A = $\mathfrak{I}$(A+i)

… i …
A: [0,0,1,0,1,1,0]
A+i: [0,0,1,1,0,0,0]
$\mathfrak{I}$(A+i): [0,0,1,1,0,1,1]

Interactive Knowledge Acquisition via Attribute Exploration

- Sometimes, $\mathbb{K}$ is not entirely known from the beginning, but implicitly present as an expert‘s knowledge
- Attribute exploration determines the stembase of $\mathbb{K}$ by asking expert for missing information



Interactive Knowledge Acquisition via Attribute Exploration

- Sometimes, $\mathbb{K}$ is not entirely known from the beginning, but implicitly present as an expert's knowledge
- Attribute exploration determines the stembase of $\mathbb{K}$ by asking expert for missing information
  - M known and fixed
  - H $\subseteq$ G objects that are known in advance (as well as their attributes)
- Idea: use stembase algorithm on incomplete context which is updated on the fly

Stem Base Algorithm Revisited

- Input formal context $\mathbb{K}=(H,M,J)$ where $J=(H\times M)\cap I$
- Create list $\mathfrak{I}$ of implications, initially empty
  Let $A = [0,0,\ldots,0]$ *(bit representation of empty set)*
- Repeat
  - Add $A \rightarrow A''$ to $\mathfrak{I}$ in case $A \neq A''$
  - Starting from $i = |M|+1$, decrement i ...til
    - i=0 or
    - The ith bit of A is 0 and applying $\mathfrak{I}$ to A+i produces 1s only at
  - If i=0 output $\mathfrak{I}$ and exit
  - Let $A = \mathfrak{I}(A+i)$

Has to be altered, because implication valid in $\mathbb{K}$ might be invalid in $\mathbb{K}$ since refuted by an object not yet recorded. Then augmenting $\mathbb{K}$ by this object allows to refine the hypothesis.

Making It Interactive...

- Instead of just adding $A \to A''$ to $\mathfrak{I}$, do the following control loop:
  - While $A \neq A''$
    - Ask expert whether $A \to A''$ is valid in $\mathbb{K}$
    - If yes, add $A \to A''$ to $\mathfrak{I}$ and exit while-loop, otherwise ask for counterexample and add it to $\underline{\mathbb{K}}$
- What is a counterexample for $A \to A''$?
  - An object having all attributes from A but missing some from A''
- How to add a counterexample g to $\underline{\mathbb{K}}$=(H,M,J)?
  - $H_{new} = H \cup \{g\}$
  - $J_{new} = J \cup \{(g,m) \mid m \text{ is attribute of } g \text{ in } \mathbb{K}\}$
  - Essentially: just add a line to the cross table

Making It Interactive...

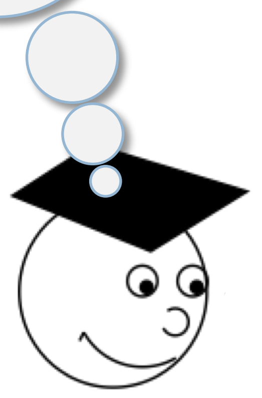
- Instead of just adding $A \rightarrow A''$ to $\Im$, do the following control loop:
  - While $A \neq A'$,
    - Ask expert whether $A \rightarrow A''$ is valid in $\mathbb{K}$
    - If yes, add $A \rightarrow A''$ to $\Im$ and exit while-loop,
      otherwise ask for counterexample g and add it to $\underline{\mathbb{K}}$
- Remarks:
  - Attribute set of g has to comply with the implications already confirmed
  - Changing $\underline{\mathbb{K}}$ changes the operator $(.)''$
  - It is not obvious (but has to be proven) that this indeed works, i.e. the enumeration done beforehand is not corrupted by updating the context

Stem Base Algorithm Revisited

- Input formal context $\mathbb{K}$
- Create list $\Im$ of implications, initially empty
  Let A = [0,0,…,0]   *(bit representation of empty set)*
- Repeat
  - While A ≠ A',
    - Ask expert whether $A \rightarrow A''$ is valid in $\mathbb{K}$
    - If yes, add $A \rightarrow A''$ to $\Im$ and exit while-loop,
      otherwise ask for counterexample g and add it to $\mathbb{K}$
  - Starting from i = |M|+1, decrement i until
    - i=0 or
    - The ith bit of A is 0 and
      applying $\Im$ to A+i produces 1s only at positions greater than i
  - If i=0 output $\Im$ and exit
  - Let A = $\Im$(A+i)

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [0, 0, 0, 0, 0]

A'': [0, 0, 0, 0, 1]

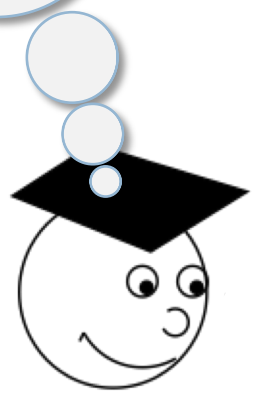
$\{\} \rightarrow \{\text{cold}\}$?

(are all elements cold?)

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [0, 0, 0, 0, 0]

A'': [0, 0, 0, 0, 1]

$\{\} \rightarrow \{\text{cold}\}$?

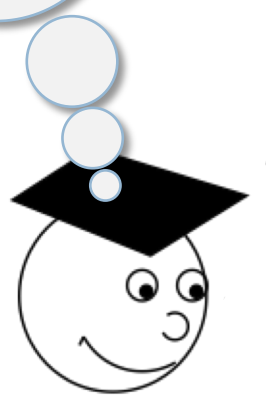
(are all elements cold?)

no: air is not cold!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

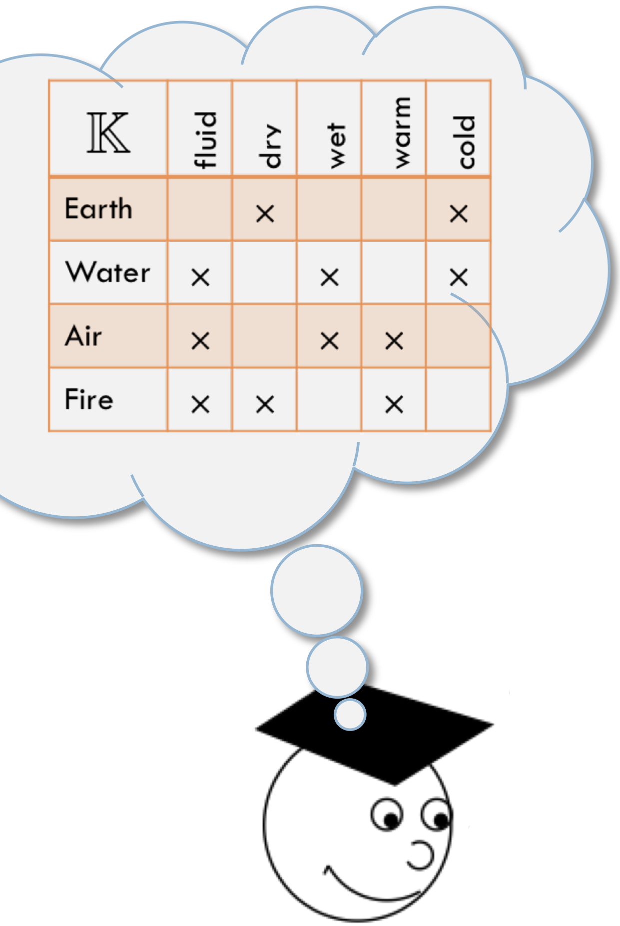


A: [0, 0, 0, 0, 0]

A'': [0, 0, 0, 0, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	

A Tiny Example: the Four Elements



$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

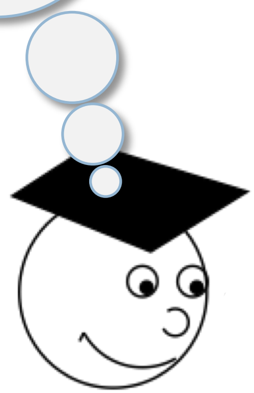
A: [0, 0, 0, 0, 1]

A'': [0, 0, 0, 0, 1]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [0, 0, 0, 1, 0]

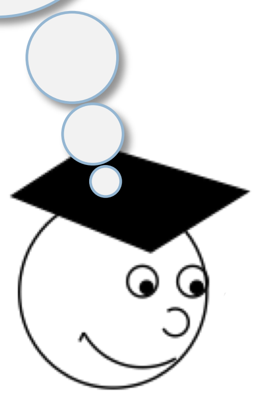
A'': [1, 0, 1, 1, 0]

$\{\text{warm}\} \rightarrow \{\text{wet}, \text{fluid}\}$?

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



$$A: [0, 0, 0, 1, 0]$$

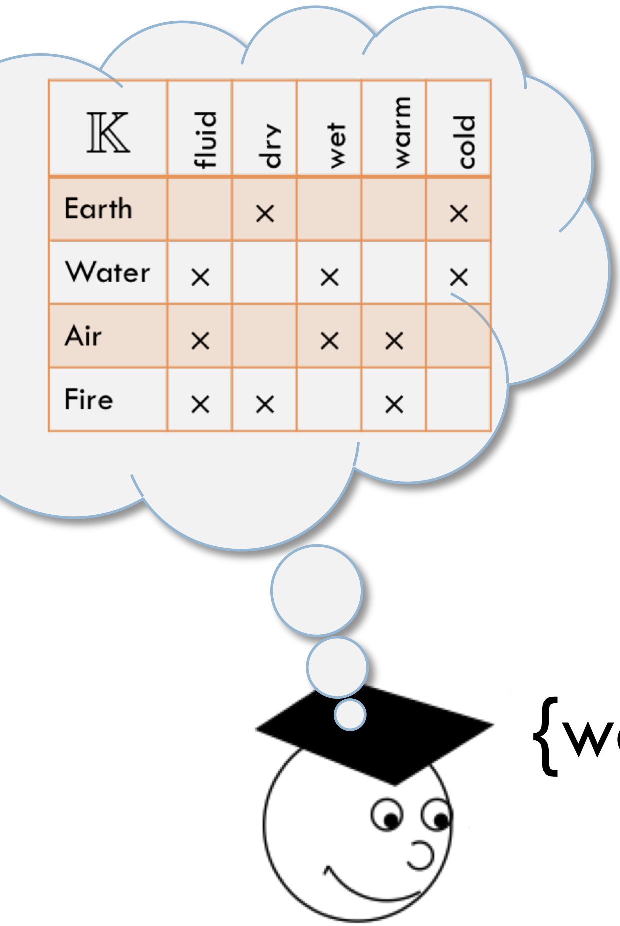
$$A'': [1, 0, 1, 1, 0]$$

$$\{warm\} \rightarrow \{wet, fluid\}?$$

no: fire is warm but not wet!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements



$\{\text{warm}\} \rightarrow \{\text{fluid}\}$?

A: [0, 0, 0, 1, 0]

A'': [1, 0, 0, 1, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

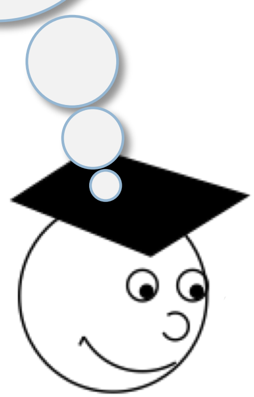
A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

$[0,0,0,1,0] \rightarrow [1,0,0,1,0]$

A: [0, 0, 0, 1, 0]

A'': [1, 0, 0, 1, 0]

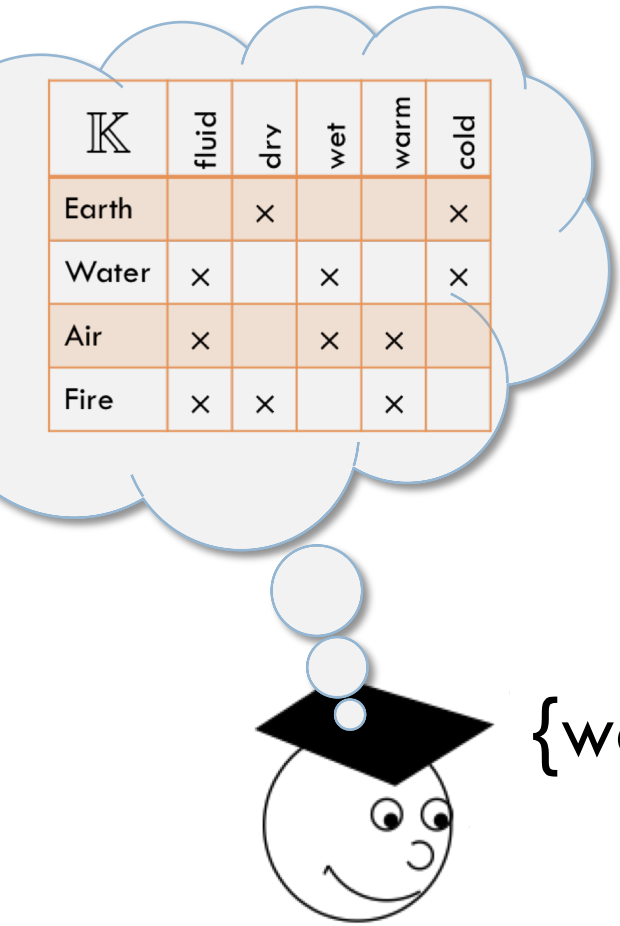


$\{\text{warm}\} \rightarrow \{\text{fluid}\}$?

yes!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

A: [0, 0, 1, 0, 0]

A'': [1, 0, 1, 0, 0]

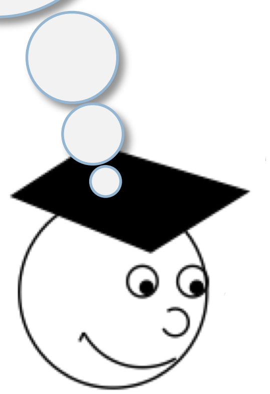
{wet} → {fluid}?

yes!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

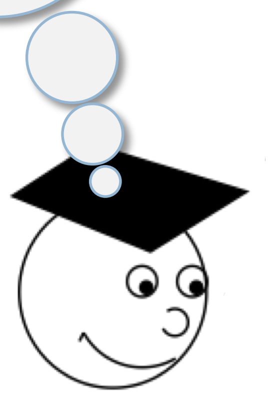
A: [0, 1, 0, 0, 0]

A'': [0, 1, 0, 0, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

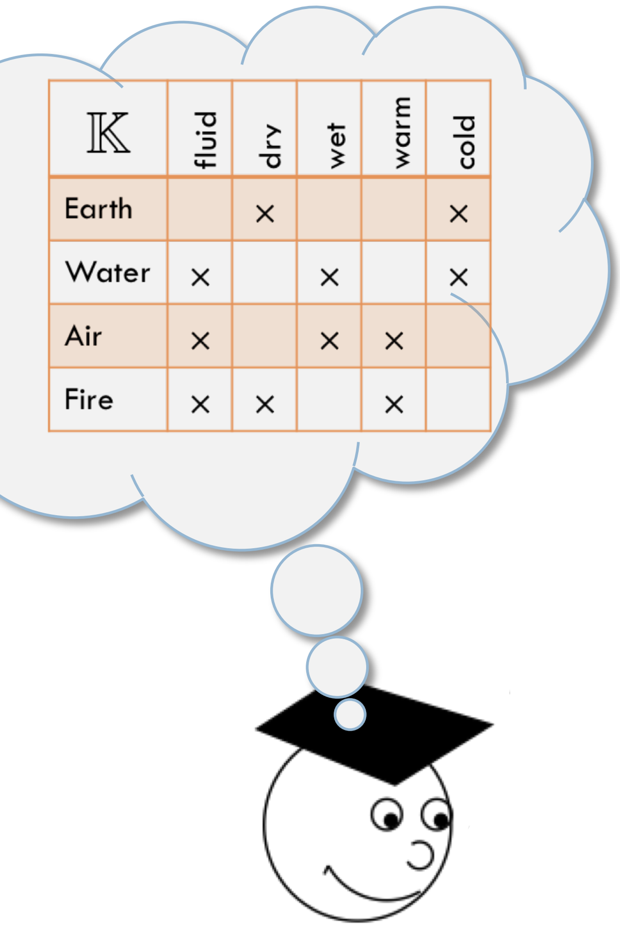
[0,0,1,0,0]→[1,0,1,0,0]

A: [0, 1, 0, 0, 1]

A'': [0, 1, 0, 0, 1]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements



$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

$[0,0,0,1,0] \rightarrow [1,0,0,1,0]$

$[0,0,1,0,0] \rightarrow [1,0,1,0,0]$

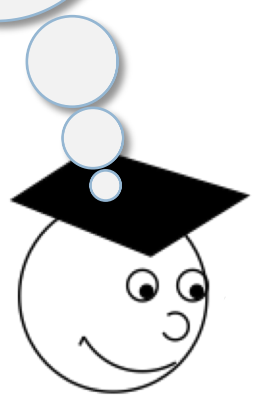
A: [1, 0, 0, 0, 0]

A'': [1, 0, 0, 0, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

A: [1, 0, 0, 0, 1]

A'': [1, 0, 1, 0, 1]

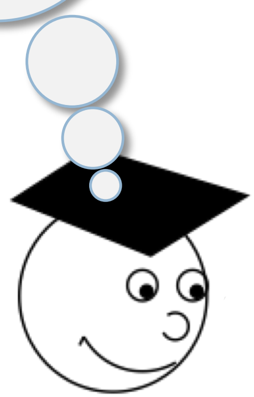
{fluid,cold} → {wet}?

yes!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]$\rightarrow$[1,0,0,1,0]

[0,0,1,0,0]$\rightarrow$[1,0,1,0,0]

[1,0,0,0,1]$\rightarrow$[1,0,1,0,1]

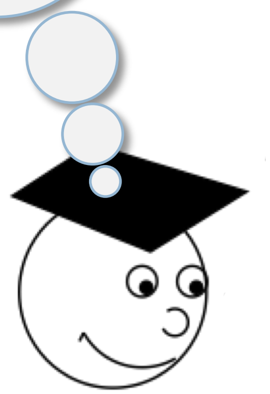
A: [1, 0, 0, 1, 0]

A'': [1, 0, 0, 1, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

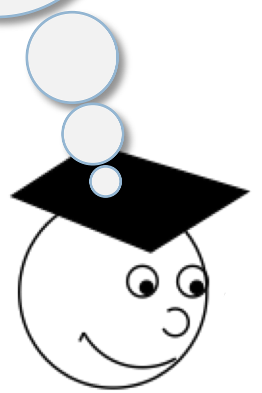
A: [1, 0, 1, 0, 0]

A'': [1, 0, 1, 0, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

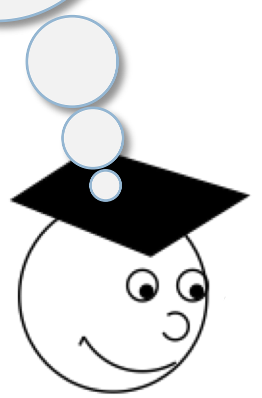
A: [1, 0, 1, 0, 1]

A'': [1, 0, 1, 0, 1]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



$[0,0,0,1,0] \to [1,0,0,1,0]$

$[0,0,1,0,0] \to [1,0,1,0,0]$

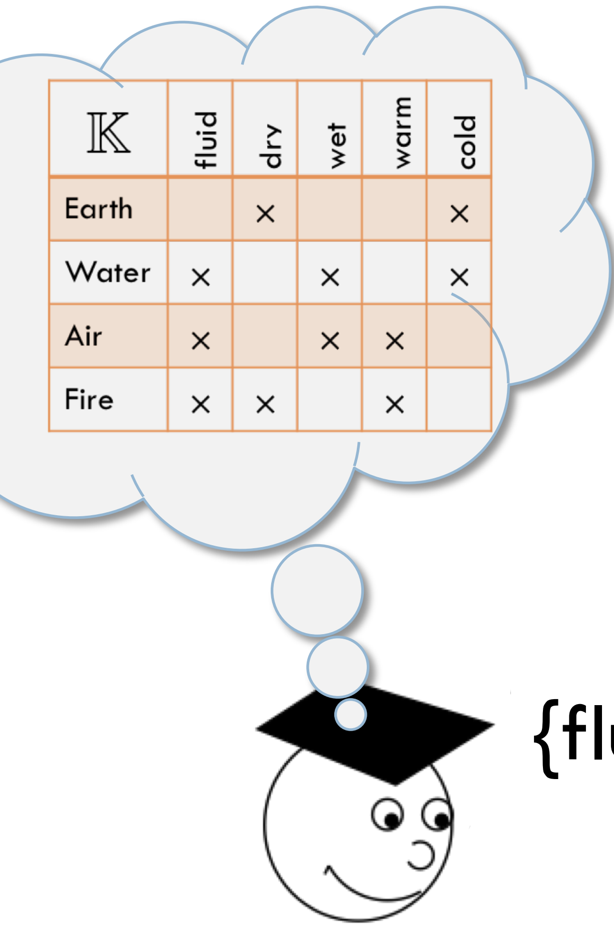
$[1,0,0,0,1] \to [1,0,1,0,1]$

A: [1, 0, 1, 1, 0]

A'': [1, 0, 1, 1, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

A: [1, 0, 1, 1, 1]

A'': [1, 1, 1, 1, 1]

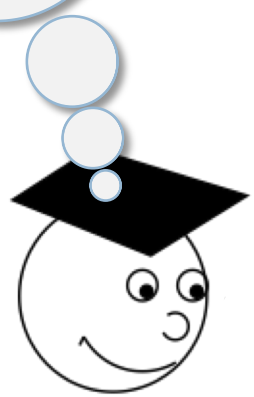
{fluid,wet,warm,cold}
→ everything?

yes!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

𝕂	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [1, 1, 0, 0, 0]

A'': [1, 1, 0, 1, 0]

{fluid,dry} → {warm}?

yes!

[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

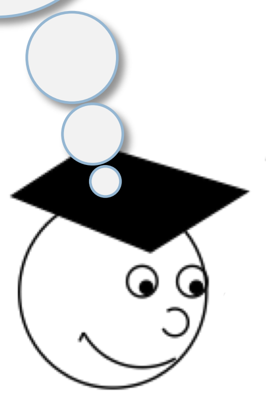
[1,0,1,1,1]→[1,1,1,1,1]

[1,1,0,0,0]→[1,1,0,1,0]

𝕂	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

[1,1,0,0,0]→[1,1,0,1,0]

A: [1, 1, 0, 1, 0]

A'': [1, 1, 0, 1, 0]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A: [1, 1, 0, 1, 1]

A'': [1, 1, 1, 1, 1]

[0,0,0,1,0]→[1,0,0,1,0]

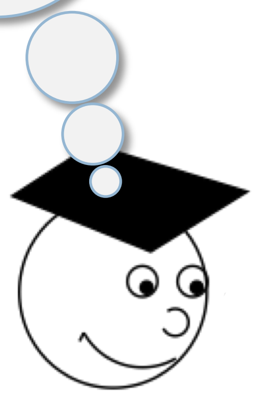
[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

[1,1,0,0,0]→[1,1,0,1,0]

[1,1,0,1,1]→[1,1,1,1,1]



{fluid,dry,warm,cold}

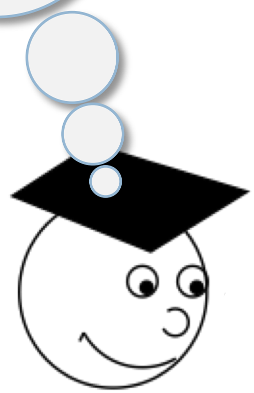
→ everything?

yes!

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [1, 1, 1, 1, 0]

A'': [1, 1, 1, 1, 1]

{fluid,dry,wet,warm}

→ everything?

yes!

[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

[1,1,0,0,0]→[1,1,0,1,0]

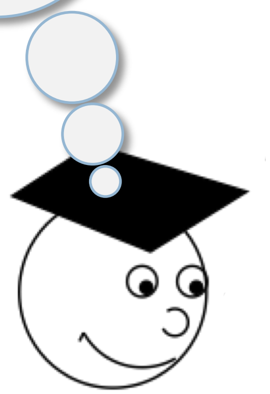
[1,1,0,1,1]→[1,1,1,1,1]

[1,1,1,1,0]→[1,1,1,1,1]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

A Tiny Example: the Four Elements

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	



A: [1, 1, 1, 1, 1]

i=0 → terminate

[0,0,0,1,0]→[1,0,0,1,0]

[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

[1,1,0,0,0]→[1,1,0,1,0]

[1,1,0,1,1]→[1,1,1,1,1]

[1,1,1,1,0]→[1,1,1,1,1]

$\mathbb{K}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	
Fire	×	×		×	

Extensions of Classical Attribute Exploration

- Allow for a-priori implications
  - Notion of *relative stem base* (Stumme 1996)
- Allow for arbitrary propositional background knowledge
  - Notion of *frame context* (Ganter 1999)
- Allow for partial description of objects
  - Notion of *partial context* (Burmeister, Holzer 2005)
- Allow for complete exploration of non-propositional logics
  - Horn logic with bounded variables: *rule exploration* (Zickwolff 1991)
  - DLs with bounded role depth: *relational exploration* (Rudolph 2004)

References I

- [Burmeister, Holzer 2005] Burmeister, P., Holzer, R., Treating Incomplete Knowledge in Formal Concept Analysis, In: B. Ganter, G. Stumme, R. Wille (Eds.): Formal Concept Analysis: State of the Art. LNAI 3626. Springer, Heidelberg 2005.
- [Ganter 1987] Ganter, B.: Algorithmen zur formalen Begriffsanalyse. In: B. Ganter, R. Wille, K.E. Wolf (Eds.): Beiträge zur Begriffsanalyse, pages 241– 254, B.I. Wissenschaftsverlag, Mannheim, 1987.
- [Ganter 1996] Ganter, B.: Attribute exploration with background knowledge. In: Theoretical Computer Science 217(2), pages 215–233, 1999.
- [Ganter 1984] Ganter, B.: Two Basic Algorithms in Concept Analysis. FB4-Preprint 831, TH Darmstadt, 1984.
- [Ganter & Wille 1999] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, 1999.

References II

- [Guigues & Duquenne 1986] Guigues, J.-L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. In: Math. Sci Humaines 95, pages 5–18, 1986.
- [Rudolph 2004] Rudolph, S.: Exploring Relational Structures via FLE . In: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.): Conceptual Structures at Work, ICCS 2004, pages 196–212, Huntsville, USA, LNCS 3127, Springer, 2004.
- [Stumme 1996] Stumme, G.: Attribute Exploration with Background Implications and Exceptions, In: H.-H. Bock, W. Polasek (eds.): Data analysis and information systems, Springer, 457--469, 1996.
- [Zickwolff 1991] Zickwolff, M.: Rule Exploration: First Order Logic in Formal Concept Analysis. PhD thesis, TH Darmstadt, 1991.

THANK YOU.

Sebastian Rudolph