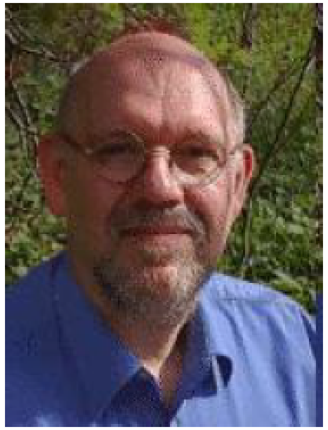
INTRODUCTION TO FORMAL CONCEPT ANALYSIS

Lecture 9

Next Closure Algorithm



Developed 1984 by Bernhard Ganter.

Can be used

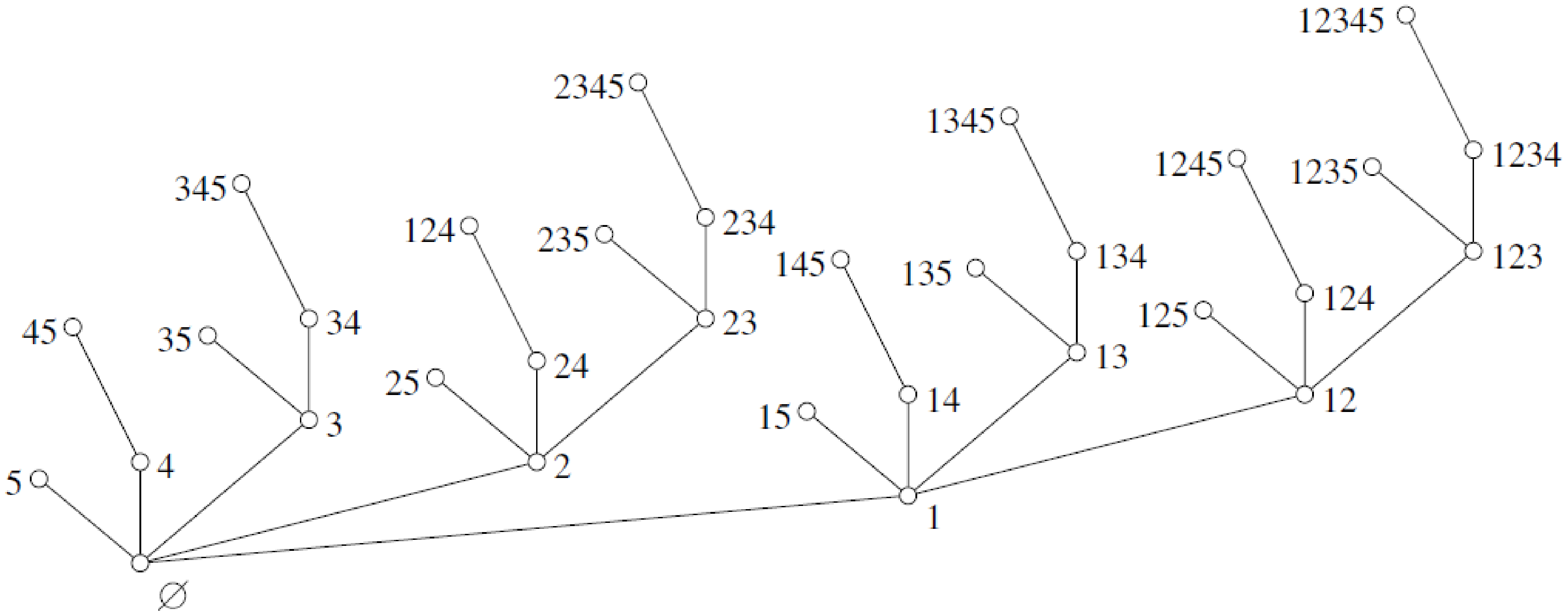
- to compute the concept lattice, or
- to compute the concept lattice together with the stem base, or
- for interactive knowledge exploration.

The algorithm computes the concept intents in the *lectic order*.

Next Closure Algorithm: Lectic Order

Let $M = \{1, \ldots, n\}$. We say that $A \subseteq M$ is *lectically smaller* than $B \subseteq M$, if $B \neq A$ and the smallest element in which A and B differ belongs to B:

$$A < B :\Leftrightarrow \exists i \in B \setminus A : A \cap \{1, 2, \ldots, i-1\} = B \cap \{1, 2, \ldots, i-1\}$$



Next Closure Algorithm: Theorem

Some definitions before we start:

$$A <_i B :\Leftrightarrow i \in B \backslash A \wedge A \cap \{1, 2, \ldots, i-1\} = B \cap \{1, 2, \ldots, i-1\}$$

$$A + i := (A \cap \{1, 2, \ldots, i-1\}) \cup \{i\}$$

Theorem

The smallest concept intent larger than a given set $A \subset M$ with respect to the lectic order is

$$A \oplus i := (A + i)'',$$

with i being the largest element of M with $A <_i A \oplus i$.

Algorithm All Closures

Input: A closure operator $X \to X''$ on a finite set M.

Output: All closed sets in lectic order.

```
Begin
    First Closure;
    repeat
            Output A;
            Next Closure;
    until not success;
end.
```

Algorithm First Closure

Input: A closure operator $X \to X''$ on a finite set M.

Output: The closure A of the empty set.

```
Begin
A := ∅'';
end.
```

Algorithm Next Closure

Input: A closure operator $X \to X''$ on a finite set M, and a subset $A \subseteq M$.

Output: A is replaced by the lectically next closed set.

```
Begin
  for all i ∈ M starting downwards with the largest element of
  M do;
            if i ∈ A then
                    A := A\{i}
else
    B := (A ∪ {i})''
    if B\A does not contain any element < i then return B
return ⊥
end.
```

	a	b	c	d	e	f	g	h
1	×		×		×		×	
2	×			×	×		×	
3				×	×		×	
4		×		×	×		×	
5	×		×			×	×	
6	×		×		×		×	
7		×	×		×		×	
8	×		×		×			×
9		×	×		×		×	
10	×		×		×		×	
11	×		×		×			×
12		×	×		×		×	
13	×		×			×	×	

The simulation starts with the first intent, $\varnothing''$, computing all intents in the lectic order:

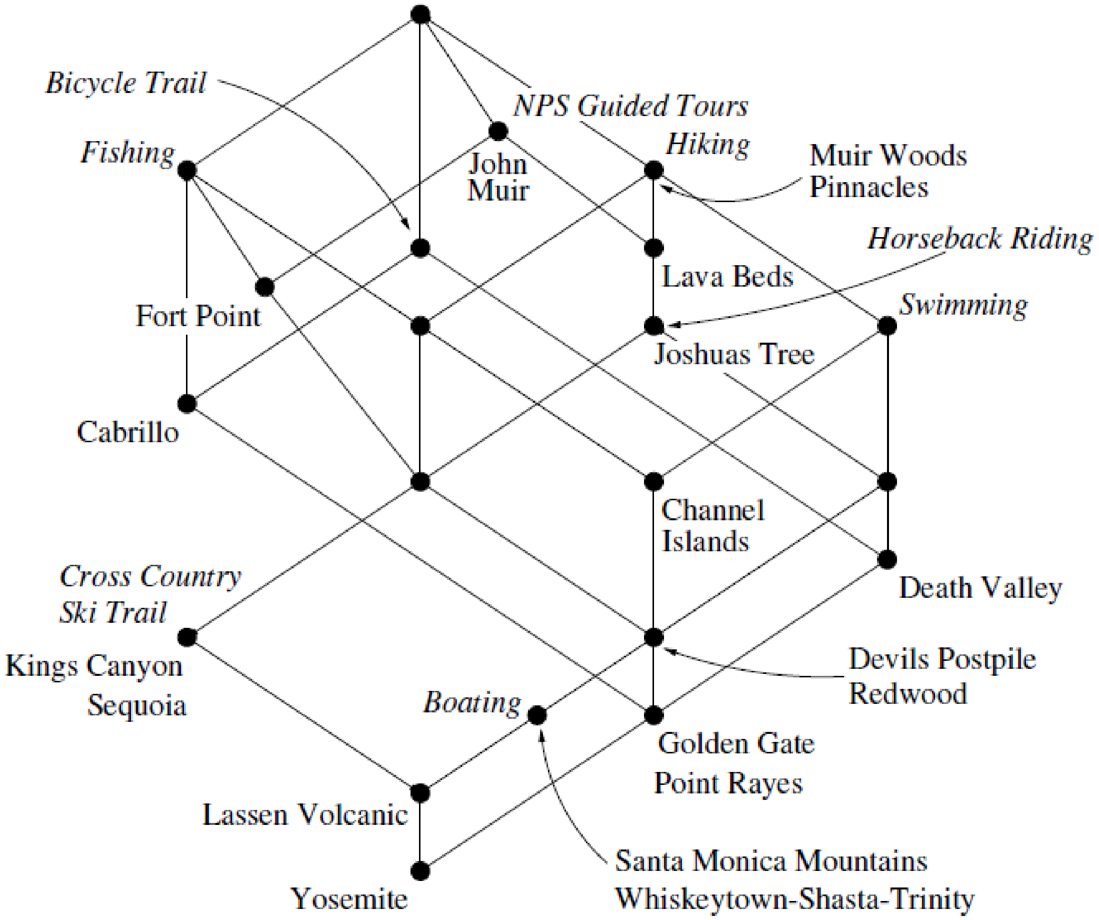
Last generated intent	A	i	$B = (A \cup \{i\})''$	smallest new element	success?
	$\emptyset$	$\emptyset''$ = first intent			yes
$\emptyset$	$\emptyset$	h	$\{a, c, e, h\}$	a	no
	$\emptyset$	g	g	g	yes
$\{g\}$	$\{g\}$	h	M	a	no
	$\emptyset$	f	$\{a, c, f, g\}$	a	no
	$\emptyset$	e	$\{e\}$	e	yes
$\{e\}$	$\{e\}$	h	$\{a, c, e, h\}$	a	no
	$\{e\}$	g	$\{e, g\}$	g	yes
$\{e, g\}$	$\{e, g\}$	h	M	a	no
	$\{e\}$	f	M	a	no
	$\emptyset$	d	$\{d, e, g\}$	d	yes
$\{d, e, g\}$	$\{d, e, g\}$	h	M	a	no
	$\{d, e\}$	f	M	a	no
	$\emptyset$	c	$\{c, g\}$	c	yes
$\{c\}$	$\{c\}$	h	$\{a, c, e, h\}$	a	no
	$\{c\}$	g	$\{c, g\}$	g	yes
$\{c, g\}$	$\{c, g\}$	h	M	a	no
	$\{c\}$	f	$\{a, c, f, g\}$	a	no
	$\{c\}$	e	$\{c, e\}$	e	yes
$\{c, e\}$	$\{c, e\}$	h	$\{a, c, e, h\}$	a	no
	$\{c, e\}$	g	$\{c, e, g\}$	g	yes
$\{c, e, g\}$	$\{c, e, g\}$	h	M	a	no
	$\{c, e\}$	f	M	a	no
	$\{c\}$	d	M	a	no
	$\emptyset$	b	$\{b, e, g\}$	b	yes

Last generated intent	A	i	$B = (A \cup \{i\})''$	smallest new element	success?
$\{b, e, g\}$	$\{b, e, g\}$	h	M	a	no
	$\{b, e\}$	f	M	a	no
	$\{b\}$	d	$\{b, d, e, g\}$	d	yes
$\{b, d, e, g\}$	$\{b, d, e, g\}$	h	M	a	no
	$\{b, d, e\}$	f	M	a	no
	$\{b\}$	c	$\{b, c, e, g\}$	c	yes
$\{b, c, e, g\}$	$\{b, c, e, g\}$	h	M	a	no
	$\{b, c, e\}$	f	M	a	no
	$\{b, c\}$	d	M	a	no
	$\emptyset$	a	$\{a\}$	a	yes
$\{a\}$	$\{a\}$	h	$\{a, c, e, h\}$	c	no
	$\{a\}$	g	$\{a, g\}$	g	yes
$\{a, g\}$	$\{a, g\}$	h	M	b	no
	$\{a\}$	f	$\{a, c, f, g\}$	c	no
	$\{a\}$	e	$\{a, e\}$	e	yes
$\{a, e\}$	$\{a, e\}$	h	$\{a, c, e, h\}$	c	no
	$\{a, e\}$	g	$\{a, e, g\}$	g	yes
$\{a, e, g\}$	$\{a, e, g\}$	h	M	b	no
	$\{a, e\}$	f	M	b	no
	$\{a\}$	d	$\{a, d, e, g\}$	d	yes
$\{a, d, e, g\}$	$\{a, d, e, g\}$	h	M	b	no
	$\{a, d, e\}$	f	M	b	no
	$\{a\}$	c	$\{a, c\}$	c	yes
$\{a, c\}$	$\{a, c\}$	h	$\{a, c, e, h\}$	e	no
	$\{a, c\}$	g	$\{a, c, g\}$	g	yes

Last generated intent	A	i	$B = (A \cup \{i\})''$	smallest new element	success?
$\{a, c, g\}$	$\{a, c, g\}$	h	M	b	no
	$\{a, c\}$	f	$\{a, c, f, g\}$	f	yes
$\{a, c, f, g\}$	$\{a, c, f, g\}$	h	M	b	no
	$\{a, c\}$	e	$\{a, c, e\}$	e	yes
$\{a, c, e\}$	$\{a, c, e\}$	h	$\{a, c, e, h\}$	h	yes
$\{a, c, e, h\}$	$\{a, c, e\}$	g	$\{a, c, e, g\}$	g	yes
$\{a, c, e, g\}$	$\{a, c, e, g\}$	h	M	b	no
	$\{a, c, e\}$	f	M	b	no
	$\{a, c\}$	d	M	b	no
	$\{a, \}$	b	M	b	yes
M					

Implications

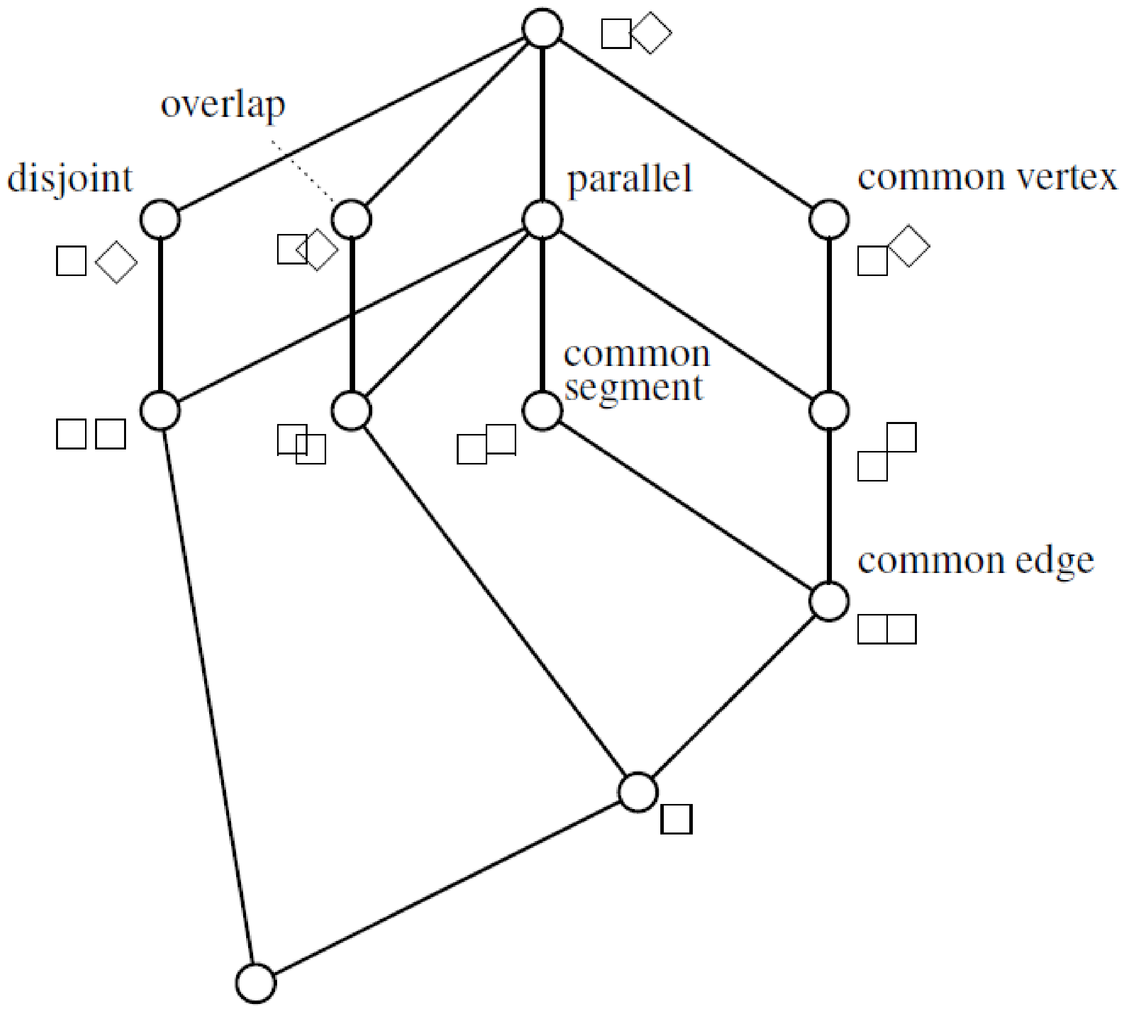
Def.: An *implication* $X \rightarrow Y$ *holds* in a context, if every object that has all attributes from X also has all attributes from Y.



Examples:

- $\{Swimming\} \rightarrow \{Hiking\}$
- $\{Boating\} \rightarrow \{Swimming, Hiking, NPS\ Guided\ Tours, Fishing, Horseback\ Riding\}$
- $\{Bicycle\ Trail, NPS\ Guided\ Tours\} \rightarrow \{Swimming, Hiking, Horseback\ Riding\}$

Attribute Logic



We are dealing with implications over an possibly *infinite* set of objects!

Concept Intents and Implications

Def.: A subset $T \subseteq M$ *respects* an implication $A \to B$,
if $A \nsubseteq T$ or $B \subseteq T$ holds.

(We then also say that T is a *model* of $A \to B$.)

T *respects a set* $\mathcal{L}$ of implications, if T respects every implication in $\mathcal{L}$.

Lemma: An implication $A \to B$ holds in a context, iff $B \subseteq A''$ ($\Leftrightarrow A' \subseteq B'$). It is then respected by all concept intents.

Implications and Closure Systems

Lemma: If $\mathcal{L}$ is a set of implications in M, then

$$\mathrm{Mod}(\mathcal{L}) := \{X \subseteq M \mid X \text{ respects } \mathcal{L}\}$$

is a closure system on M.
The respective closure operator $X \mapsto \mathcal{L}(X)$ is constructed in the following way: For a set $X \subseteq M$, let

$$X^{\mathcal{L}} := X \cup \bigcup\{B \mid A \to B \in \mathcal{L}, A \subseteq X\}.$$

We form the sets $X^{\mathcal{L}}, X^{\mathcal{L}\mathcal{L}}, X^{\mathcal{L}\mathcal{L}\mathcal{L}}, \ldots$ until a set

$$\mathcal{L}(X) := X^{\mathcal{L}\ldots\mathcal{L}}$$

is obtained with $\mathcal{L}(X)^{\mathcal{L}} = \mathcal{L}(X)$ (i.e., a fixpoint).[1] $\mathcal{L}(X)$ is then the closure of X for the closure system $\mathrm{Mod}(\mathcal{L})$.

---

[1] If M is infinite, this may require infinitely many iterations.

Implications and Closure Systems

Def.: An implication $A \to B$ *follows (semantically)* from a set $\mathcal{L}$ of implications in M if each subset of M respecting $\mathcal{L}$ also respects $A \to B$. A family of implications is called *closed* if every implication following from $\mathcal{L}$ is already contained in $\mathcal{L}$.

Lemma: A set $\mathcal{L}$ of implications in M is closed, iff the following conditions (*Armstrong Rules*) are satisfied for all $W, X, Y, Z \subseteq M$:

1. $X \to X \in \mathcal{L}$,
2. If $X \to Y \in \mathcal{L}$, then $X \cup Z \to Y \in \mathcal{L}$,
3. If $X \to Y \in \mathcal{L}$ and $Y \cup Z \to W \in \mathcal{L}$, then $X \cup Z \to W \in \mathcal{L}$.

Remark: You should know these rules from the database lecture!

Pseudo-Intents and the Stem Base

Def.: A set $\mathcal{L}$ of implications of a context (G, M, I) is called *complete*, if every implication that holds in (G, M, I) follows from $\mathcal{L}$.
A set $\mathcal{L}$ of implications is called *non-redundant* if no implication in $\mathcal{L}$ follows from other implications in $\mathcal{L}$.

Def.: $P \subseteq M$ is called *pseudo intent* of (G, M, I), if

- $P \neq P''$, and
- if $Q \subsetneq P$ is a pseudo intent, then $Q'' \subseteq P$.

Theorem: The set of implications

$$\mathcal{L} := \{P \to P'' \mid P \text{ is pseudo intent}\}$$

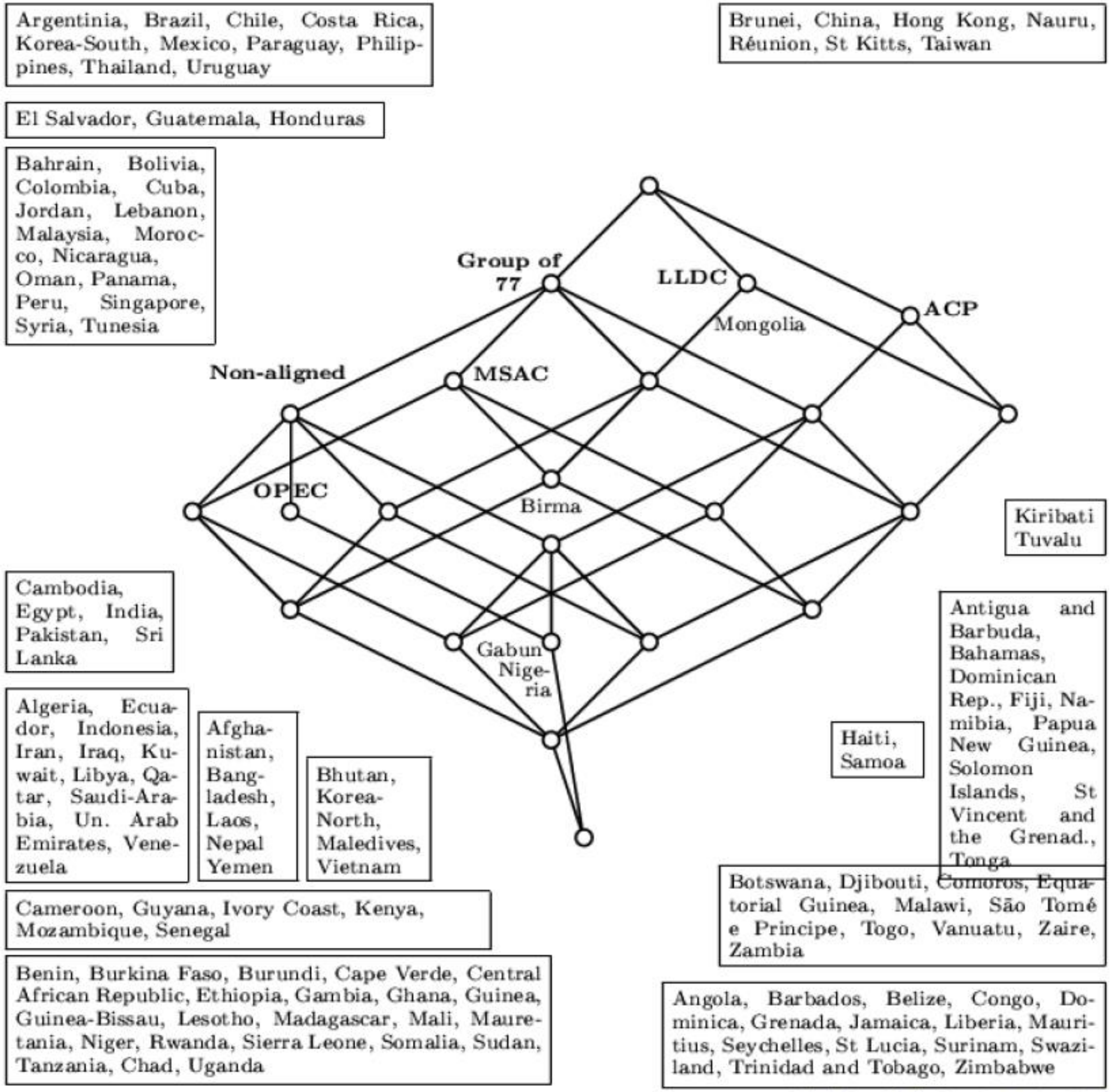
is non-redundant and complete. We call $\mathcal{L}$ the *stem base*.

Pseudo-Intents and the Stem Base

Example: membership of developing countries in supranational groups
(Source: Lexikon Dritte Welt. Rowohlt-Verlag, Reinbek 1993)

	Group of 77	Non-aligned	LLDC	MSAC	OPEC	ACP
Afghanistan	×	×	×	×		
Algeria	×	×			×	
Angola	×	×				×
Antigua and Barbuda	×					×
Argentina	×					
Bahamas	×					×
Bahrain	×	×				
Bangladesh	×	×	×	×		
Barbados	×	×				×
Belize	×	×				×
Benin	×	×	×	×		×
Bhutan	×	×	×			
Bolivia	×	×				
Botswana	×	×	×			×
Brazil	×					
Brunei						
Burkina Faso	×	×	×	×		×
Burundi	×	×	×	×		×
Cambodia	×	×		×		
Cameroon	×	×		×		×
Cape Verde	×	×	×	×		×
Central African Rep.	×	×	×	×		×
Chad	×	×	×	×		×
Chile	×					
China						
Colombia	×	×				
Comoros	×	×	×			×
Congo	×	×				×
Costa Rica	×					
Cuba	×	×				
Djibouti	×	×	×			×
Dominica	×	×				×
Dominican Rep.	×					×
Ecuador	×	×			×	
Egypt	×	×		×		
El Salvador	×			×		
Equatorial Guinea	×	×	×			×
Ethiopia	×	×	×	×		×
Fiji	×					×
Gabon	×	×			×	×
Gambia	×	×	×	×		×
Ghana	×	×	×	×		×
Grenada	×	×				×
Guatemala	×			×		
Guinea	×	×	×	×		×
Guinea-Bissau	×	×	×	×		×
Guyana	×	×		×		×
Haiti	×		×	×		×
Honduras	×			×		
Hong Kong						
India	×	×		×		
Indonesia	×	×			×	
Iran	×	×			×	
Iraq	×	×			×	
Ivory Coast	×	×		×		×
Jamaica	×	×				×
Jordan	×	×				
Kenya	×	×		×		×
Kiribati			×			×
Korea-North	×	×	×			
Korea-South	×					
Kuwait	×	×			×	
Laos	×	×	×	×		
Lebanon	×	×				
Lesotho	×	×	×	×		×
Liberia	×	×				×
Libya	×	×			×	
Madagascar	×	×	×	×		×
Malawi	×	×	×			×
Malaysia	×	×				
Maledives	×	×	×			
Mali	×	×	×	×		×
Mauretania	×	×	×	×		×
Mauritius	×	×				×
Mexico	×					
Mongolia			×			
Morocco	×	×				
Mozambique	×	×		×		×
Myanmar	×		×	×		
Namibia	×					×
Nauru						
Nepal	×	×	×	×		
Nicaragua	×	×				
Niger	×	×	×	×		×
Nigeria	×	×			×	×
Oman	×	×				
Pakistan	×	×		×		
Panama	×	×				
Papua New Guinea	×					×
Paraguay	×					
Peru	×	×				
Philippines	×					
Qatar	×	×			×	
Réunion						
Rwanda	×	×	×	×		×
Samoa	×		×	×		×
São Tomé e Principe	×	×	×			×
Saudi Arabia	×	×			×	
Senegal	×	×		×		×
Seychelles	×	×				×
Sierra Leone	×	×	×	×		×
Singapore	×	×				
Solomon Islands	×					×
Somalia	×	×	×	×		×
Sri Lanka	×	×		×		
St Kitts						
St Lucia	×	×				×
St Vincent& Grenad.	×					×
Sudan	×	×	×	×		×
Surinam	×	×				×
Swaziland	×	×				×
Syria	×	×				
Taiwan						
Tanzania	×	×	×	×		×
Thailand	×					
Togo	×	×	×			×
Tonga	×					×
Trinidad and Tobago	×	×				×
Tunisia	×	×				
Tuvalu			×			×
Uganda	×	×	×	×		×
United Arab Emirates	×	×			×	
Uruguay	×					
Vanuatu	×	×	×			×
Venezuela	×	×			×	
Vietnam	×	×	×			
Yemen	×	×	×	×		
Zaire	×	×	×			×
Zambia	×	×	×			×
Zimbabwe	×	×				×

The abbreviations stand for: LLDC := Least Developed Countries, MSAC := Most Seriously Affected Countries, OPEC := Organization of Petrol Exporting Countries, ACP := African, Caribbean and Pacific Countries.


Argentinia, Brazil, Chile, Costa Rica, Korea-South, Mexico, Paraguay, Philippines, Thailand, Uruguay
Brunei, China, Hong Kong, Nauru, Réunion, St Kitts, Taiwan
El Salvador, Guatemala, Honduras
Bahrain, Bolivia, Colombia, Cuba, Jordan, Lebanon, Malaysia, Morocco, Nicaragua, Oman, Panama, Peru, Singapore, Syria, Tunesia
Group of 77
LLDC
ACP
Mongolia
Non-aligned
MSAC
OPEC
Birma
Kiribati Tuvalu
Cambodia, Egypt, India, Pakistan, Sri Lanka
Gabun Nigeria
Antigua and Barbuda, Bahamas, Dominican Rep., Fiji, Namibia, Papua New Guinea, Solomon Islands, St Vincent and the Grenad., Tonga
Algeria, Ecuador, Indonesia, Iran, Iraq, Kuwait, Libya, Qatar, Saudi-Arabia, Un. Arab Emirates, Venezuela
Afghanistan, Bangladesh, Laos, Nepal Yemen
Bhutan, Korea-North, Maledives, Vietnam
Haiti, Samoa
Botswana, Djibouti, Comoros, Equatorial Guinea, Malawi, São Tomé e Principe, Togo, Vanuatu, Zaire, Zambia
Cameroon, Guyana, Ivory Coast, Kenya, Mozambique, Senegal
Benin, Burkina Faso, Burundi, Cape Verde, Central African Republic, Ethiopia, Gambia, Ghana, Guinea, Guinea-Bissau, Lesotho, Madagascar, Mali, Mauretania, Niger, Rwanda, Sierra Leone, Somalia, Sudan, Tanzania, Chad, Uganda
Angola, Barbados, Belize, Congo, Dominica, Grenada, Jamaica, Liberia, Mauritius, Seychelles, St Lucia, Surinam, Swaziland, Trinidad and Tobago, Zimbabwe

Pseudo-Intents and the Stem Base

stem base of the developing countries context:

$$\{\text{OPEC}\} \rightarrow \{\text{Group of 77, Non-Alligned}\}$$
$$\{\text{MSAC}\} \rightarrow \{\text{Group of 77}\}$$
$$\{\text{Non-Alligned}\} \rightarrow \{\text{Group of 77}\}$$
$$\{\text{Group of 77, Non-Alligned, MSAC, OPEC}\} \rightarrow \{\text{LLDC, AKP}\}$$
$$\{\text{Group of 77, Non-Alligned, LLDC, OPEC}\} \rightarrow \{\text{MSAC, AKP}\}$$

Computing the Stem Base With Next Closure

The algorithm Next Closure to compute all concept intents and the stem base:

1. The set $\mathcal{L}$ of all implications is initialized to $\mathcal{L} = \varnothing$.
2. The lectically first concept intent or pseudo-intent is $\varnothing$.
3. If A is an intent or a pseudo-intent, the lectically next intent/pseudo-intent is computed by checking all $i \in M \setminus A$ in descending order, until $A <_i \mathcal{L}(A + i)$ holds.
   Then $\mathcal{L}(A + i)$ is the next intent or pseudo-intent.
4. If $\mathcal{L}(A + i) = (\mathcal{L}(A + i))''$ holds, then $\mathcal{L}(A + i)$ is a concept intent, otherwise it is a pseudo-intent and the implication $\mathcal{L}(A + i) \to (\mathcal{L}(A + i))''$ is added to $\mathcal{L}$.
5. If $\mathcal{L}(A + i) = M$, finish. Else, set $A \leftarrow \mathcal{L}(A + i)$ and continue with Step 3.

ATTRIBUTE EXPLORATION

A Brief Introduction

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

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. $[0,1,0,0,1,1]+3 = [0,1,1,0,0,0]$

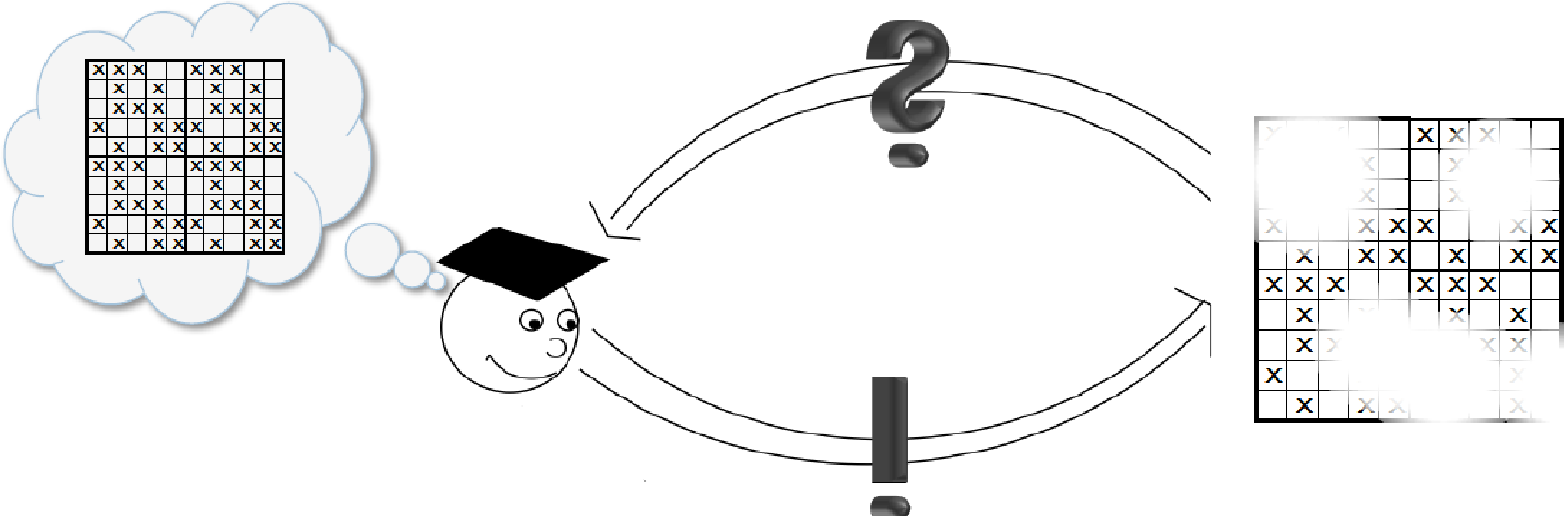
How to Compute the Stem Base

- Input formal context $\mathbb{K}$
- Create list $\Im$ of implications, initially empty
  Let A = [0,0,...,0] *(bit representation of empty set)*
- Repeat
  - Add $A \rightarrow A''$ to $\Im$ 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 $\Im$ to A+i produces 1s only at positions greater than i
  - If i=0 output $\Im$ and exit
  - Let A = $\Im$(A+i)

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

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

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 and add it to $\underline{\mathbb{K}}$
- What is a counterexample for $A \rightarrow A''$?
  - 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

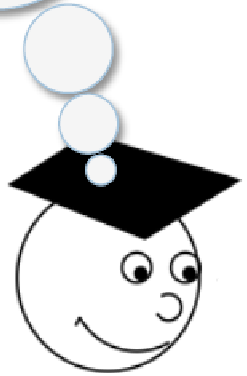
- Input formal context $\mathbb{K}$
- Create list $\Im$ of implications, initially empty
  Let A = [0,0,...,0] *(bit representation of empty set)*
- Repeat
  - While A ≠ A',
    - Ask expert whether A → A'' is valid in $\mathbb{K}$
    - If yes, add A → A'' to $\Im$ and exit while-loop,
      otherwise 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?)

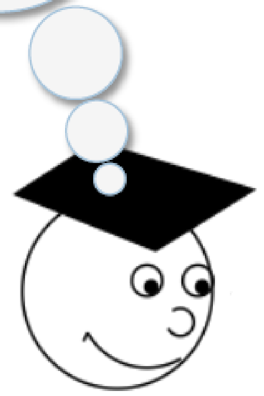
$\underline{\mathbb{K}}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×

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]



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

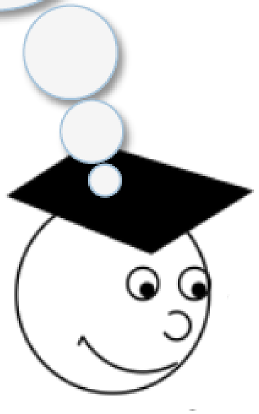
(are all elements cold?)

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	×	×		×	



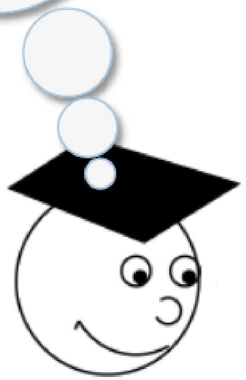
A: [0, 0, 0, 0, 0]

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

$\underline{\mathbb{K}}$	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	×	×		×	



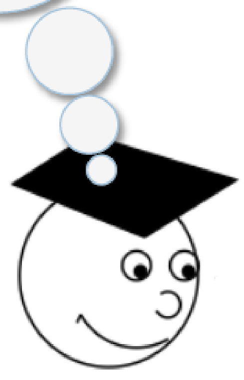
A: [0, 0, 0, 0, 1]

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

$\underline{\mathbb{K}}$	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} → {wet,fluid}?

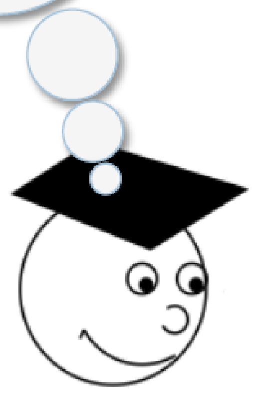
$\underline{\mathbb{K}}$	fluid	dry	wet	warm	cold
Earth		×			×
Water	×		×		×
Air	×		×	×	

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]



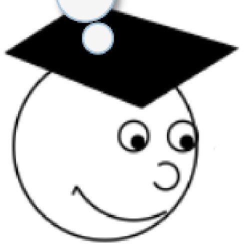
{warm} → {wet,fluid}?

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

no: fire is warm but not wet!

A Tiny Example: the Four Elements

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



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

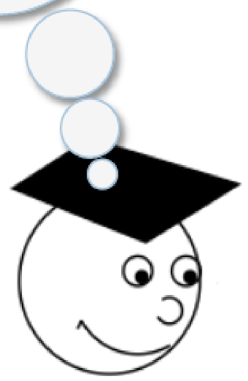
A'': [1, 0, 0, 1, 0]

{warm} → {fluid}?

$\mathbb{K}$	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]

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

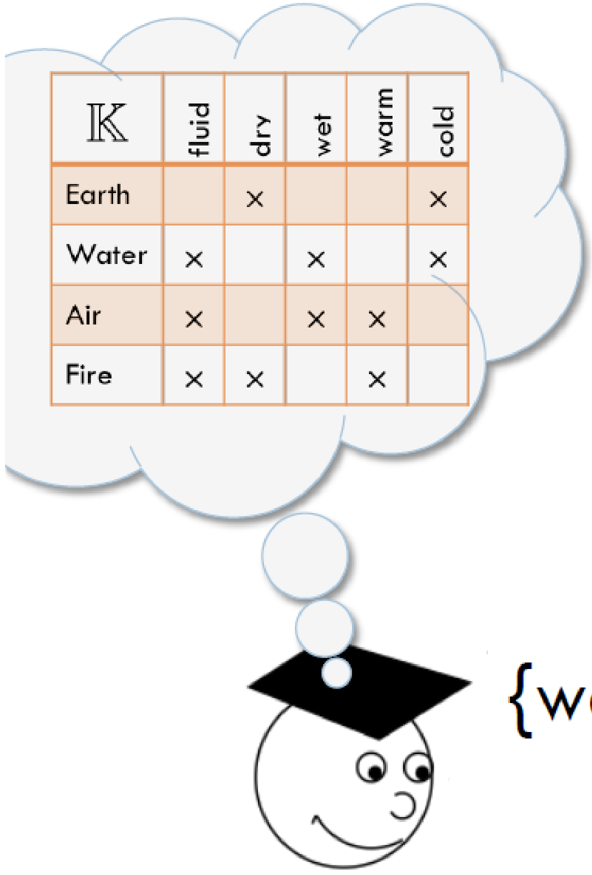
A'': [1, 0, 0, 1, 0]

{warm} → {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]

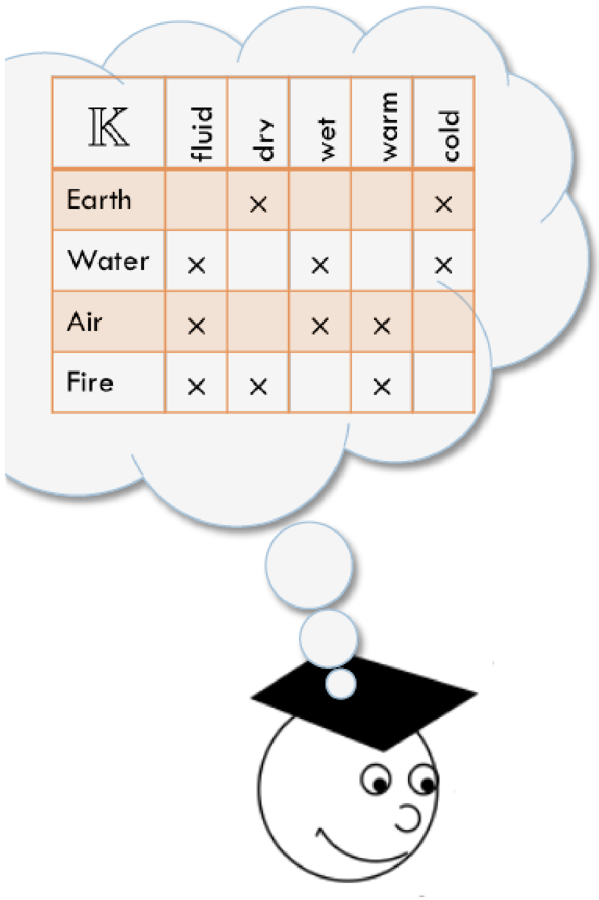
A'': [1, 0, 1, 0, 0]

{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]

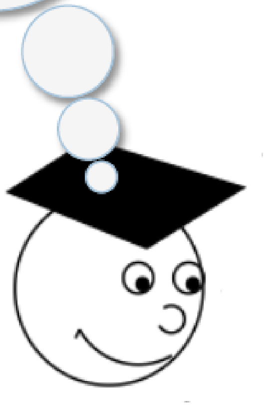
A: [0, 1, 0, 0, 0]

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

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

A Tiny Example: the Four Elements

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



[0,0,0,1,0]→[1,0,0,1,0]

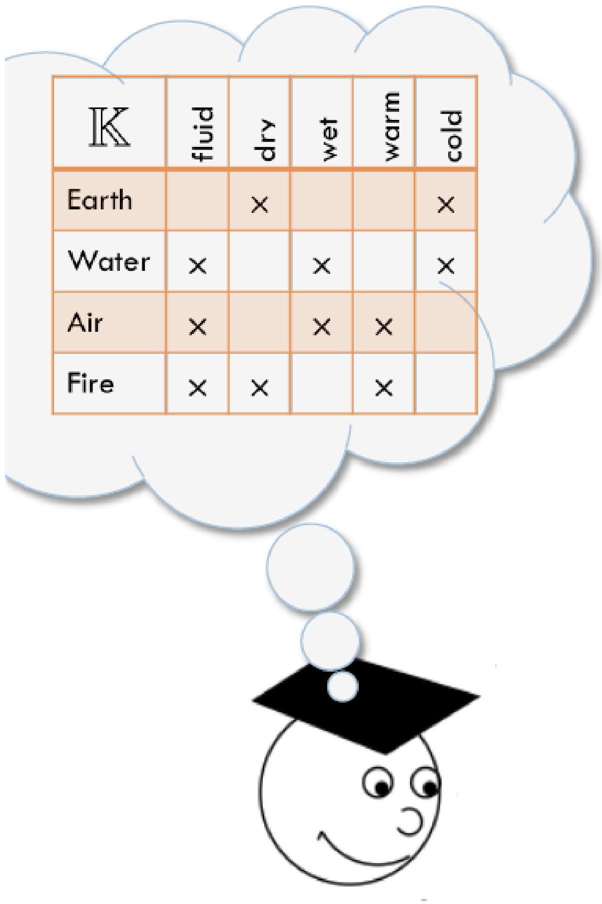
[0,0,1,0,0]→[1,0,1,0,0]

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

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

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]

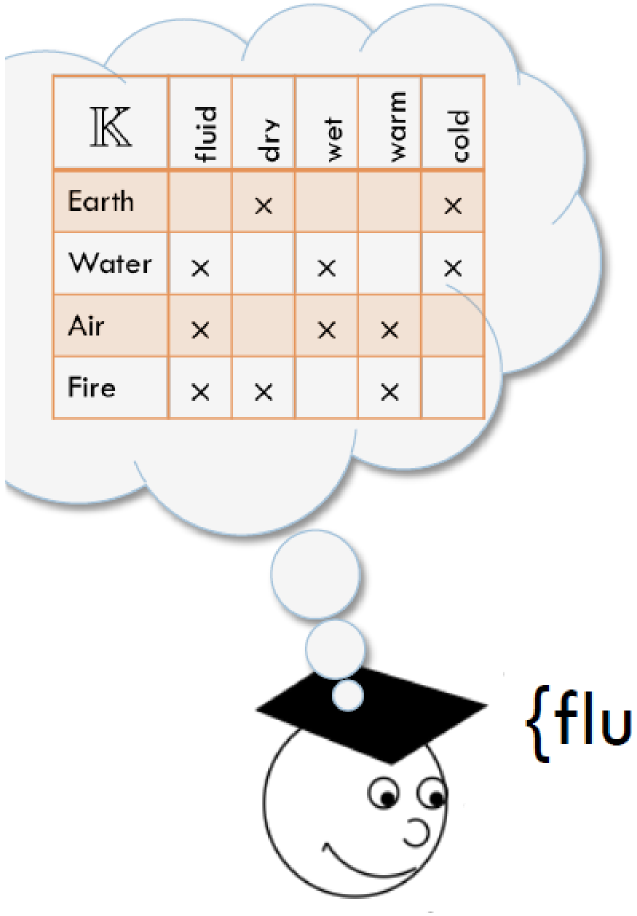
[0,0,1,0,0]⟶[1,0,1,0,0]

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

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

$\underline{\mathbb{K}}$	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]

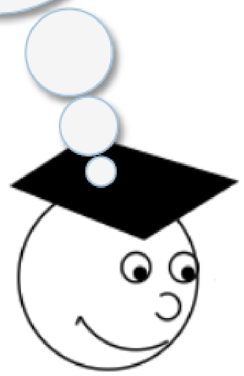
{fluid,cold} → {wet}?

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]⟶[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]

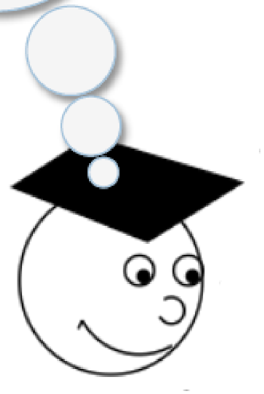
A: [1, 0, 0, 1, 0]

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

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

A Tiny Example: the Four Elements

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]

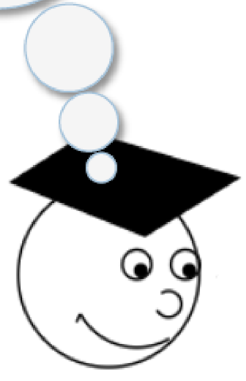
A: [1, 0, 1, 0, 0]

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

<u>𝕂</u>	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]

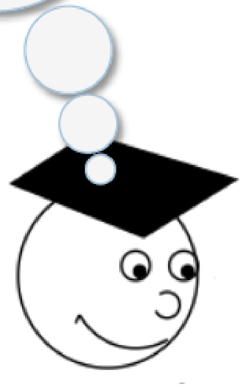
A: [1, 0, 1, 0, 1]

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]→[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, 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

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

[0,0,0,1,0]→[1,0,0,1,0]

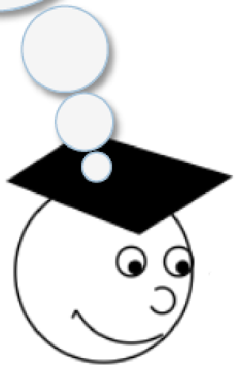
[0,0,1,0,0]→[1,0,1,0,0]

[1,0,0,0,1]→[1,0,1,0,1]

[1,0,1,1,1]→[1,1,1,1,1]

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

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



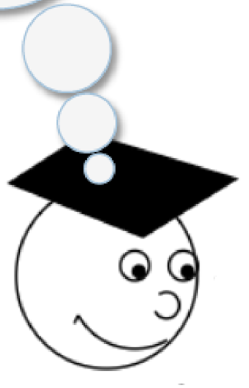
{fluid,wet,warm,cold}
→ everything?

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	×	×		×	



[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]

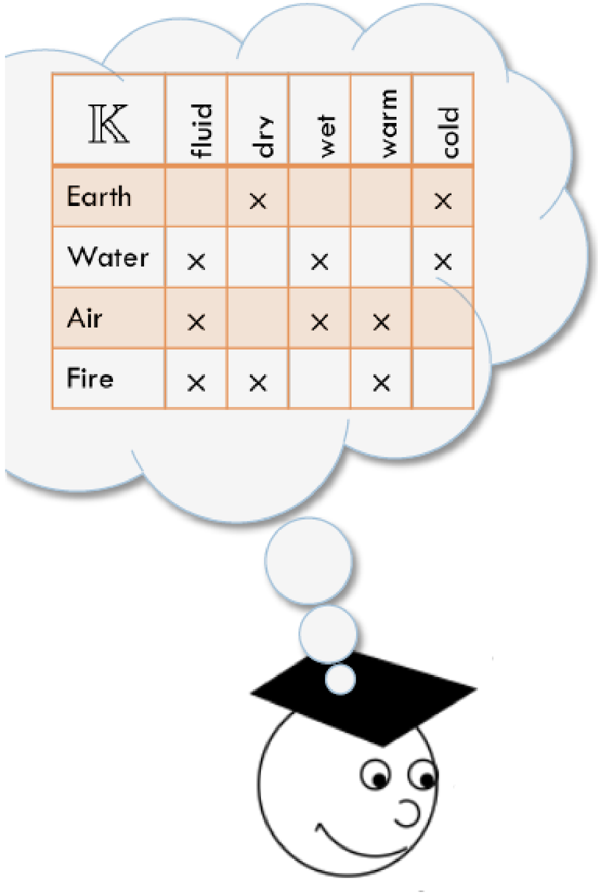
A: [1, 1, 0, 0, 0]
A'': [1, 1, 0, 1, 0]

{fluid,dry} → {warm}?

yes!

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

A Tiny Example: the Four Elements



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

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

[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]

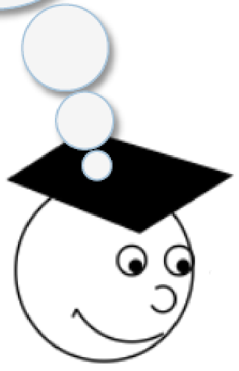
[1,0,1,1,1]→[1,1,1,1,1]

[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	×	×		×	



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

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

{fluid,dry,warm,cold}

→ 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]

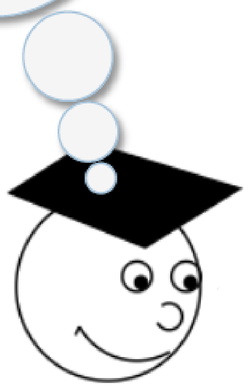
[1,1,0,0,0]→[1,1,0,1,0]

[1,1,0,1,1]→[1,1,1,1,1]

$\mathbb{K}$	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	×	×		×	



{fluid,dry,wet,warm}
→ everything?

yes!

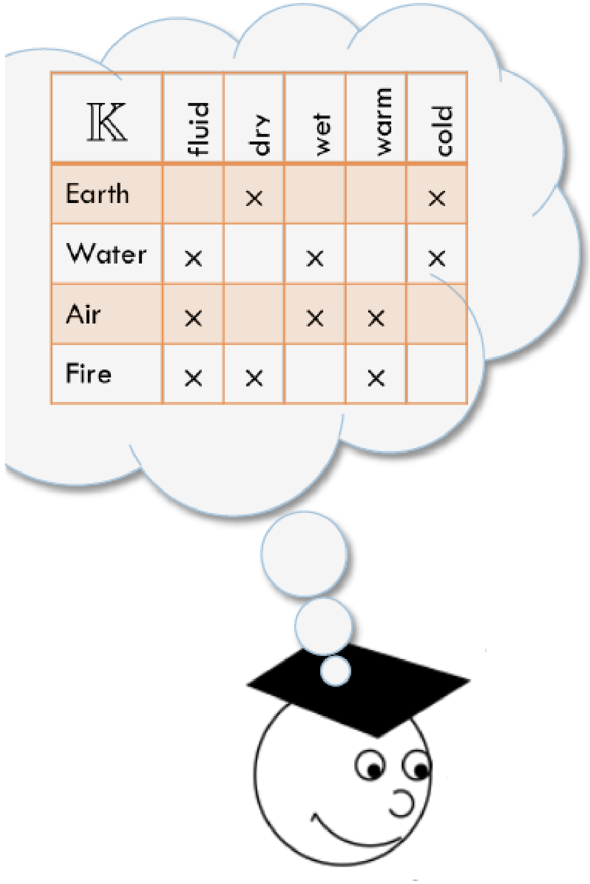
A: [1, 1, 1, 1, 0]

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]
[1,1,1,1,0]→[1,1,1,1,1]

$\underline{\mathbb{K}}$	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]

$\underline{\mathbb{K}}$	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)

Attribute Exploration

Attribute exploration allows us to compute the stem base interactively, without knowing the context beforehand (or knowing only parts of the context).

We modify the NEXT CLOSURE algorithm for computing the stem base.

The context can be modified while the list $\mathcal{L}$ of implications is computed by taking into account new objects. If these objects respect all implications that have been computed so far, then the computation can be continued with the results obtained so far. This is the result of the following Lemma:

Lemma: Let $\mathbb{K}$ be a context and let $P_1, P_2, \ldots, P_n$ be the first n pseudo-intents of $\mathbb{K}$ with respect to the lectic order. If $\mathbb{K}$ is extended by an object g the object intent g' of which respects the implications $P_i \to P_i''$, $i \in \{1, \ldots, n\}$, then $P_1, P_2, \ldots, P_n$ are also the lectically first n pseudo-intents of the extended context.

Attribute Exploration

Therefore, if we have found a new pseudo-intent P, we can stop the algorithm and ask, whether the implication $P \to P''$ should be added to $\mathcal{L}$.

The user can answer this question in the affirmative or add a counter-example, which must not contradict the implications he has confirmed so far. In the extreme case, the procedure can be started with a context the object set of which is empty. In this case, the user will have to enter all counter-examples, thereby creating a concept system with a given "attribute logic".

Instead of describing this program in detail, we shall demonstrate its functioning by means of an example: We compute the concept lattice for

$$G = \mathbb{N}$$
$$M = \{\text{even, odd, prime, square, cubic, not prime, not square, not cubic}\}$$

Attribute Exploration: Example

We start with the context

	even	odd	prime	square	cubic	not prime	not square	not cubic
1		×		×	×	×		
2	×		×				×	×
3		×	×				×	×
4	×			×		×		×
5		×	×				×	×
6	×					×	×	×
7		×	×				×	×
8	×				×	×	×	
9		×		×		×		×
27		×			×	×	×	
64	×			×	×	×		

Attribute Exploration: Example

1. suggested implication:

   $$\text{not prime, not square, not cubic} \rightarrow \text{even?}$$

   The answer is "no", since 15 is an odd number that is neither prime nor square nor cubic.

2. suggested implication:

   $$\text{cubic} \rightarrow \text{not prime?}$$

   That is true.

3. suggested implication (all attributes are contained):

   $$\text{cubic, not prime, not cubic} \rightarrow \text{even, odd, prime, square, not square?}$$

   The remark tells the user/expert that all attributes are contained to indicate that the premise could be inconsistent.

Attribute Exploration: Example

4. suggested implication:

$$\text{square} \to \text{not prime?}$$

   That is true.

5. suggested implication (all attributes are contained):

$$\text{square, not prime, not square} \to \text{even, odd, prime, cubic, not cubic?}$$

   The implication is accepted because of the attributes in the premise that are negating each other.

6. suggested implication:

$$\text{prime} \to \text{not square, not cubic?}$$

   This implication also is a property of the natural numbers.

Attribute Exploration: Example

7. suggested implication (all attributes are contained):

   prime, not prime, not square, not cubic $\rightarrow$ even, odd, square, cubic?

   Here also all attributes appear and the implication is acccepted because of the attributes in the premise that are negating each other.

8. suggested implication (all attributes are contained):

   even, odd $\rightarrow$ prime, square, cubic, not prime, not square, not cubic?

   This implication is automatically accepted.

Attribute Exploration: Example

Attribute exploration stops here. Seven implications were accepted and one counterexample added to the context. The object set

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 27, 64\}$$

but also the object set

$$\{1, 2, 3, 4, 6, 8, 9, 15, 27, 64\}$$

of the reduced context has the property, that for every non-valid implication there exists at least one counterexample.

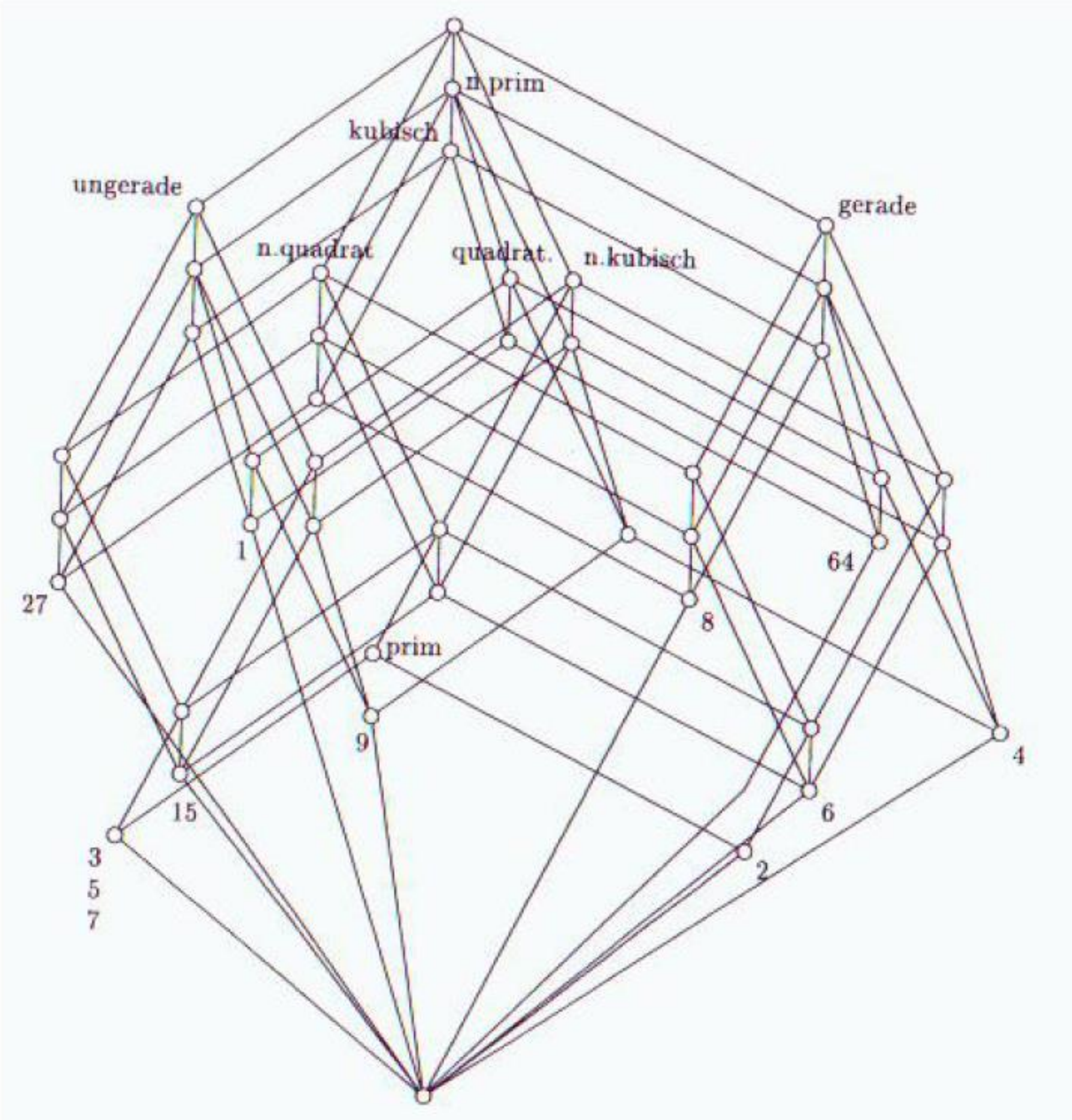
Attribute Exploration: Example

The accepted implications, i.e., the stem base, which holds for all natural numbers, looks this way:

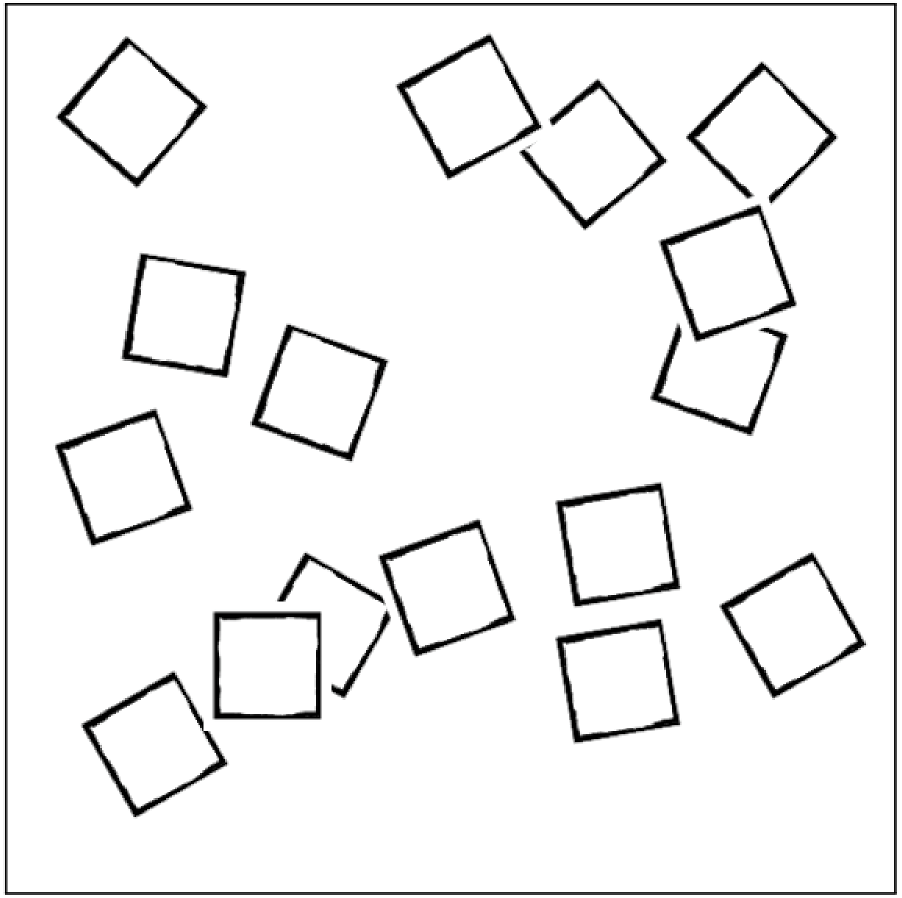
1. $\langle 4 \rangle$: cubic $\rightarrow$ not prime
2. $\langle 4 \rangle$: square $\rightarrow$ not prime
3. $\langle 4 \rangle$: prime $\rightarrow$ not square, not cubic
4. $\langle 0 \rangle$: cubic, not cubic $\rightarrow \bot$
5. $\langle 0 \rangle$: square, not square $\rightarrow \bot$
6. $\langle 0 \rangle$: prime, not prime $\rightarrow \bot$
7. $\langle 0 \rangle$: even, odd $\rightarrow \bot$

Attribute Exploration: Example

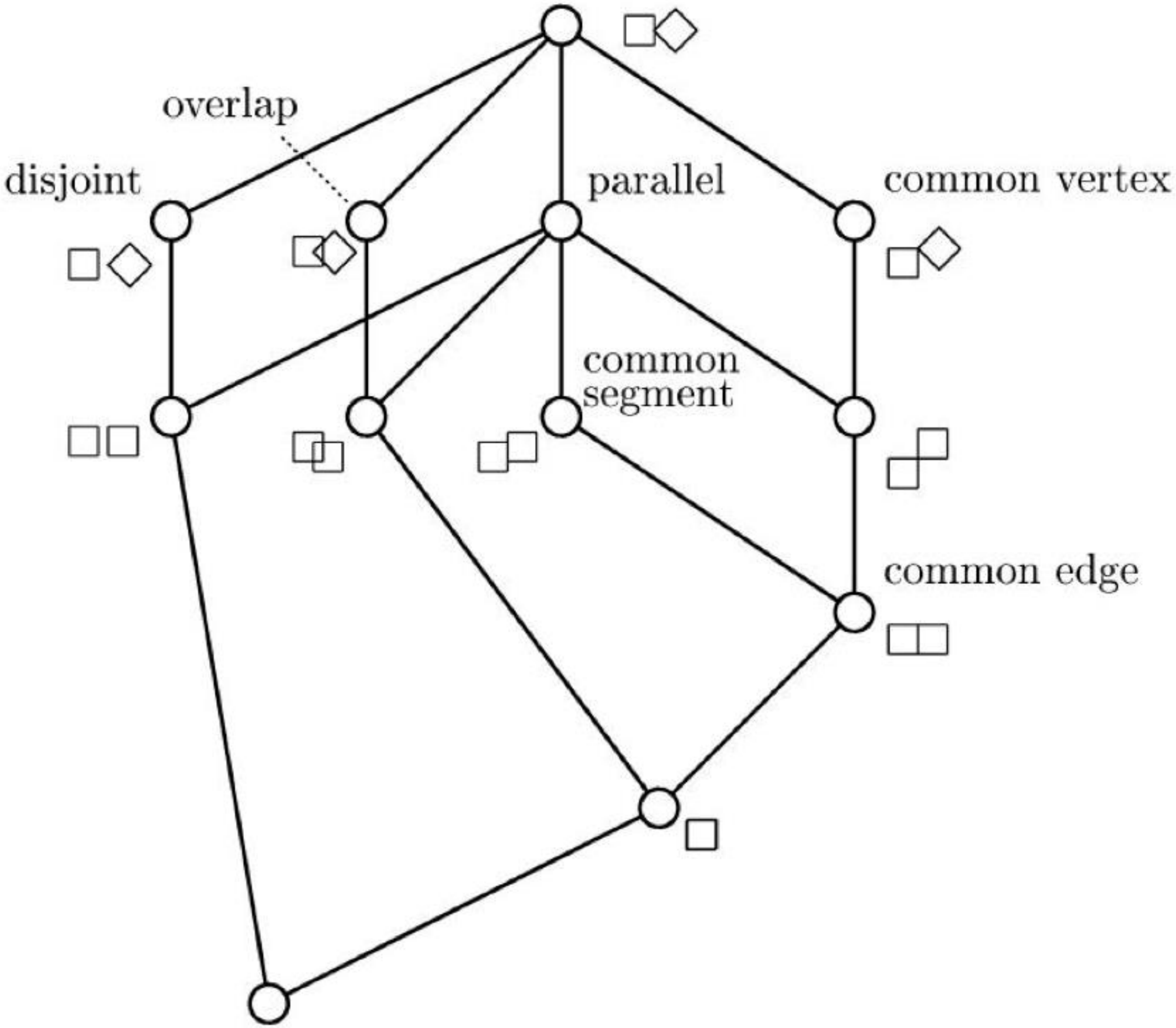
The corresponding concept lattice. All implications that can be read off hold for *all* natural numbers.



Particles under a microscope



Pairs of squares



But did we consider all possible cases?

How can we decide if our selection of examples is complete?

But did we consider all possible cases?

How can we decide if our selection of examples is complete?

A possible strategy is to prove that every implication that holds for these examples, holds in general.

- Compute the canonical base of the context of examples, and
- prove that these implications hold in general,
- or find counterexamples and extend the example set.

But did we consider all possible cases?

How can we decide if our selection of examples is complete?

A possible strategy is to prove that every implication that holds for these examples, holds in general.

- Compute the canonical base of the context of examples, and
- prove that these implications hold in general,
- or find counterexamples and extend the example set.

This can nicely be organised in an algorithm, called **attribute exploration**.

Canonical base of the example set

- common edge → parallel, common vertex, common segment
- common segment → parallel
- parallel, common vertex, common segment → common edge
- overlap, common vertex → parallel, common segment, common edge
- overlap, parallel, common segment → common edge, common vertex
- overlap, parallel, common vertex → common segment, common edge
- disjoint, common vertex → $\bot$
- disjoint, parallel, common segment → $\bot$
- disjoint, overlap → $\bot$

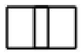
Two of the implications do not hold in general

- common edge $\rightarrow$ parallel, common vertex, common segment
- common segment $\rightarrow$ parallel
- parallel, common vertex, common segment $\rightarrow$ common edge
- overlap, common vertex $\rightarrow$ parallel, common segment, common edge
- overlap, parallel, common segment $\rightarrow$ common edge, common vertex
- overlap, parallel, common vertex $\rightarrow$ common segment, common edge
- disjoint, common vertex $\rightarrow \perp$
- disjoint, parallel, common segment $\rightarrow \perp$
- disjoint, overlap $\rightarrow \perp$

Conterexamples for the two implications

- overlap, common vertex $\rightarrow$ parallel, common segment, common edge
- Counterexample: 

Conterexamples for the two implications

- overlap, common vertex $\rightarrow$ parallel, common segment, common edge
- Counterexample: 
- overlap, parallel, common segment $\rightarrow$ common edge, common vertex
- Counterexample: 

A better choice of examples