Détournements de treillis

Véronique ventos, Henry Soldano,
Marc Champesme

Conceptual clustering

- Goal: Organizing a set of objects by building conceptual hierarchies
- Set of classes (clusters) organized using a generality order
- Each class has a symbolic description

Conceptual clustering

- Prediction
- Modelisation of a domain: summarizing large data sets

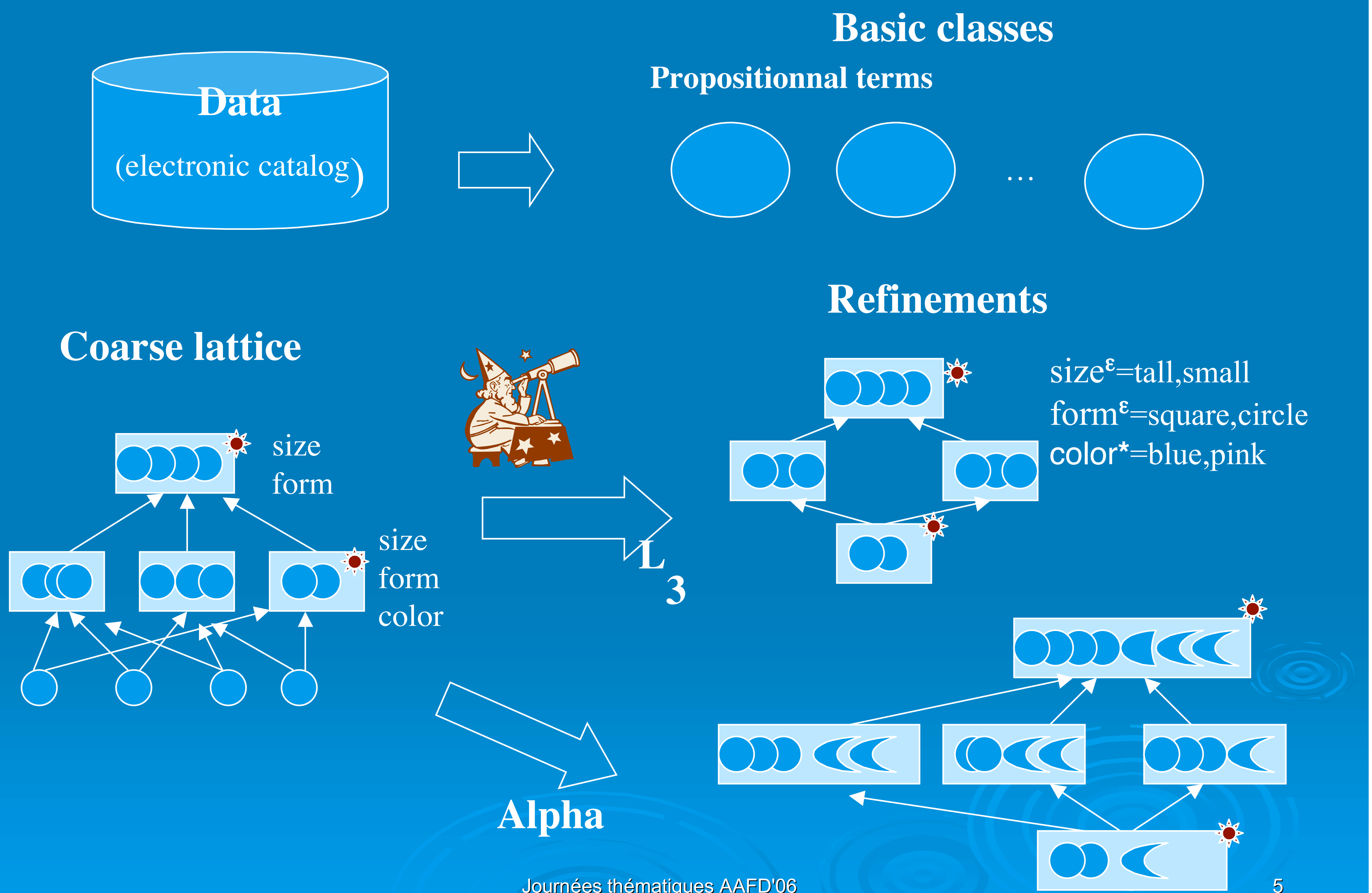
- Data structures: partitions, hierarchies (strict or not), lattices …

Galois lattices

- Principle: generate all possible distinct clusters w.r.t. a set of instances and a given language

- But: complexity issues

ZooM



Basic classes
Propositionnal terms
Data
(electronic catalog)
…
Refinements
Coarse lattice
size
form
size
form
color
L3
size$^{\varepsilon}$=tall,small
form$^{\varepsilon}$=square,circle
color*=blue,pink
Alpha

1. Définitions alpha
2. Comparaison avec les treillis de concepts fréquents
3. Règles alpha
4. Travaux en cours

Galois Connections

- Let m1: P → Q and m2: Q → P be maps between two ordered sets (P,≤) and (Q,≤). Such a pair of maps is called a Galois connection if:

1. $p1 \leq p2 \Rightarrow m1(p1) \geq m1(p2)$
2. $q1 \leq q2 \Rightarrow m2(q1) \geq m2(q2)$
3. $p \leq m2(m1(p))$ and $q \leq m1(m2(q))$

Galois connection: an example

- Let *int* and *ext* be two maps such that:
- *int*: $\mathcal{P}(\mathcal{I}) \rightarrow \mathcal{L}$ and *ext*: $\mathcal{L} \rightarrow \mathcal{P}(\mathcal{I})$ with:
  - *int*(e1) = set of attributes common to the instances in e1 (Least Common Subsumer)
  - *ext*(c1) = set of instances which have all attributes in c1 ($\{i \in \mathcal{I}$ such that i *isa* c1$\}$)
- *int* and *ext* define a Galois connection

Treillis de Galois

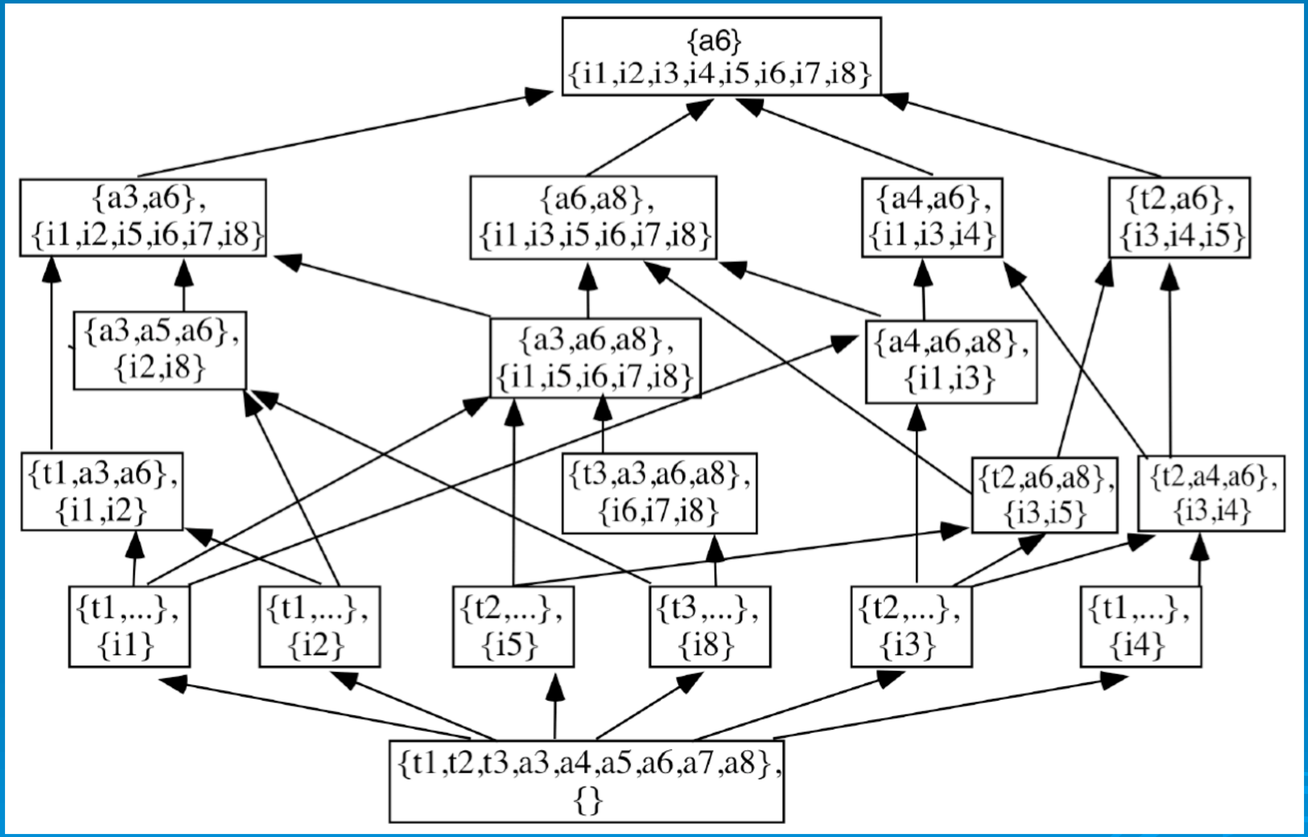
- Let G = {(c,e) such that c is a closed term of $\mathcal{L}$ (c = *int*(e)) and e is a closed element of $\mathcal{P}(\mathcal{I})$ (e = *ext*(c))}

- G with ≤ such that (c1,e1) ≤ (c2,e2) iff e1 ⊆ e2 denoted as G(*int*,*ext*) is a **Galois lattice**

Instance descriptions

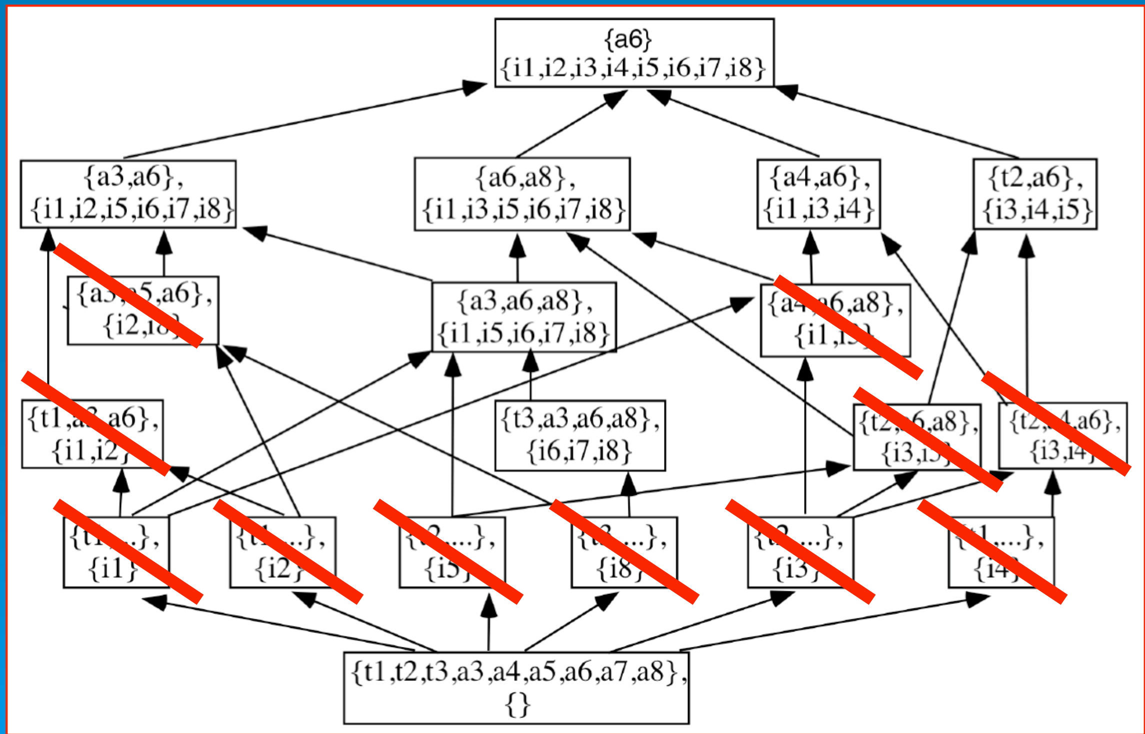
	t1	t2	t3	a3	a4	a5	a6	a7	a8
i1	1			1	1		1		1
i2	1			1		1	1		
i3		1			1		1		1
i4		1			1		1	1	
i5		1		1			1		1
i6			1	1			1		1
i7			1	1			1		1
i8			1	1		1	1		1

A Galois lattice



Frequent concept lattices

- Closed frequent item sets organized in a lattice



1. Alpha definitions

- Change the extensional map (*ext*) according to a partition of data

- Ex: the attributes t1, t2 and t3 express the type of the instances, they allow us to construct 3 basic classes BC1, BC2, BC3

Basic classes

BC1={i1,i2}, *int*(BC1)= {t1,a3,a6}

BC2={i3,i4,i5}, *int*(BC2)= {t2,a6}

BC3={i6,i7,i8}, *int*(BC3)= {t3,a3,a6,a8}

	t1	t2	t3	a3	a4	a5	a6	a7	a8
i1	1			1	1		1		1
i2	1			1		1	1		
i3		1			1		1		1
i4		1			1		1	1	
i5		1		1			1		1
i6			1	1			1		1
i7			1	1			1		1
i8			1	1		1	1		1

Alpha Galois lattices

- **Alpha satisfaction**: let $\propto$ be a number belonging to [0,100]. Let e be a subset of instances and T a term of the language:

  e *alpha satisfies* T (e $sat_{\propto}$ T) iff

  $|ext_e(T)| \geq |e| \cdot \propto /100$

Alpha Galois lattices

Alpha membership relation :

i isa_{α} T iff i *isa* T and BCl(i) sat_{α} T

Alpha extension: let $\mathcal{I}$ be a set of instances, BC a set of basic classes, and T a term:

$ext_{\alpha}(T) = \{i \in \mathcal{I}$ such that i isa_{α} T$\}$

Alpha extension

- ext $_{\alpha}$(T) = {i ∈ ℐ such that i *isa* T and BC(i) sat $_{\alpha}$ T}

T={a6,a8}

ext(T) = *ext*$_{0}$(T)=
{i1, i3,i5, i6,i7,i8}

ext$_{60}$(T)= {i3,i5, i6,i7,i8}

ext$_{100}$(T)={i6,i7,i8}

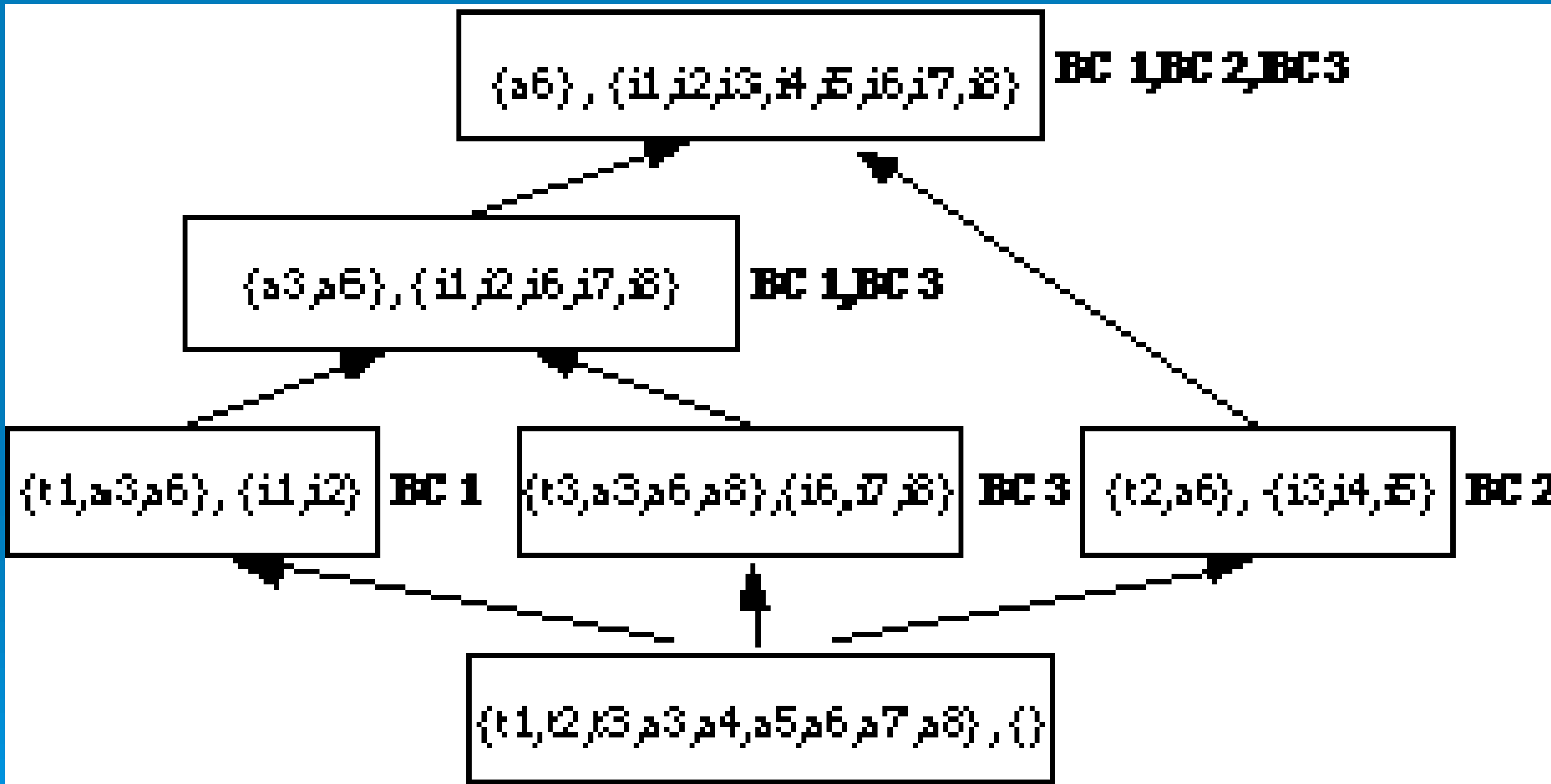
	t1	t2	t3	a3	a4	a5	a6	a7	a8
i1	1			1	1		1		1
i2	1			1		1	1		
i3		1			1		1		1
i4		1			1		1	1	
i5		1		1			1		1
i6			1	1			1		1
i7			1	1			1		1
i8			1	1		1	1		1

Alpha Galois lattices

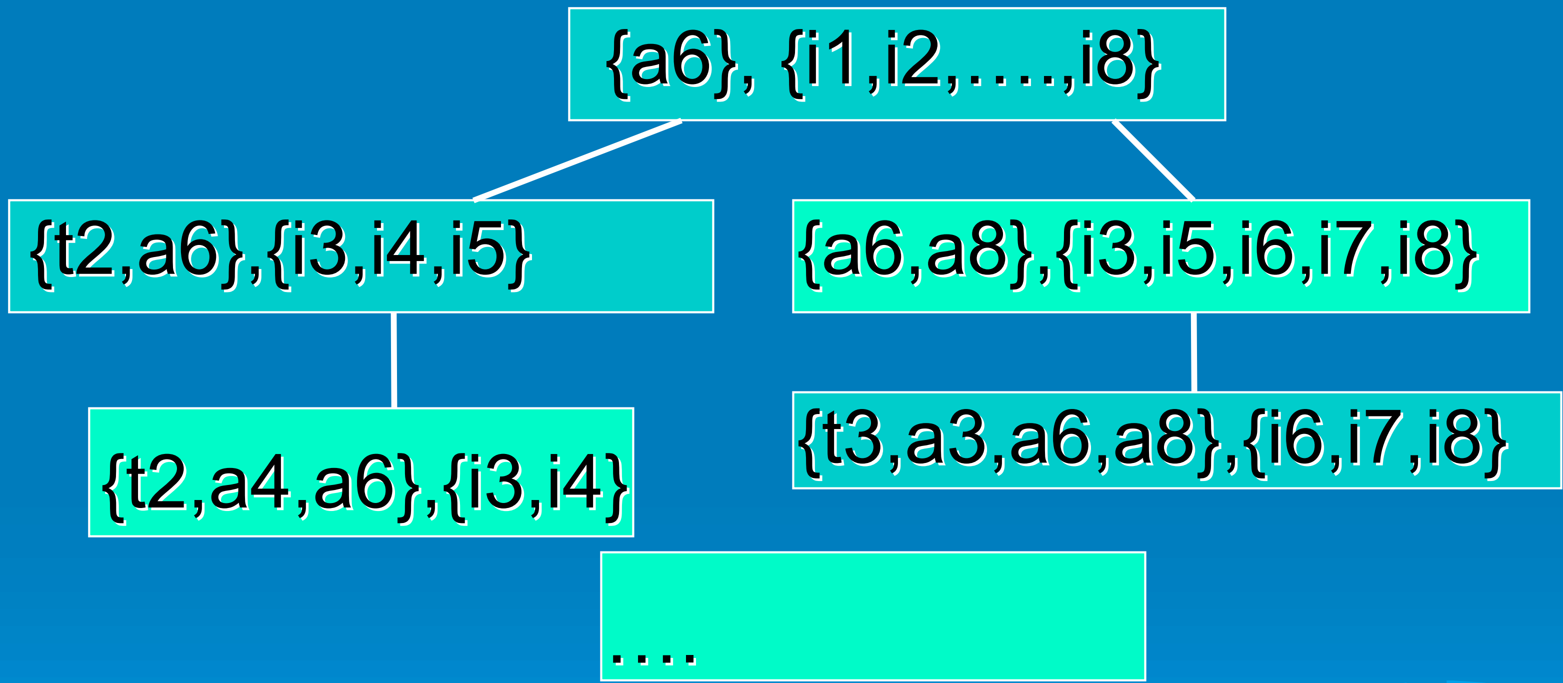
- Let G = {(c,e) such that c is a closed term of $\mathcal{L}$ (c = *int*(e)) and e is a closed element of a subset of $\mathcal{P}(\mathcal{I})$ (e = ext_α (c)}

- G(*int*, ext_α) with ≤ such that (c1,e1) ≤ (c2,e2) iff e1 $\subseteq$ e2 is a Alpha **Galois lattice** nested in G(*int*,*ext*)

$\alpha = 100$


{a6}, {i1,i2,i3,i4,i5,i6,i7,i8}
BC 1,BC 2,BC 3
{a3,a6},{i1,i2,i6,i7,i8}
BC 1,BC 3
{t1,a3,a6}, {i1,i2}
BC 1
{t3,a3,a6,a8},{i6,i7,i8}
BC 3
{t2,a6}, {i3,i4,i5}
BC 2
{t1,t2,t3,a3,a4,a5,a6,a7,a8}, {}

α =60



{a6}, {i1,i2,….,i8}
{t2,a6},{i3,i4,i5}
{a6,a8},{i3,i5,i6,i7,i8}
{t2,a4,a6},{i3,i4}
{t3,a3,a6,a8},{i6,i7,i8}
….

2. Comparison

- One basic class : alpha Galois lattice = frequent concept lattice
- Several basic classes : local frequency vs global frequency

Experiments (Cnet)

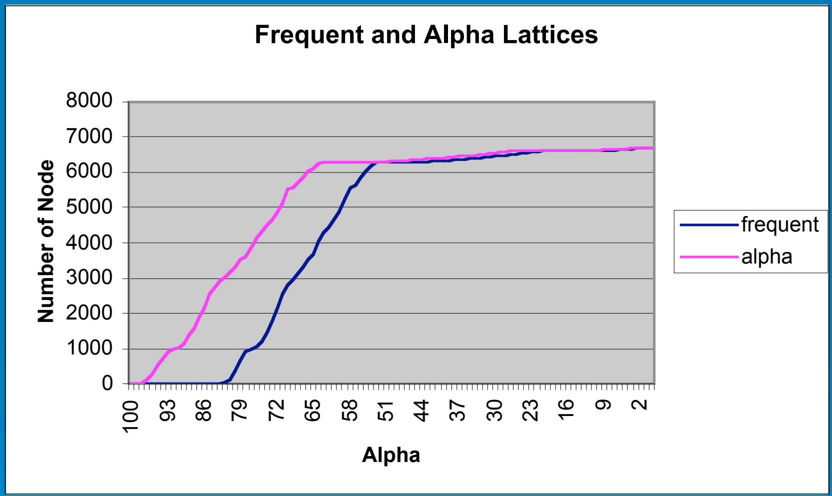
- 2274 products
- 59 basic classes
- 234 attributes
- 8 attributes (average) for a basic class description

Experiments

Alpha	100	98	96	94	92	91
Nodes	211	664	8198	44021	100734	165369

……a global view is untractable……

Experiment 1:
alpha vs frequent



3 not homogeneous classes :

Laptop 252 instances (39 attributes)

H-D 45 instances (22 attributes)

N-S 4 instances (16 attributes)

Experiment 1

$\alpha = 100 \rightarrow 92$: a node appears under *Hard-Drive* : products sold with « support » (42 on 45). This attribute is not globally frequent (42 on 301).

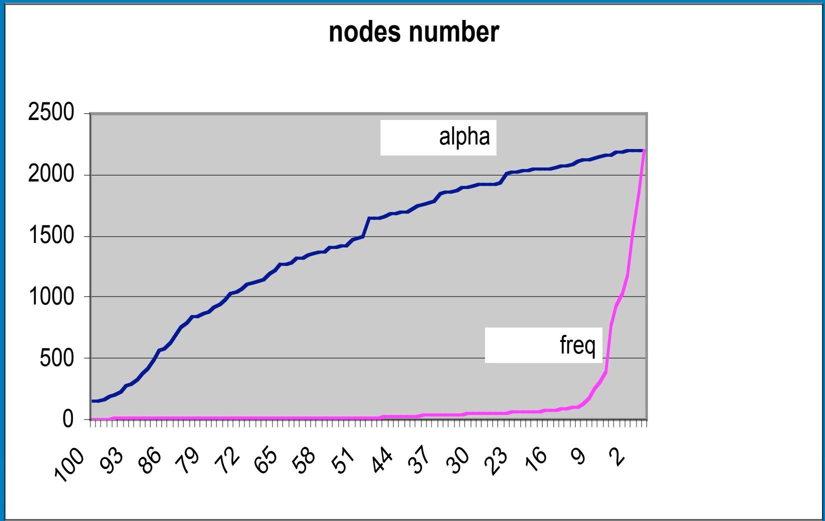
$\alpha = 0 \rightarrow 6$: nodes are removed under *Laptop*: the attribute DigitalSignalProcessor is really unfrequent (exceptional attribute)

$$ext_6(\{laptop,\ldots, DSP\}) = \emptyset$$

Experiment 2 :

Alpha vs Frequent

24 homogeneous classes



Few attributes are globally frequent

(pseudo) classification des espèces

mammals | insects | birds

A	B	C	D	E	F	G	H
mammals	Flies	Has_Wings	Has_Feath...	Has_Hairs	Viviparous	Has_Teats	Oviparous
cow	0	0	0	X	X	X	0
flying_squi...	X	0	0	X	X	X	0
tiger	0	0	0	X	X	X	0
rabbit	0	0	0	X	X	X	0
! platypus	0	0	0	X	0	X	X
bat	X	X	0	X	X	X	0

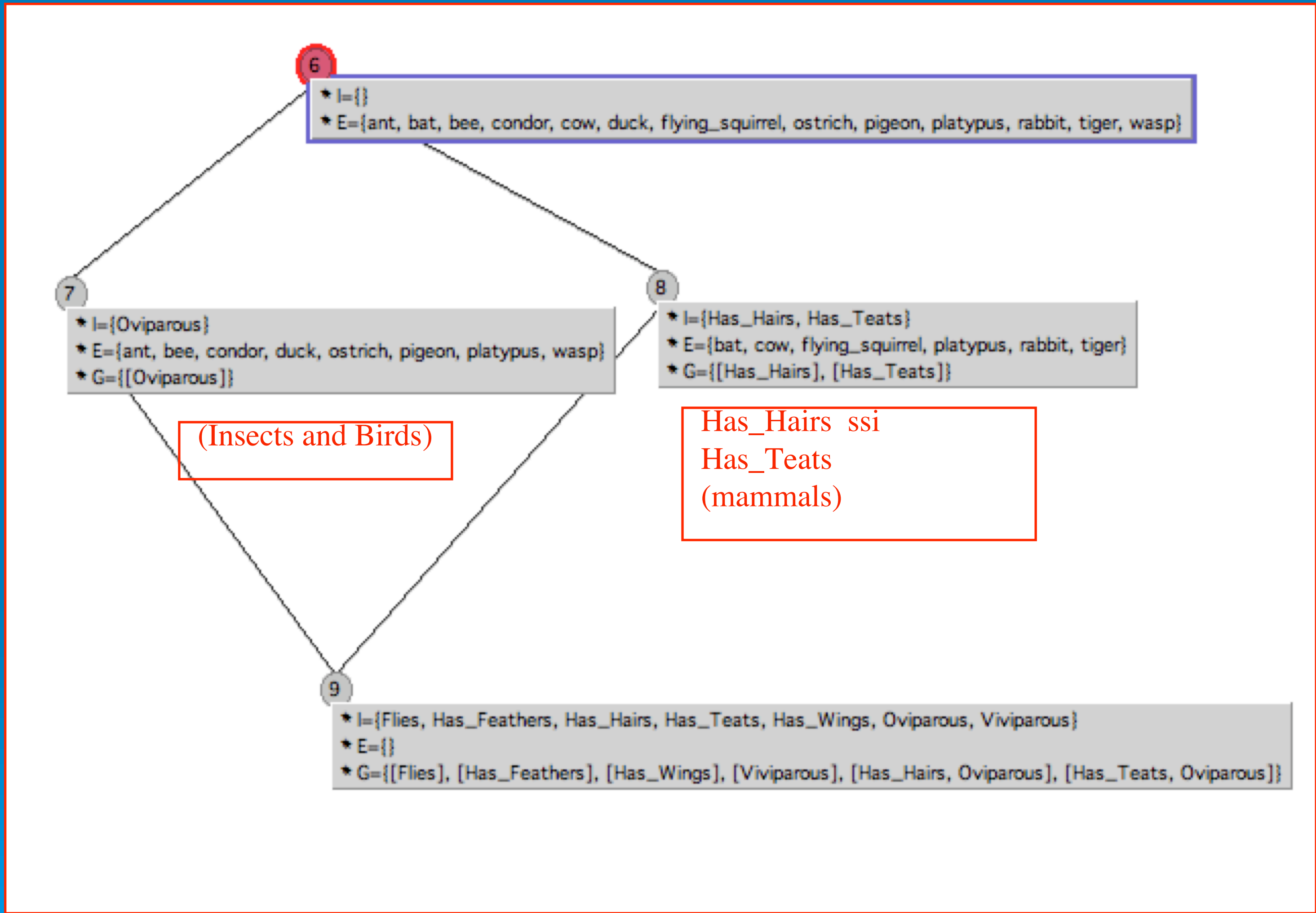
mammals | insects | birds

A	B	C	D	E	F	G	H
insects	Flies	Has_Wings	Has_Feath...	Has_Hairs	Viviparous	Has_Teats	Oviparous
ant	0	0	0	0	0	0	X
bee	X	X	0	0	0	0	X
wasp	X	X	0	0	0	0	X

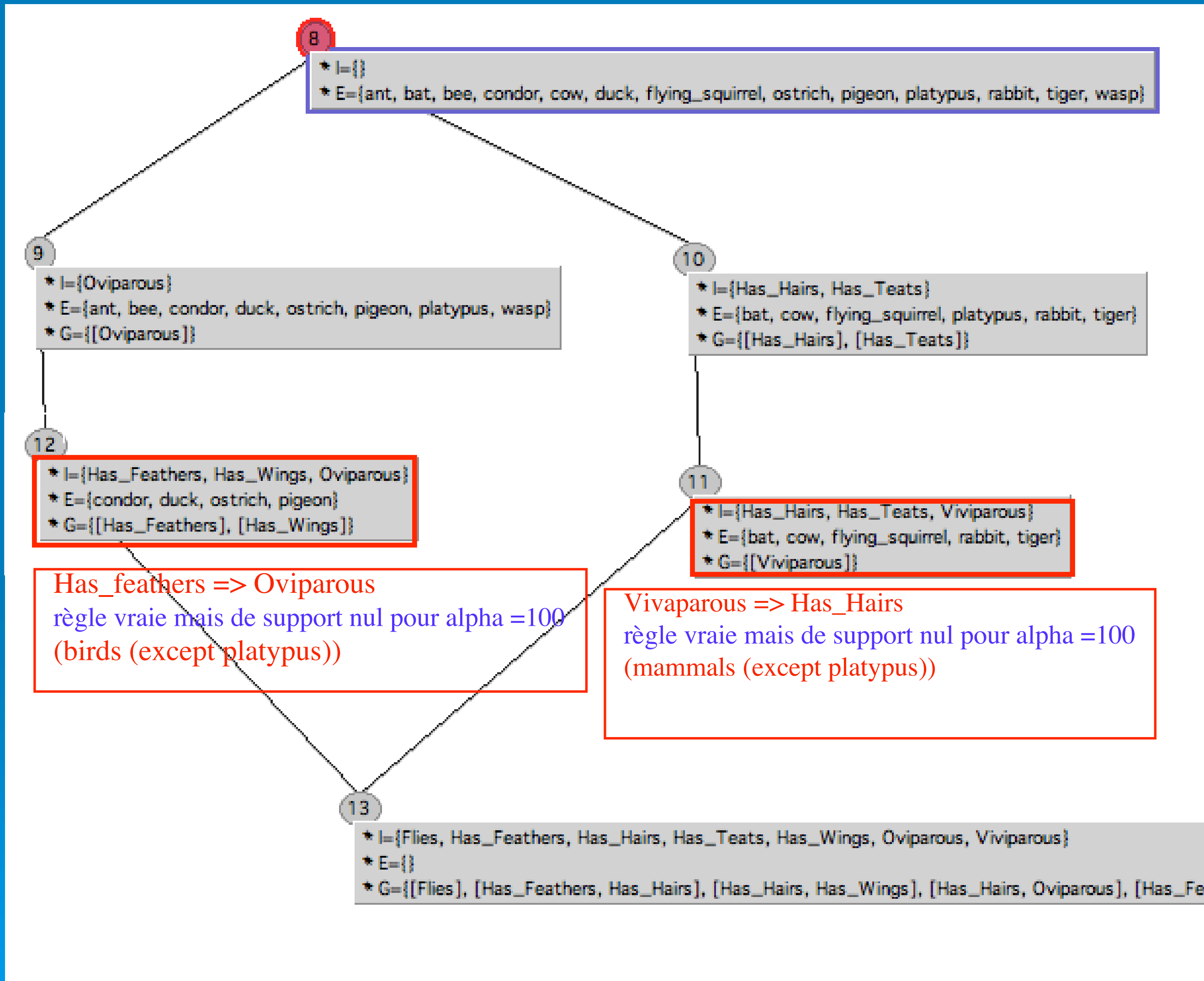
mammals | insects | birds

A	B	C	D	E	F	G	H
birds	Flies	Has_Wings	Has_Feath...	Has_Hairs	Viviparous	Has_Teats	Oviparous
condor	X	X	X	0	0	0	X
duck	X	X	X	0	0	0	X
ostrich	0	X	X	0	0	0	X
! platypus	0	0	0	X	0	X	X
pigeon	X	X	X	0	0	0	X

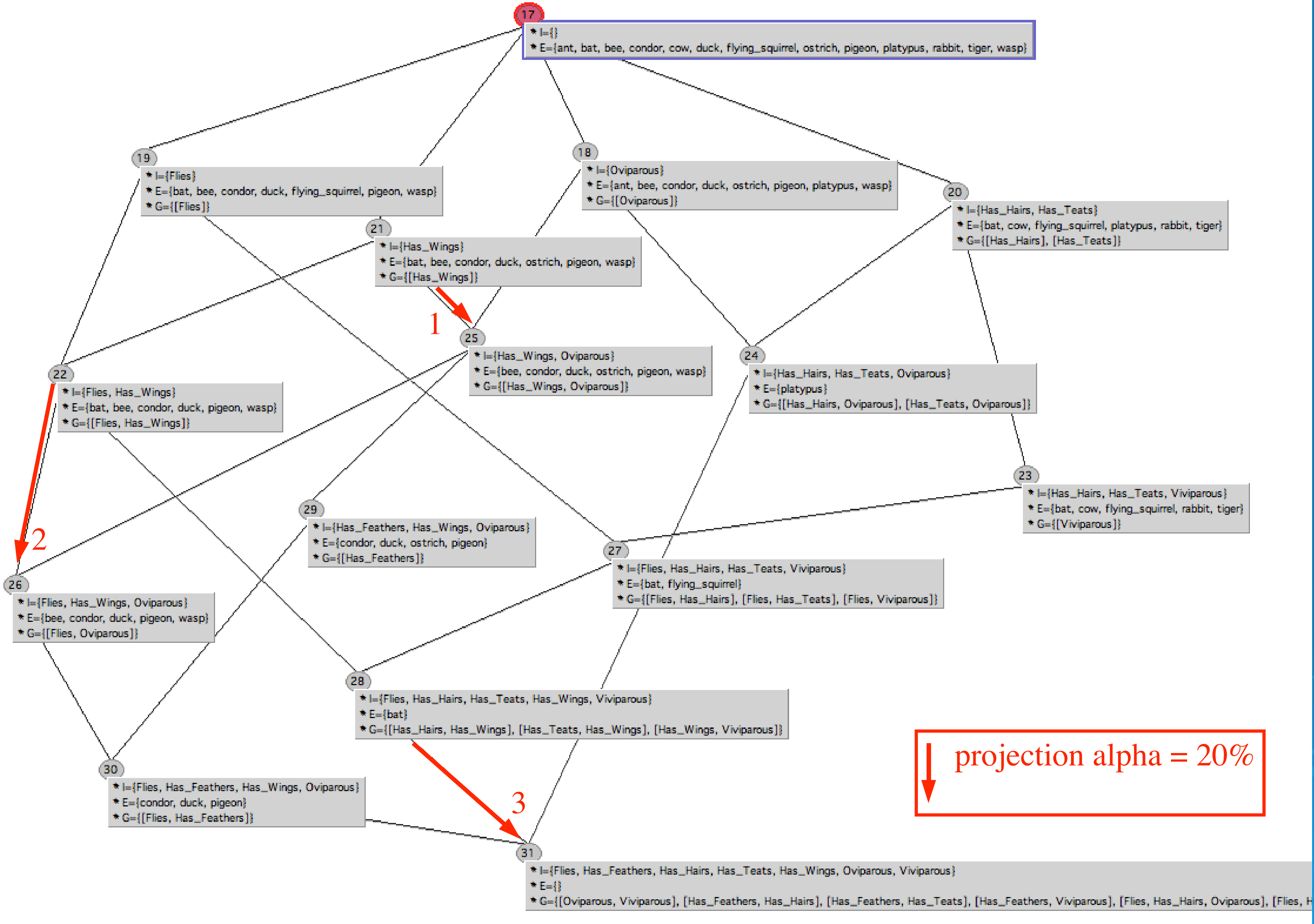
Les concepts alpha = 100%



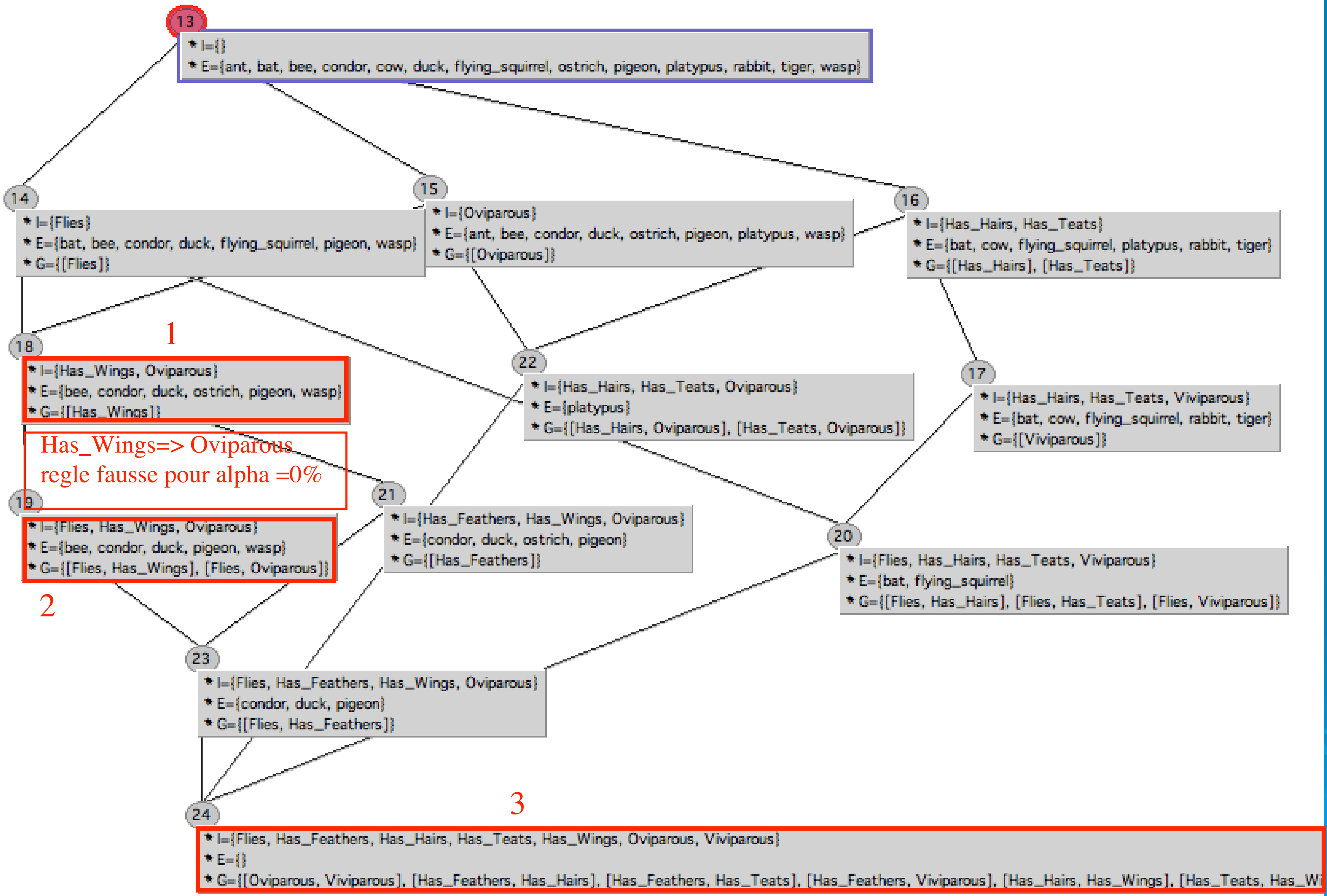
Les concepts alpha=80%



Les concepts originaux



Les concepts alpha=20%



4. Alpha rules

int(ext({square})) = {square, blue}

square → square,blue (implication rule)

square ≈≈> square,blue, small (association rule)

int(ext_{α}({square})) = {square, blue, small}

square $\rightarrow_{\alpha}$ square,blue, small

- *if* t1 $\rightarrow_{\alpha}$ t2 *and* t2 $\rightarrow_{\alpha}$ t3 *then* t1 $\rightarrow_{\alpha}$ t3
- *If* t1 $\rightarrow_{\alpha}$ t2 *then for all* $\alpha' > \alpha$ t1 $\rightarrow_{\alpha'}$ t2

→ Transitivity, monotonicity

Alpha Association rules

Definition 11 *An α-association rule is a pair of terms T_1 and T_2, denoted as $T_1 \rightarrow_\alpha T_2$.*

The support and confidence of an α-association rule $r = T_1 \rightarrow_\alpha T_2$ are defined as follows :

$$\alpha\text{-}supp(r) = \frac{|ext_\alpha(T_1 \cup T_2)|}{|I|}$$

$$\alpha\text{-}conf(r) = \frac{|ext_\alpha(T_1 \cup T_2)|}{|ext_\alpha(T_1)|}$$

The α-association rule $r = T_1 \rightarrow_\alpha T_2$ holds on the pair $(I, \mathcal{BC})$ whenever $\alpha\text{-}supp(r) \geq minsupp$ and $\alpha\text{-}conf(r) \geq minconf$.

Guigues-Duquenne and Luxenburger Bases of 100-Rules and 60-Rules

Alpha =100

Règles exactes GD

R0 :	$\rightarrow$ a6	Supp = 1.0	Conf = 1.0
R1 :	t1 $\rightarrow$ a3	Supp = 0.25	Conf = 1.0
R2 :	t3 $\rightarrow$ a3 a8	Supp = 0.37	Conf = 1.0
R3 :	a8 $\rightarrow$ t3 a3	Supp = 0.75	Conf = 1.0

Règles approximatives L

R4 :	a6 $\rightarrow$ a3	Supp = 0.62	Conf = 0.62
R5 :	a3 a6 $\rightarrow$ t3 a8	Supp = 0.37	Conf = 0.6

–

Alpha = 60

Règles exactes GD

R0 :	$\rightarrow$ a6	Supp = 1.0	Conf = 1.0
R1 :	t1 $\rightarrow$ a3	Supp = 0.25	Conf = 1.0
R2 :	t3 $\rightarrow$ a3 a8	Supp = 0.37	Conf = 1.0
R3 :	a4 $\rightarrow$ t2	Supp = 0.37	Conf = 1.0
R4 :	a3 a8 $\rightarrow$ t3	Supp = 0.62	Conf = 1.0

Règles approximatives L

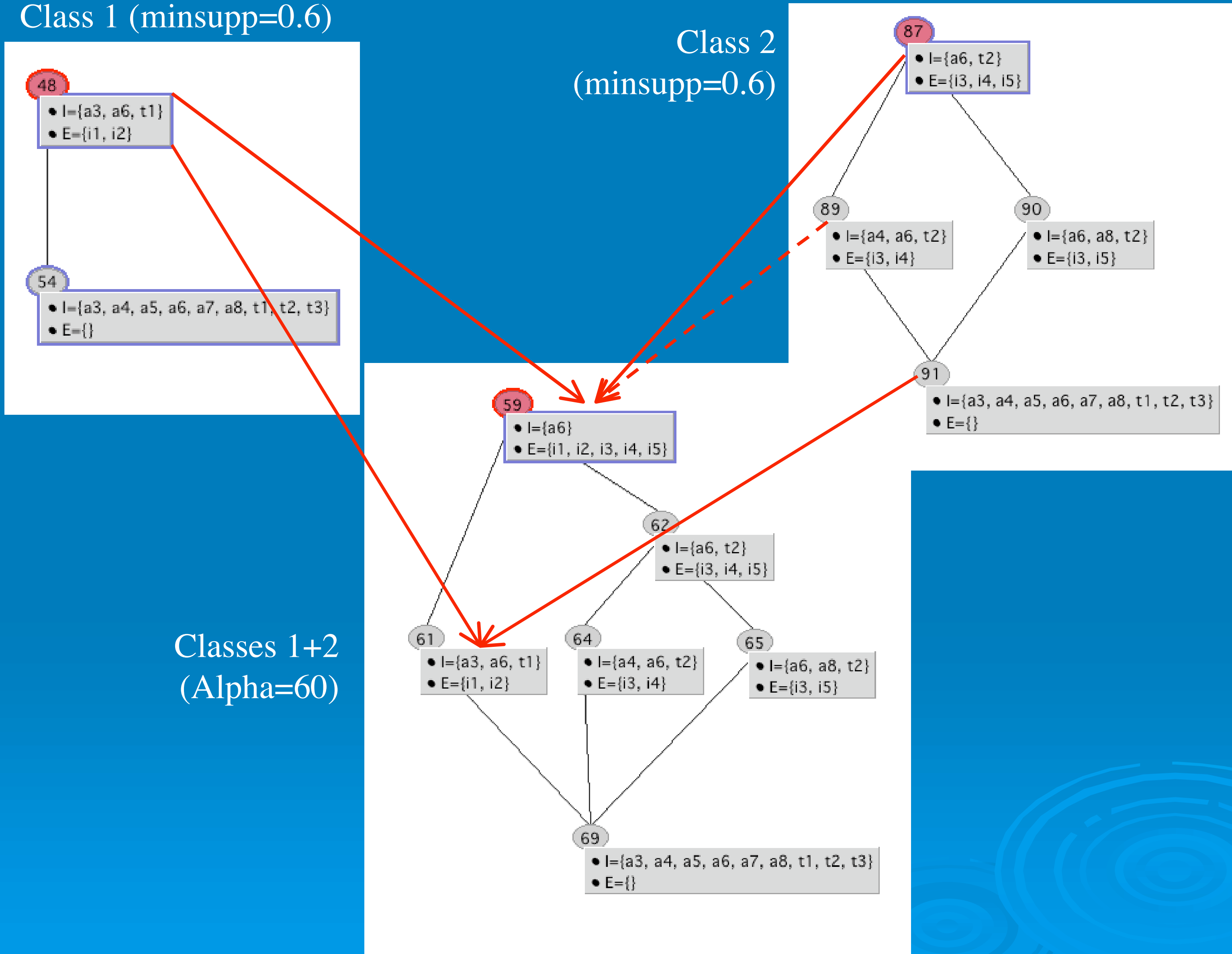
R5 :	a6 $\rightarrow$ a3	Supp = 0.62	Conf = 0.62
R6 :	a6 $\rightarrow$ a8	Supp = 0.62	Conf = 0.62
R7 :	t2 a6 $\rightarrow$ a4	Supp = 0.25	Conf = 0.66
R8 :	t2 a6 $\rightarrow$ a8	Supp = 0.25	Conf = 0.66
R9 :	a3 a6 $\rightarrow$ t3 a8	Supp = 0.37	Conf = 0.6
R10 :	a6 a8 $\rightarrow$ t3 a3	Supp = 0.37	Conf = 0.6

Rules extracted from a *« Frequent » Alpha lattice*

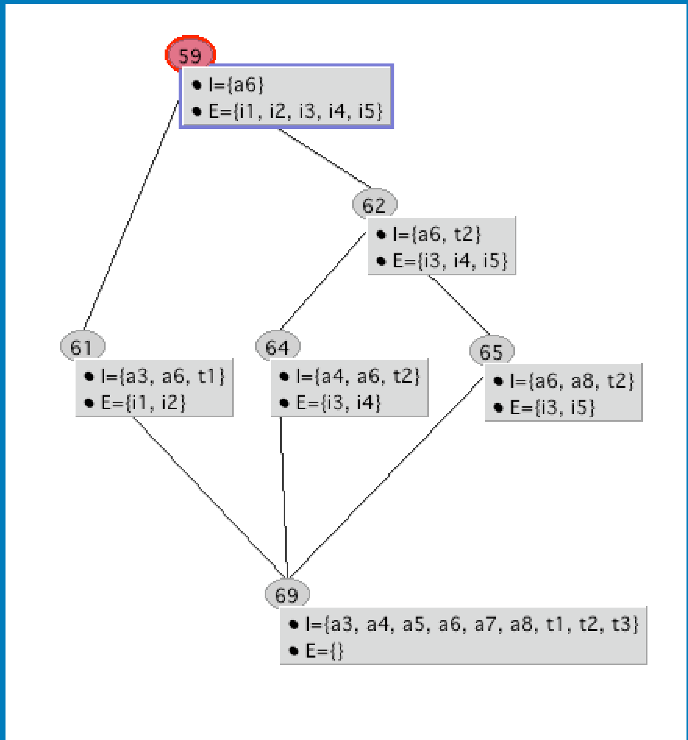
Travaux en cours

- Construction incrémentale de TG alpha
- Généralisation des treillis alpha
- Treillis de Galois - alpha

Construction incrémentale

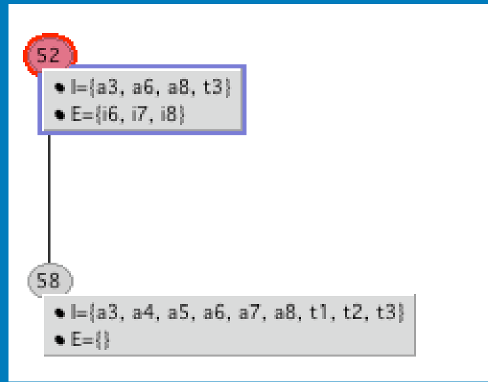


Merging Frequent lattices

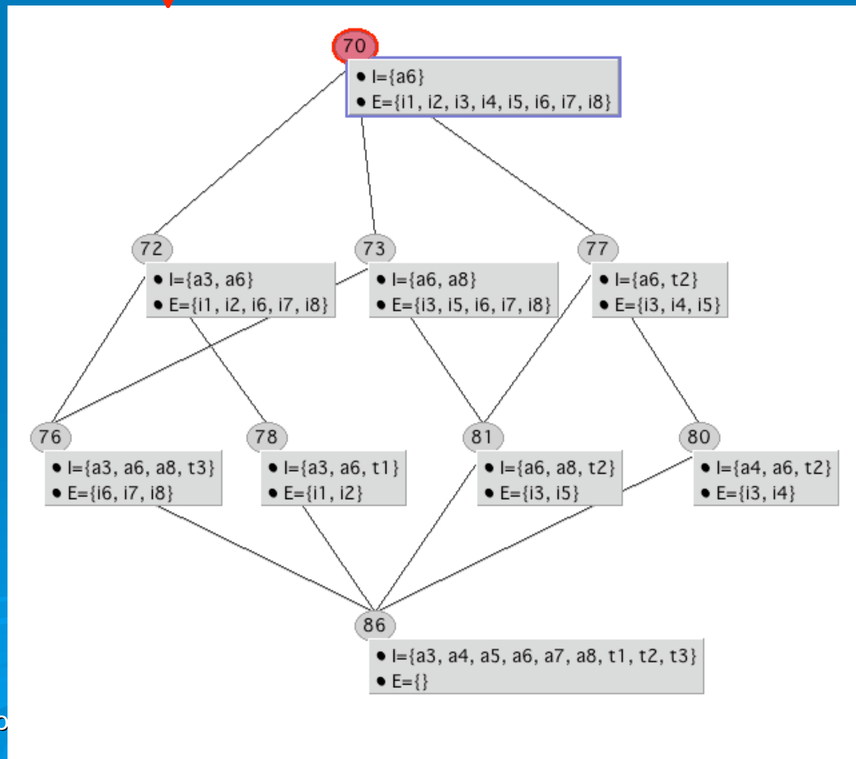


Classes 1+2
(Alpha=60)

Class 3
(minsupp=60)

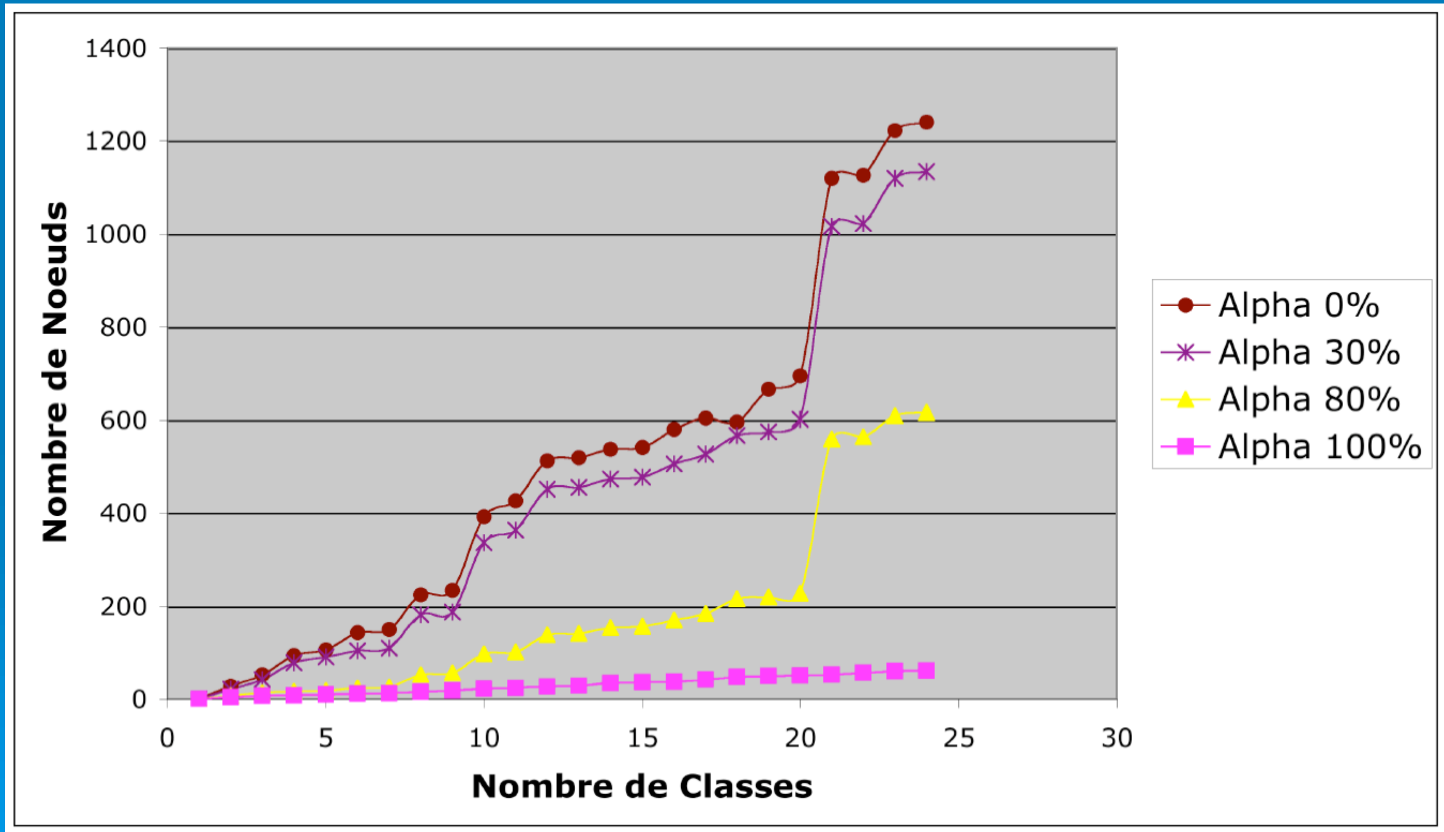


Classes 1+2+3
(Alpha=60)

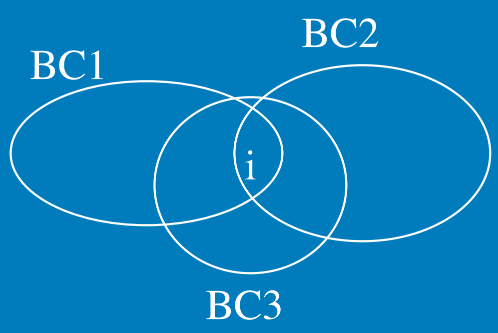


Experiment

- Merging 24 frequent lattices to obtain an Alpha lattice :



The overlaping case



- ext $_{\alpha}$(T) =

{ i $\in \mathcal{I}$ such that

i *isa* T and
there exists BCk such that:
i $\in$ BCk and BCk sat $_{\alpha}$ T }

A generalized definition of Alpha Galois lattices

Treillis de Galois - alpha

- Affaiblissement de la notion d'extension :

ext $_{-\alpha}$(A) = { i ∈ $\mathcal{I}$ telles que i *isa* A ou
BC(i) sat $_{\alpha}$ A}

Problème : si on garde la fonction int, la composition de int et ext $_{-\alpha}$ n'est pas un opérateur de fermeture

Ex : ext $_{-\alpha}$(Vole) = {titi, bip-bip} int({titi, bip-bip}) = {bec, plumes, ailes}

Treillis de Galois - alpha

- Ajout de connecteurs de type défaut exception : δ et ε
- Subsomption : δVole subsume Vole et Vole$^{\varepsilon}$
- On sature les instances avec des propriétés exceptionnelles (bip-bip est saturée avec Vole$^{\varepsilon}$)
- Int s'applique sur les instances saturées

Ext Moins-alpha: défauts et exceptions.

classe CB1 = {0}
classe CB2={1,2,3}
avec moinsalpha = 66 %

original

A	B	C	D
original	x	y	z
0	0	X	0
1	0	X	0
2	X	0	0
3	X	0	0

original | moinsalpha

A	B	C	D
moinsalpha	x	y	z
0	0	X	0
1	X !	X	0
2	X	0	0
3	X	0	0

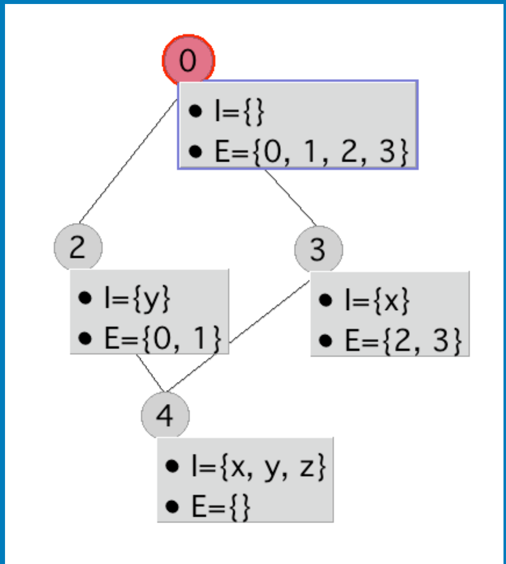
x est présent dans plus de 66 % des instances de la classe CB2 donc x est ajouté à l'instance 1.

x est absent de l'instance 1, mais présent dans plus de 66 % des instances de la classe CB2, donc x^{ε} est ajouté à l'instance 1 ainsi que δx·

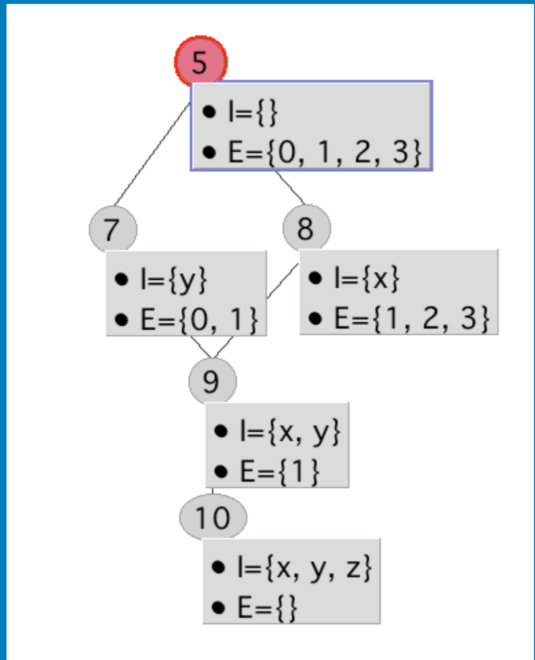
original | moinsalpha | moinsalphade

A	B	C	D	E	F	G	H	I	J
moinsalph...	x	y	z	ex	ey	ez	dx	dy	dz
0	0	X	0	0	0	0	0	X	0
1	0	X	0	X	0	0	X	X	0
2	X	0	0	0	0	0	X	0	0
3	X	0	0	0	0	0	X	0	0

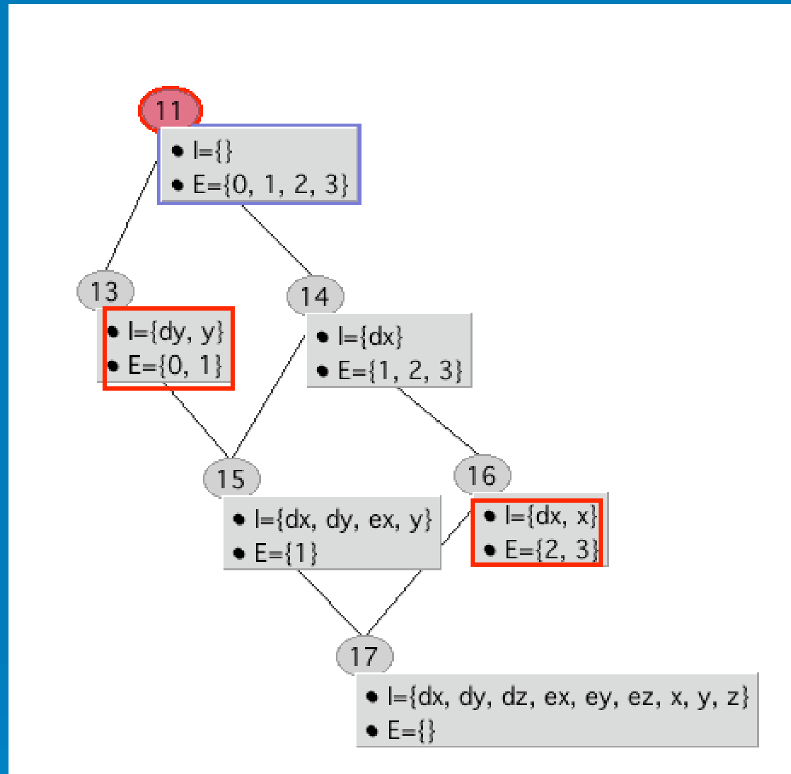
Treillis de concepts



Treillis original



Treillis de l'extension moinsAlpha approchée



Treillis original étendu par les connecteurs δ et ε

Apprentissage supervisé et TG

- Principe général : construire et comparer des TG construits avec et sans utilisation d'un étiquetage

- Notion d'extension exclusive (projection vers le top)

Conclusion

- Evaluation d'algorithmes de clustering
- Raisonnement alpha
- Liens apprentissage supervisé et non supervisé