« Treillis marseillais »

Marseille – Avril 2007

La base canonique directe

Karell Bertet

Laboratoire L3I – Université de La Rochelle - France

Plan

1. The canonical direct basis CDB

- Some definitions (implicational system, closure system)
- Identity between 5 basis (*cahiers bleux CAMS, submitted)*
- Link with Horn clauses

2. Algorithmical aspects

- Generation of a closure, of the whole family
- Incremental generation of the CDB *(CLA'O6)*
- A joint use of the two basis: the CDB and the CB *(ICFCA'O7)*

3. Some applications

- Lattice theory and datas
- Data-mining and symbolic methods
- Recognition of noisy images of symbols (*Phd Stéphanie Guillas)*

4. Conclusion

Σ: unary implicational system

•**<u>Unary implicational system Σ on S</u>** : binary relation between P(S) and S denoted **UIS** :

$$\Sigma \subseteq P(S) \times S$$

•**<u>Implication :</u>** pair of the binary relation

$$(B,x) \in \Sigma \quad \text{denoted} \quad B \to x$$

Premisse

Conclusion

•**<u>Implicational system Σ</u>**^{c} **<u>on S</u>** : binary relation on P(S), denoted **IS**:

$$\Sigma^{c} \subseteq P(S) \times P(S)$$

•To every IS, one can associate an unique UIS as follows:

$$B \to A \in \Sigma^{c} \quad \Leftrightarrow \quad \{ B \to x \ : \ x \in A \} \subseteq \Sigma$$

F_Σ: closure system

- A subset $X \subseteq S$ **verifies** the implication $B \rightarrow x \in \Sigma$ if

$$B \subseteq X \Rightarrow x \in X$$

- To every UIS Σ one can associate the **family F_Σ** of all the subsets of S verifying all the implications of Σ:

$$F_\Sigma = \{X \subseteq S : X \text{ verified } B \rightarrow x \text{ for all } B \rightarrow x \in \Sigma\}$$

- Two UIS Σ and Σ' are **equivalent** when $F_\Sigma = F_{\Sigma'}$
- F_Σ is a **Moore family** (i.e. closed under intersection, and containing S)

$\Rightarrow$ $(F_\Sigma, \subseteq)$ is a **lattice**

- F_Σ is a **closure system** $\Rightarrow$ it is associated to a **closure operator** φ_Σ

φ_Σ: closure operator

- To every closure system F_Σ on S, one can associate a **closure operator** φ_Σ defined on P(S), for $X \subseteq S$:

$$\varphi_\Sigma (X) = \text{smaller subset of } F_\Sigma \text{ containing } X = \cap \{F \in F_\Sigma : X \subseteq F\}$$

- F_Σ is the set of **fixed points** of φ_Σ:

$$F_\Sigma = \{\mathbf{F} \subseteq S : F = \varphi_\Sigma (F) \}$$

- φ_Σ is, with $X, X' \subseteq S$:
  - **idempotent:** $\varphi_\Sigma (\varphi_\Sigma (X)) = \varphi_\Sigma (X)$
  - **extensiv:** $X \subseteq \varphi_\Sigma (X)$
  - **isotone:** $X \subseteq X' \Rightarrow \varphi_\Sigma (X) \subseteq \varphi_\Sigma (X')$

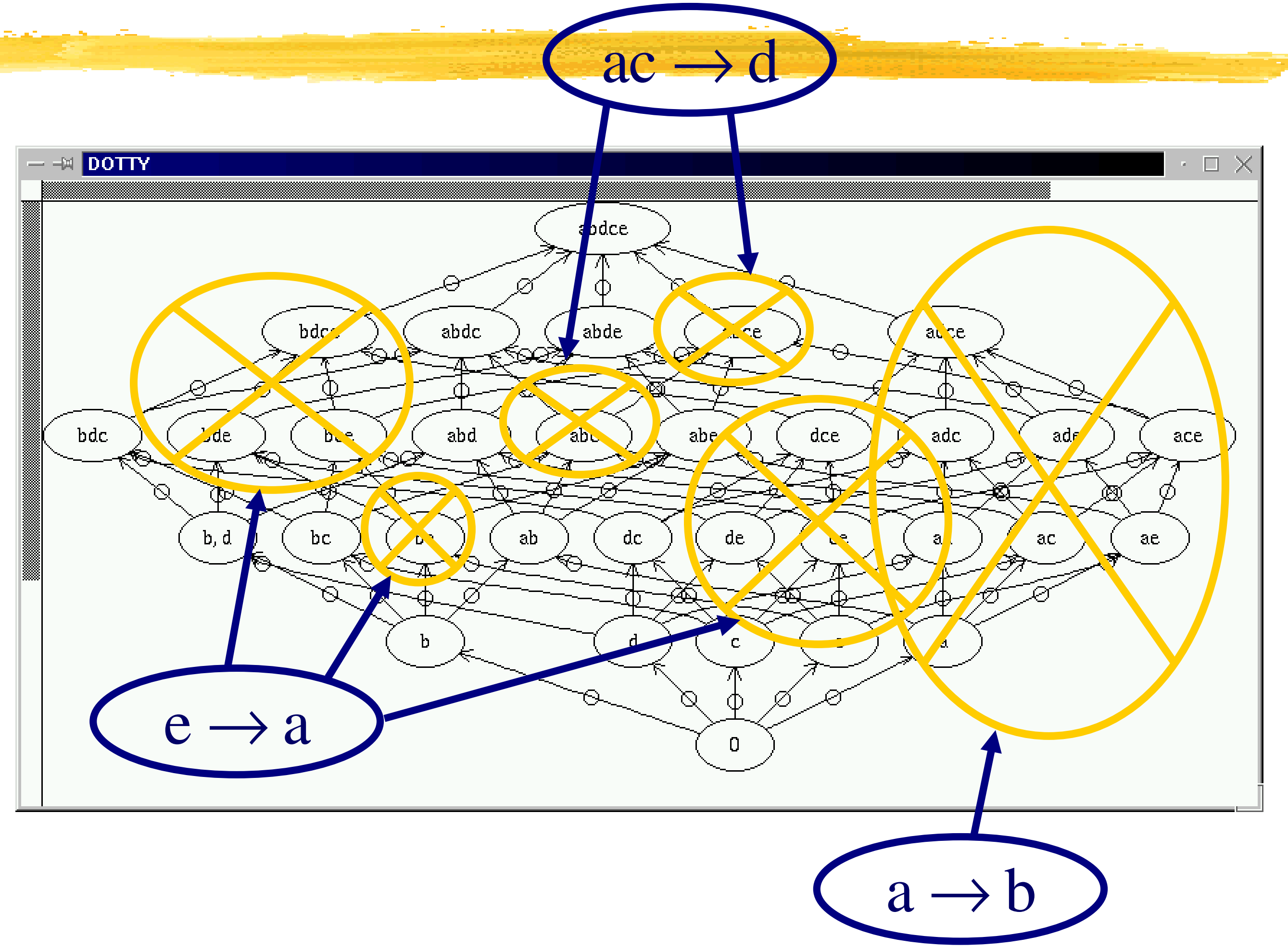
Example

$S = \{a,b,c,d,e\}$

$\Sigma = \{ \; a \rightarrow b,$

$ac \rightarrow d,$

$e \rightarrow a\}$



Example

$\varphi_\Sigma(ac) = \varphi_\Sigma(abc) = abcd$

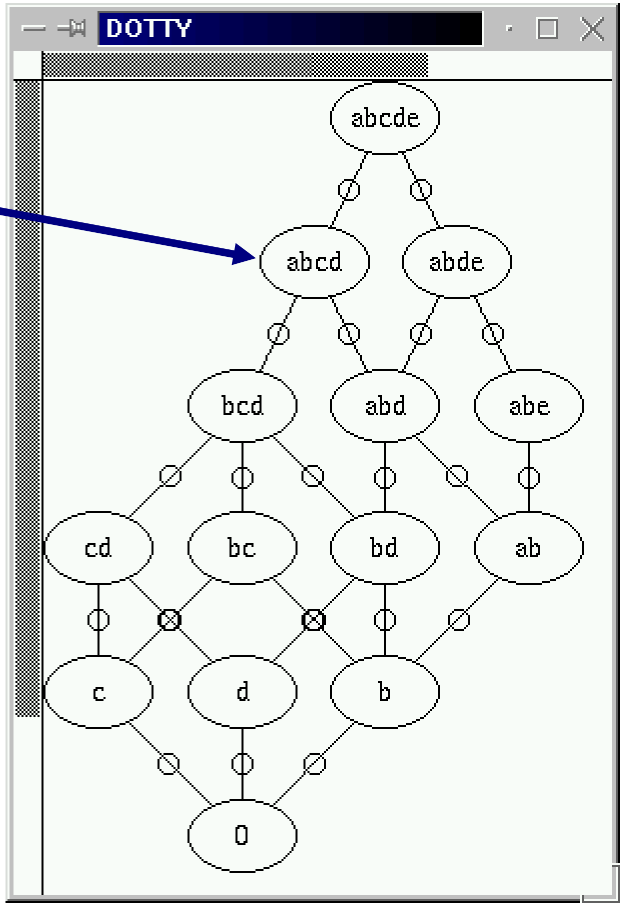
$S = \{a,b,c,d,e\}$

$\Sigma = \{\ a \rightarrow b,$

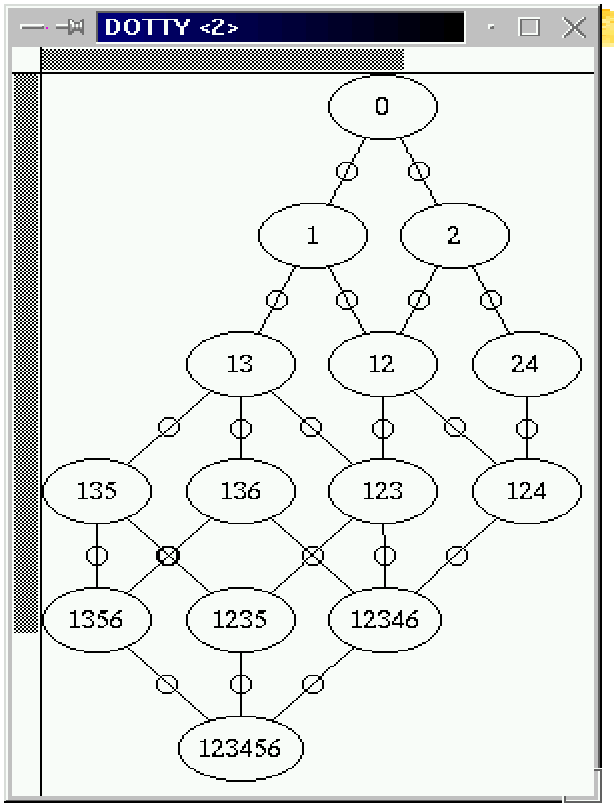
$ac \rightarrow d,$

$e \rightarrow a\}$

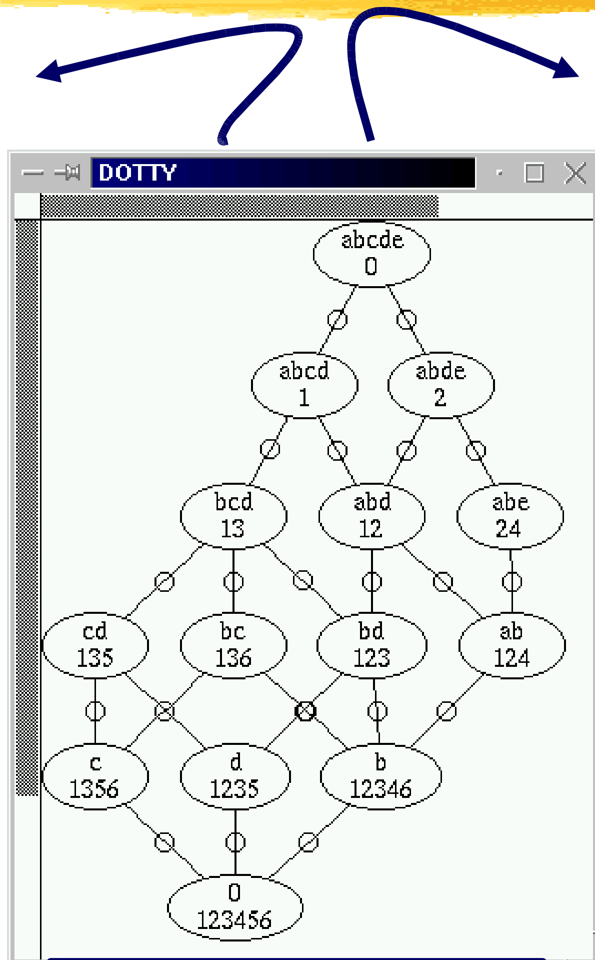
F_Σ



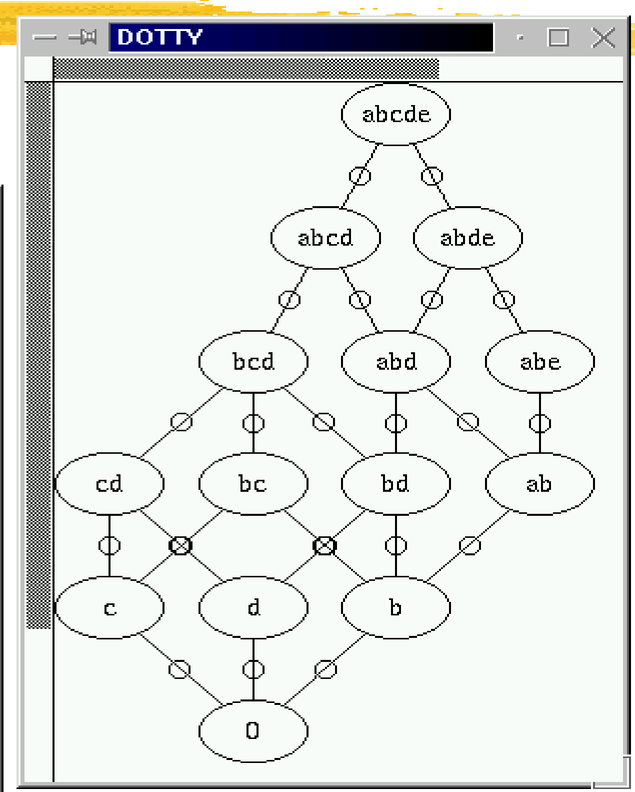
Closure systems and Galois lattice



Closure system on objects



Galois lattice



Closure system on attributes

Some properties of UIS

- An UIS Σ is **pure** if $x \notin B$ for every implication $B \rightarrow x \in \Sigma$.
- An UIS Σ is **minimal** iff, $\forall B \rightarrow x$, $\Sigma \setminus \{B \rightarrow x\}$ is not equivalent to Σ
- An UIS Σ is **minimum** iff $|\Sigma| \leq |\Sigma'|$ $\forall \Sigma'$ equivalent:
- For every UIS Σ, the closure $\varphi_\Sigma(X)$, with $X \subseteq S$, is obtained by several iterations over the implications of Σ:
  - $\varphi_\Sigma(X) = \pi(X) \cup \pi^2(X) \cup \pi^3(X) \cup \ldots..$
  - with $\pi(X) = X \cup \{ x : X \subseteq B \text{ and } B \rightarrow x \in \Sigma \}$
- An UIS Σ is **direct** iff: $\varphi_\Sigma(X) = \pi(X)$

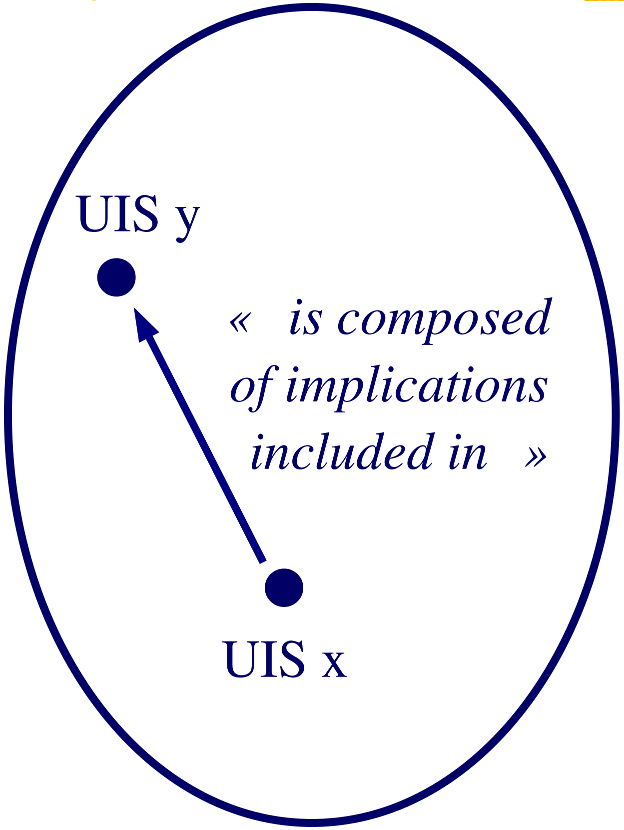
Equivalent UISs

⇒ let us consider the set of all

pure and equivalent UISs

ordered by inclusion

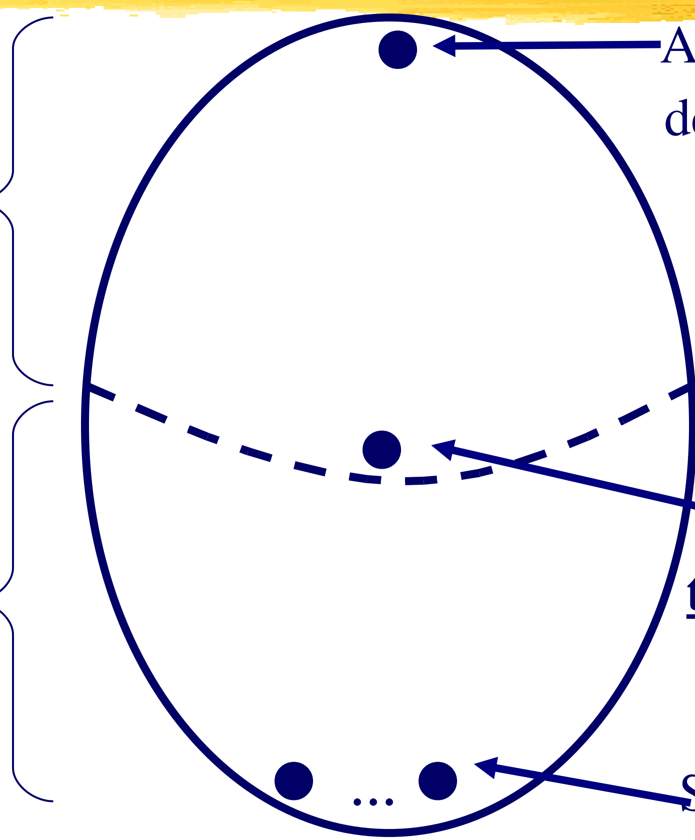
of their implications



Equivalent UISs

direct UIS

non direct UIS



An unique maximal UIS denoted **the full UIS** Σ_f

$$\Sigma_f = \{X \rightarrow \varphi_\Sigma (X)\backslash X : X \subseteq S \}$$

An unique minimal UIS between all the direct UIS, denoted **the canonical direct basis (CDB)** $\mathbf{\Sigma_{cd}}$

Several minimal UIS, but an unique minimum UIS denoted **the canonical basis (CB)** $\mathbf{\Sigma_{can}}$

Example

The canonical direct basis CDB

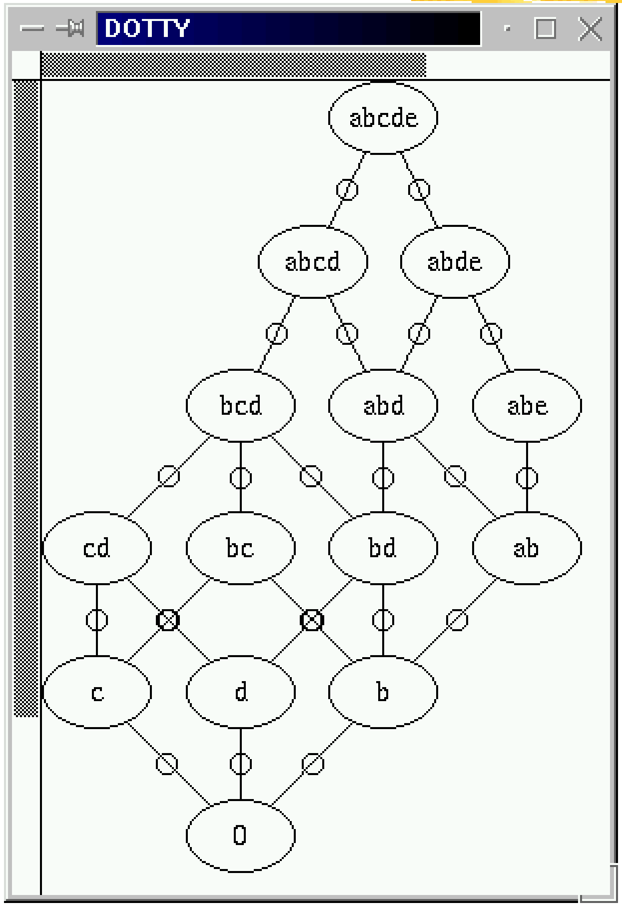
Σ_{cd} = {a → b , ac → d , e → a, **e → b , ce → d** }

•Not minimal between all the equivalent UIS:

(e → b or ce → d can be deleted)

•Direct (only one iteration to compute every closure)

•Minimal between all the direct UIS
(there exist no smaller direct UIS)



Identity between basis

The following basis are equivalent to the **canonical direct basis** Σ_{cd} (*Bertet, Monjardet, 2005*) :

- The **left minimal basis** (*Demetrovics et Hua, 1991)* also denoted the proper implications in data-mining (*Bastide et Taouil, 2002)*, or the fonctional dependencies in data-bases (*Maier, 1983)*

- The **canonical iteration free basis** *(Wild, 1994)* defined using free subsets.

- The **weak implication basis** *(Rush et Wille, 1996)* defined using minimal transversal of a family.

- The **optimal constructive basis** (*Bertet et Nebut, 2004)* defined by a generation way

The left minimal basis Σ_{lm}

Demetrovics et Hua (1991)

$$\Sigma_{lm} = \{\ B \rightarrow x \quad : \quad x \in \varphi(B) \setminus B \text{ and } B \text{ minimal}\ \}$$

$$\Sigma_{lm} = \{\ B \rightarrow x \quad : \ B \rightarrow x \in \Sigma_f \text{ and for all } Y \subset B,\ Y \rightarrow x \notin \Sigma_f\ \}$$

- **Proper implications** in data-mining, *Bastide et Taouil (2002)*
- **Functional dependencies** in data bases, *Maier (1983)*

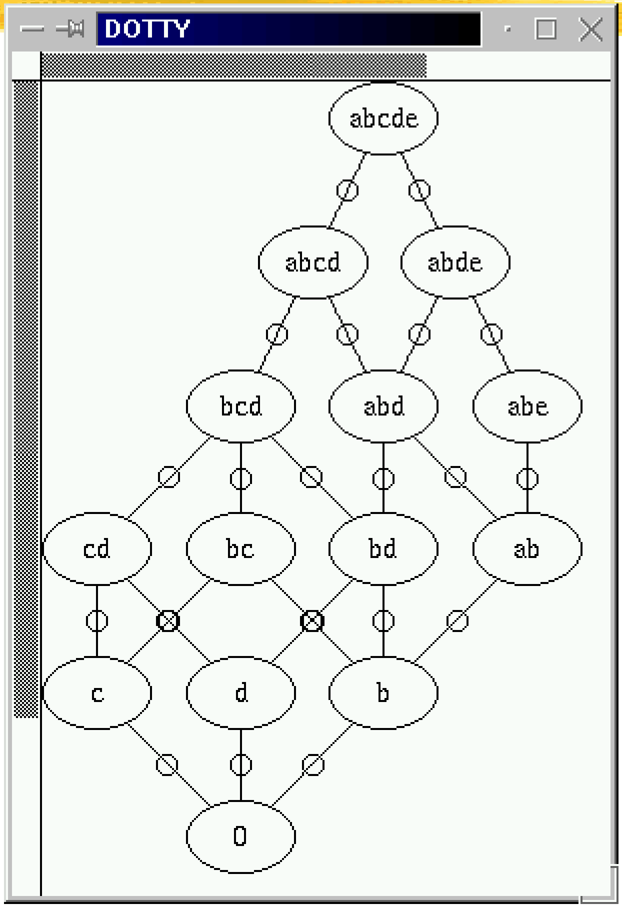
The left minimal basis Σ_{lm}

- φ_Σ (ac) = abcd

$\Rightarrow$ ac $\rightarrow$ b $\in$ Σ_f

- φ_Σ (a) = ab

$\Rightarrow$ a $\rightarrow$ b $\in$ Σ_{lm}



Σ_{lm} = {a $\rightarrow$ b , ac $\rightarrow$ d , e $\rightarrow$ a,

e $\rightarrow$ b , ce $\rightarrow$ d }

The canonical iteration free basis Σ_{cif}

Wild (1994)

Σ_{cif} = { B→ x : x∈ φ(B) \ π(B) and B is a ***free subset***}

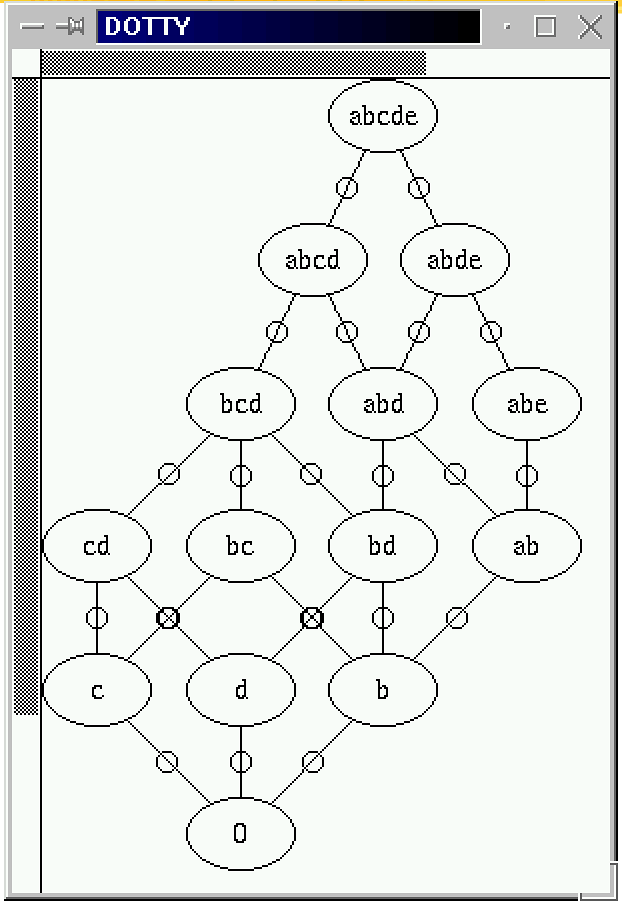
with:

- X ⊆ S ***free subset*** if, for all x ∈ X, x ∉ φ(X \ x)
- **π(X)** = X ∪ { φ(Y) : Y ⊂ X and φ(Y) ⊂ φ(X) }

The canonical iteration free basis Σ_{cif}

- ac is a free subset since $a \notin \varphi_\Sigma(c)$ and $c \notin \varphi_\Sigma(a)$
- $\varphi_\Sigma(ac) = abcd$
- $\pi(ac) = ac \cup \varphi_\Sigma(a) \cup \varphi_\Sigma(c) = ac \cup ab \cup c = abc$

  $\Rightarrow ac \rightarrow d \in \Sigma_{cif}$



$\Sigma_{cif} = \{a \rightarrow b,\ ac \rightarrow d,\ e \rightarrow a,$

$\mathbf{e \rightarrow b,\ ce \rightarrow d}\ \}$

The weak implication basis Σ_{weak}

Rush and Wille (1996)

Σ_{weak} = { B→ x : B ***blockade*** for x }

where a ***blockade*** for x is a ***minimal transversal*** of the family

F(x) = { S \ (F+x) : F ***copoint*** of x }

A ***copoint*** of x is a maximal closed set of F that doesn't contains x

Equivalently a ***copoint*** of x is a meet-irreducible m of F such that $\varphi_\Sigma(x) \uparrow$ m

The weak implication basis Σ_{weak}

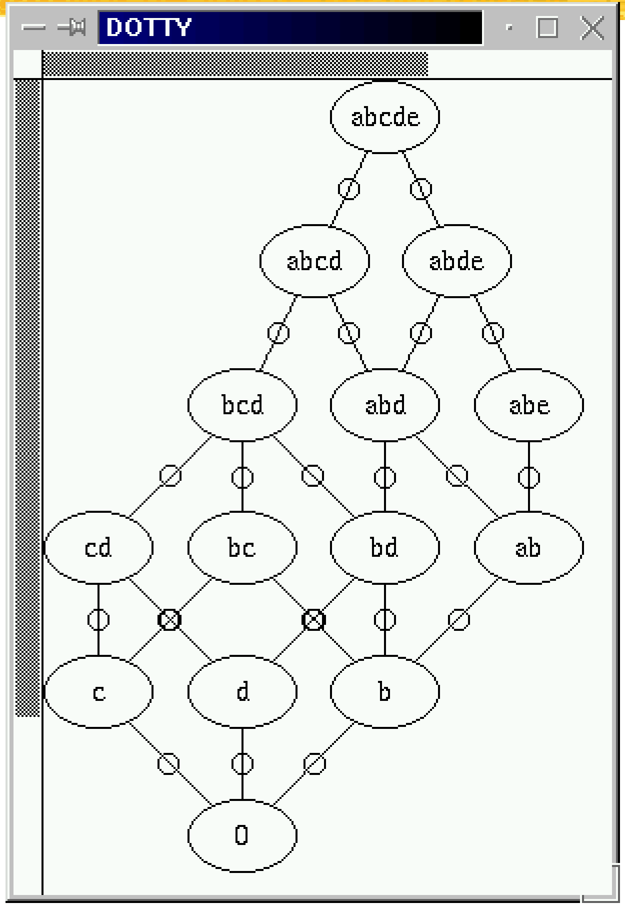
- ***copoints*** of d: $F_d = \{ bc, abe \}$
- Minimal transversal of F_d : $\{ac, ce\}$

$\Rightarrow ac \rightarrow d \in \Sigma_{weak}$

$\Rightarrow ce \rightarrow d \in \Sigma_{weak}$

$\Sigma_{weak} = \{a \rightarrow b \text{ , } ac \rightarrow d \text{ , } e \rightarrow a,$

$\mathbf{e \rightarrow b \text{ , } ce \rightarrow d} \}$



The basis associated to the dependance relation Σ_{dep}

Monjardet and Caspard (1990, 1997), Bertet (2004)

$$\Sigma_{dep} = \{ B+y \rightarrow x \; : \; x \, \delta_B \, y \text{ and } B \text{ is minimal } \}$$

where δ_B is the ***dependance relation*** of F valuated by subsets of S:

$$x \, \delta_B \, y \quad \text{iff} \quad x,y \notin \varphi(B) \quad \text{and} \quad x \in \varphi(B+y)$$

The dependance relation can also be defined using ***arrows relations***

The basis associated to the dependance relation Σ_{dep}

- $a,e \notin \varphi(\phi)$ and $a \in \varphi(e)$

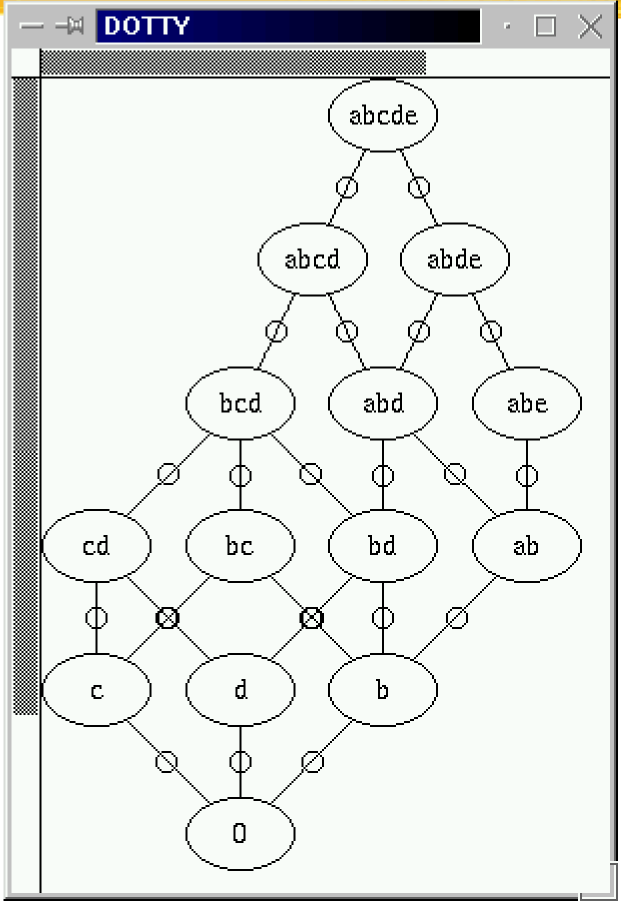
$\Rightarrow a\ \delta_{\phi}\ e \ \Rightarrow\ e \rightarrow a \in \Sigma_{dep}$

- $c,d \notin \varphi(a)$ and $d \in \varphi(ac)$

$\Rightarrow d\ \delta_{a}\ c \ \Rightarrow\ ac \rightarrow d \in \Sigma_{dep}$

$\Sigma_{dep} = \{a \rightarrow b\ ,\ ac \rightarrow d\ ,\ e \rightarrow a,$

$\mathbf{e \rightarrow b\ ,\ ce \rightarrow d}\ \}$



Horn clauses

Bijection:

Fonctions booléennes sur P(S) ⇔ Famille sur S

Exemple:

f = abc'd'+ab'cd'+a'b'c'd ⇔ F = {ab, ac, d}

Simplication
(recherche des implicants premiers)
Problème NP

En particulier:

Fonctions booléennes **de Horn** sur P(S) ⇔ Famille **de Moore** sur S

Link with Horn clauses

Une **fonction de Horn** est une formule propositionelle telle que:

- les disjonctions de la FND admettent une seule variable complémentée
- les conjonctions de la FNC admettent une seule variable non complémentée

⇒on parle d' ***implicants premiers***

Bijection

(Bertet, Monjardet- 2005)

Implicants premiers ⇔ Implications de Σ_{cd}

Exemple:

ab'd ⇔ a'+b+d' ⇔ ad → b

Plan

1. **The canonical direct basis CDB**
   - Some definitions (implicational system, closure system)
   - Identity between 5 basis (*cahiers bleux CAMS, submitted)*
   - Link with Horn clauses
2. **Algorithmical aspects**
   - Generation of a closure, of the whole family, of the basis
   - Incremental generation of the CDB *(CLA'O6)*
   - A joint use of the two basis: the CDB and the CB *(ICFCA'O7)*
3. **Some applications**
   - Lattice theory and datas
   - Data-mining and symbolic methods
   - Recognition of noisy images of symbols (*Phd Stéphanie Guillas)*
4. **Conclusion**

Algorithmical aspects

Four main generation problems:

1. **How to generate a closure φ_{Σ} (X) ?**
2. **How to generate the family F ?**
3. **How to generate the canonical basis Σ_{can} ?**
4. **How to generate the canonical direct basis Σ_{cd} ?**

Problems 2, 3 and 4: Generation are « *output sensitive* » since

- Σ_{cd} : can be exponential in $|\Sigma|$,with Σ equivalent, or in $|S|$
- F : can be exponential in $|\Sigma_{cd}|$ or in $|S|$

$\Rightarrow$ complexity is expressed for :

- the generation of ***one closed set*** of F (P-complete)
- the generation of ***one implication*** of Σ_{can} or Σ_{cd} (NP, open problem)

Generation of a closed set φ_Σ (X)

How to generate a closed set φ_Σ (X), with X $\subseteq$ S ?

•Using any UIS or the **canonical basis** Σ_{can}:

when not direct, several iterations over the implications are performed

Linclosure : **O(|Σ_{can}| |S|²)**
(Mannila, Raïhä, 1992)

•Using a direct UIS or the **canonical direct basis** Σ_{cd}:

when direct, only one iteration over the implications is needed

O(|Σ_{cd}| |S|)
(Bertet, Nebut, 2004)

Generation of a closed set $\varphi_\Sigma(X)$

Generation of φ_Σ (ce) by 2 iterations using

the canonical basis (minimal and minimum, but not direct)

$$\varphi_\Sigma (ce) \quad = \quad ce \cup \pi(ce) \cup \pi^2(ce) = ce \cup a \cup b$$

$$\Sigma_{can} = \{a \rightarrow b \text{ , } abc \rightarrow d \text{ , } e \rightarrow a\}$$

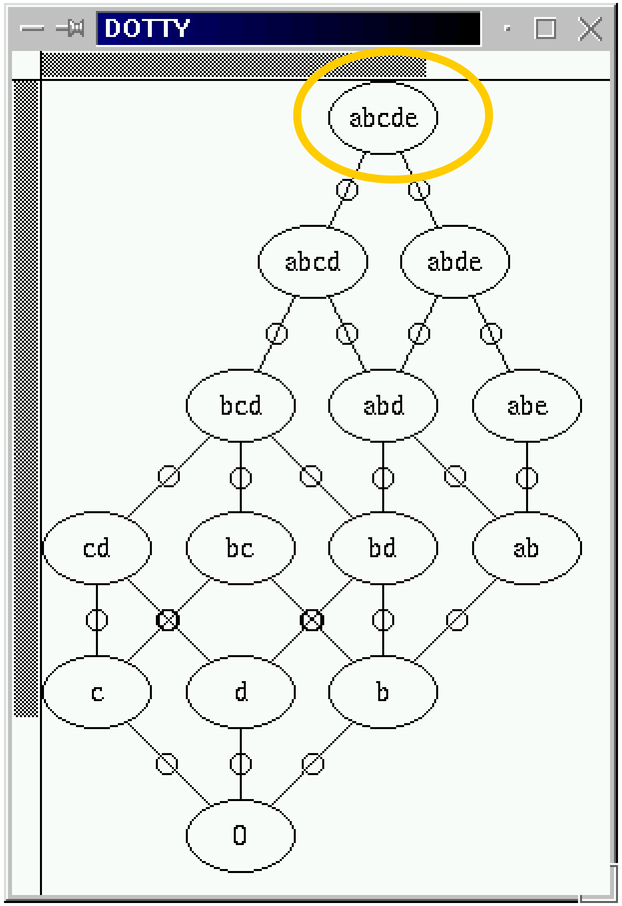
Generation of φ_Σ (ce) with <u>1 iteration</u> using

the canonical direct basis (direct, not minimal,

but minimal between all the direct UIS):

$$\varphi_\Sigma (ce) \quad = \quad ce \cup \pi(ce) \; = \; ce \cup bd$$

$$\Sigma_{cd} = \{a \rightarrow b \text{ , } ac \rightarrow d \text{ , } e \rightarrow a, \; e \rightarrow b \text{ , } ce \rightarrow d \}$$



Generation of the family F

How to generate the whole family F_Σ (thus a lattice) ?

- from any UIS Σ :

$$F_\Sigma = \varphi_\Sigma(\phi) \cup \{ \varphi_\Sigma(x) : x \in S \} \cup \{\varphi_\Sigma(F \cup F') : F, F' \in F_\Sigma \}$$

per closed set of $\mathbf{F_\Sigma}$: $\mathbf{O(|S|^2\, c(\varphi))}$ Next Closure (*Ganter, 1984*)

with $c(\varphi)$, coast of one closed set generation

- from the canonical basis : $\mathbf{O(|\Sigma_{can}|\ |S|^4)}$
- from the canonical direct basis: $\mathbf{O(|\Sigma_{cd}|\ |S|^3)}$

Generation of the canonical basis

How to formally define the canonical basis ?

• The canonical basis is defined as an IS (not an UIS)

• Thus one can associate an unique **UIS** to the canonical basis

• The definition of the canonical basis definition based on **pseudo-closed sets** *(Guigues Duquenne 1986)*:

$$\boldsymbol{\Sigma}_{\mathbf{can}} = \{\ P \rightarrow \varphi_{\Sigma}(P) \setminus P \ \text{ with } \ P \subseteq S \ \textbf{pseudo-closed set}\ \}$$

Generation of the canonical basis

How to generate the canonical basis ?

- From a context (*Ganter, 1984*): Next Closure algorithm
- From an equivalent UIS **Σ**: first minimize **Σ** before to replace premisses by pseudo-closed sets

Exponential generation per implication

(open problem)

Incremental generation of the canonical basis ?

- Attribute-incremental generation from a context (*Obiedkov, Duquenne, 2003*)

Exponential generation per implication, competitive in practice

Generation of the canonical direct basis

How to generate the canonical direct basis ?

- From an equivalent UIS Σ : *(Wild, 1995) (Bertet and Nebut, 2004)*

Generation of an **intermediate direct UIS** whose size is, in the worst case, exponential in S. Simple to implement.

Incremental generation of the canonical direct basis ?

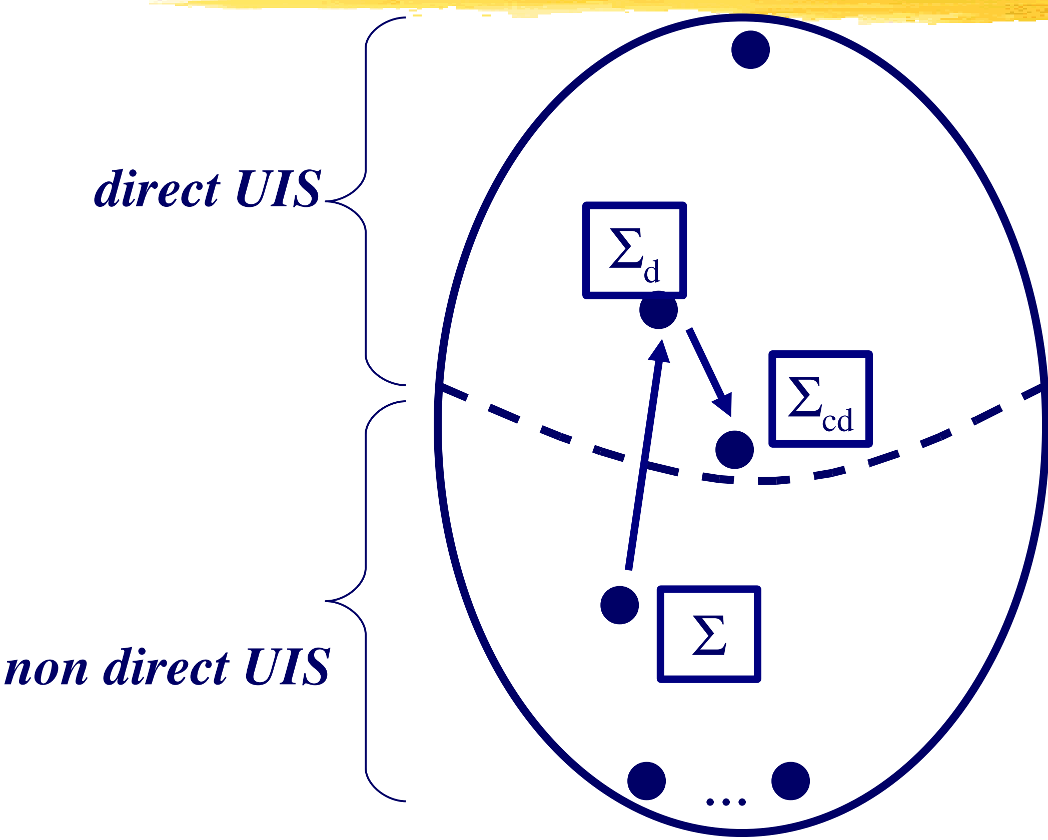
- From an equivalent UIS Σ : *(Bertet 2006)*

The size of the **intermediate direct UIS** that has to be generated is reduced. Very competitive in practice. Simple to implement.

Exponential generation per implication

(open problem)

Generation of the canonical direct basis



Generation of Σ_{cd} from an equivalent UIS Σ:

$O(|S||\Sigma_d|^2)$

(Wild, 1995)

(Bertet Nebut, 2004)

- Σ_d is an **intermediate direct UIS** generated before to be minimized.
- size of Σ_d is, in the worst case exponential in S.

  $\Rightarrow$ **exponential complexity per implication**

Generation of the canonical direct basis

(Wild, 1995) (Bertet and Nebut, 2004)

Generation of an **intermediate direct UIS** whose size is, in the worst case exponential in S.

<u>From any equivalent UIS Σ :</u>

1) first, **recursively** apply the **make-direct treatment** (to obtain the equivalent **intermediate direct UIS Σ_d**)

« *for all* $B \to x$ *and* $C+x \to d$, <u>*add* $B \cup C \to d$</u> »

2) then apply the **make-minimal treatment** (to obtain the **canonical direct basis Σ_{cd}**)

« *for all* $A \to x$ *and* $C \to x$, <u>*delete* $A \to b$ *when* $C \subset A$</u> »

A joint use of the two canonical basis

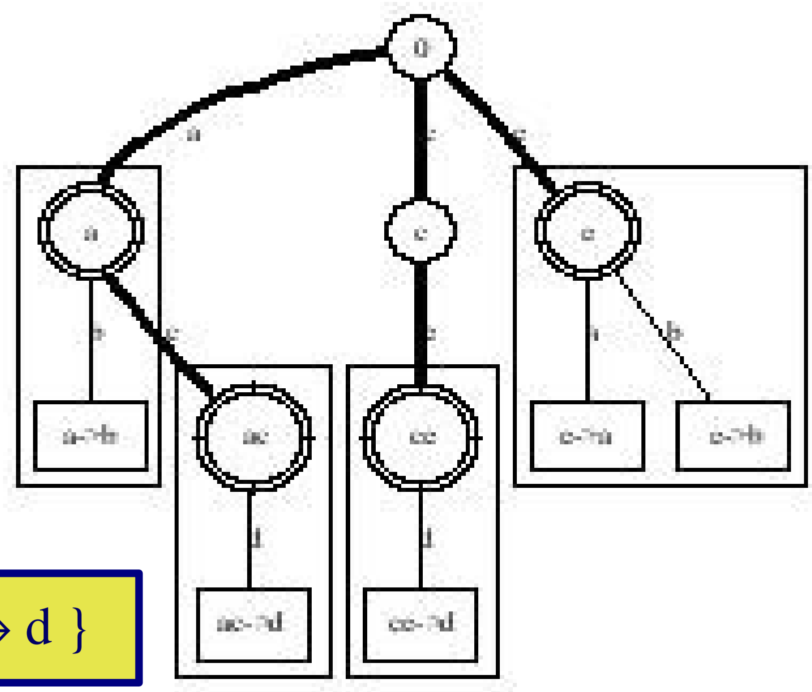
(*Bertet, Guillas, Ogier - ICFCA'07*) **Proposition of the joint use of:**

the **canonical direct basis** (for algorithmical aspects since direct)

and the **canonical basis** (minimal description, without redundancy)

- Definition of a **two-level lexicographic tree** as a data-structure to efficiently handle the two basis

- Implementation of a **java class *Rule*** to handle UIS and their basis



$$\Sigma_{cd} = \{a \rightarrow b \,,\ ac \rightarrow d \,,\ e \rightarrow a,\ e \rightarrow b \,,\ ce \rightarrow d\}$$

A joint use of the two canonical basis

A joint use of the two basis	The canonical basis	The canonical direct basis
Description	No redondancy	With redondancy (since direct)
Number of implications	Minimal	Minimal between direct UIS
Algorithmical use	Several iterations to compute one closed set	One iteration to compute one closed set
Generation	Exponential per implication	Exponential per implication
Incremental generation	Addition of a new attribute	Addition of a new implication

Incremental generation of the canonical direct basis

The incremental generation algorithm consists in limiting the size of the **intermediate direct** UIS:

- that have to be **recursively** generated by the **make-direct treatment**
- before to be minimizing by the **make-minimal treatment.**

Σ_{cd} is obtained from an UIS $\Sigma = \{ Bi \rightarrow xi : i \leq n\}$ by successively compute the canonical direct basis

$$\mathbf{\Sigma_i = (\Sigma_{i-1} \cup \{Bi \rightarrow xi\})_{cd}}$$

thus $\Sigma_1 = \{B1 \rightarrow x1 \}$

and $\Sigma_n = \Sigma_{cd}$

Generation of Σ_i from Σ_{i-1}:
$\mathbf{O(|S|.|\Sigma_i|^{(|Bi|+1)})}$

$\Rightarrow$

Incremental generation of Σ_{cd} from Σ:
$\mathbf{O(|S|.2^{(|B0|*|B1|* \dots *|Bn|)})}$

Incremental generation of Σ_{cd}

How to generate Σ_i from Σ_{i-1} ?

1. first, apply a **restriction** of the recursive **make-direct treatment** (**main Theorem)**
2. then apply the **make-minimal treatment**

Main Theorem: the ***make-direct treatment*** has to be ***non recursively*** applied to subsets of implications of Σ_i containing:

- the implication $Bi \rightarrow xi$ (since Σ_i is a **canonical direct basis**)
- and at most |Bi| implications of Σ_i

⇒ Using other subsets of implications, non minimal implication will be generated.

⇒ The size of the intermediate direct UIS that have to be generated is limited.

Experimentation 1

UIS are randomly generated with |S|=7

Number of **implications**	5	10	15	20	25
Concepts number	48	34	13	6	5
Size of the **canonical basis**	5	6	10	9	5
Size of the **canonical direct basis**	11	9	25	26	7
Generation of the **canonical direct basis**					
... by the **global algorithm**	37	10	214	257	26
... by the **incremental algorithm**	2	0	14	4	3

(number of implications of the intermediate direct UIS)

Experimentation 2

UIS are randomly generated with 15 implications

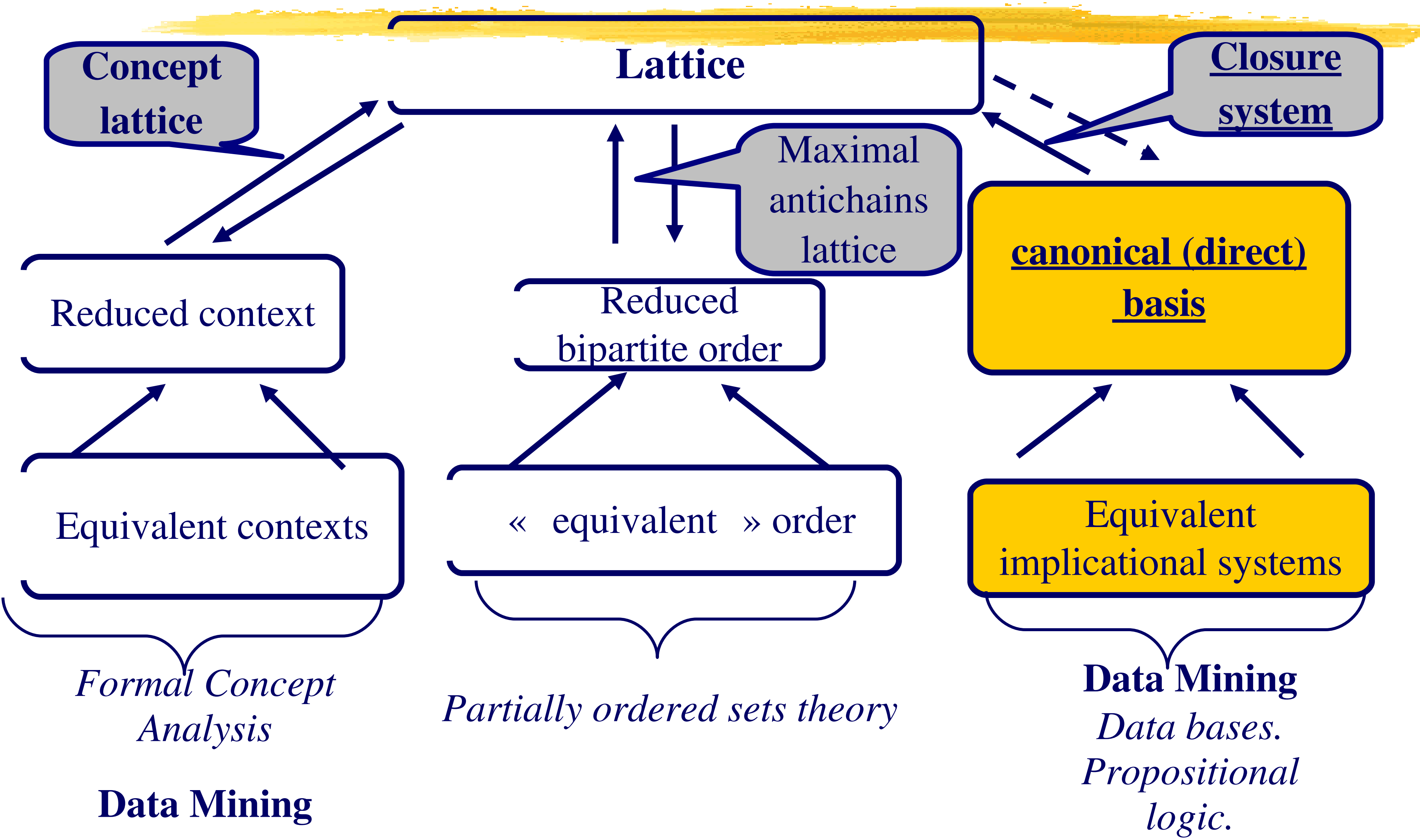
Size of the **set of elements** S	5	6	7	8	9
Conceptsnumber	5	10	14	14	89
Size of the **canonical basis**	4	5	11	10	12
Size of the **canonical direct basis**	6	11	24	32	54
Generation of the **canonical direct basis**					
... by the **global algorithm**	5	27	225	653	698
... by the**incremental algorithm**	1	0	17	9	18

(number of implications of the intermediate direct UIS)

Plan

1. **The canonical direct basis CDB**
   - Some definitions (implicational system, closure system)
   - Identity between 5 basis (*cahiers bleux CAMS, submitted)*
   - Link with Horn clauses
2. **Algorithmical aspects**
   - Generation of a closure, of the whole family, of the basis
   - Incremental generation of the CDB (*CLA'O6)*
   - A joint use of the two basis: the CDB and the CB *(ICFCA'O7)*
3. **Some applications**
   - Lattice theory and datas
   - Data-mining and symbolic methods
   - Recognition of noisy images of symbols (*Phd Stéphanie Guillas)*
4. **Conclusion**

Lattice theory



Data mining

« Est-il possible d'extraire quelque chose d'intéressant des grandes quantité de données existant actuellement ? Et comment ? »

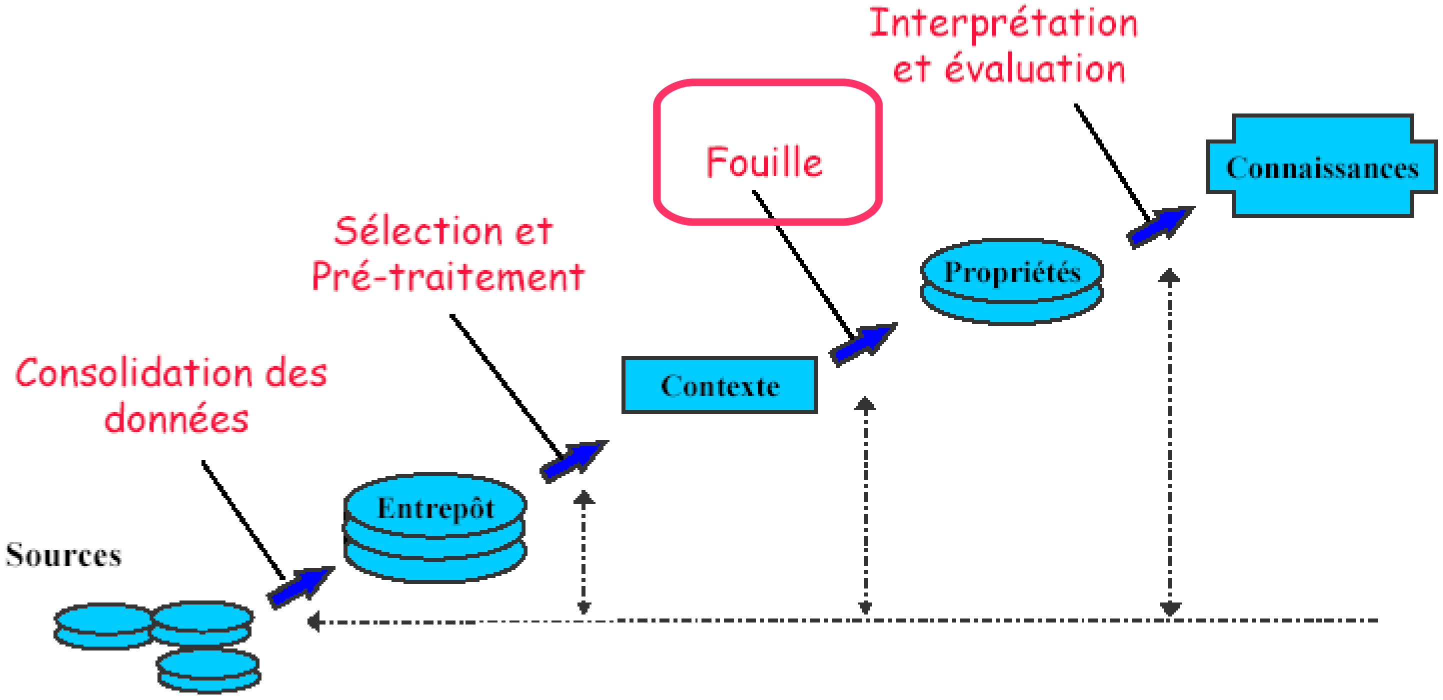
ECD (ECBD): Extraction de connaissances à partir de données (entrepots de données)

KDD: Knowledge Discovery in big Databases

Processus d'ECB: « extraire dans des *grands volumes* de données des éléments de *connaissances* non triviaux et nouveaux pouvant avoir un *sens* et un *intérêt* pour être réutilisés »

Fouille de données (data-mining): un traitement du processus d'extraction de connaissances

Processus d'extraction des connaissances



Binary datas

C/P	chips	moutarde	saucisse	boisson douce	bière
C1	x				x
C2	x	x	x	x	x
C3	x		x		
C4			x		x
C5		x	x	x	x
C6	x	x	x		x
C7	x		x	x	x
C8	x	x	x		
C9	x			x	
C1		x	x		x

Attributes, features, descriptors, ...

Objects, persons, ...

Binary relation or contexte

Data mining

Objectives of data mining:

- **Classification:** to associate a class to an object depending on its attributes. Classification needs two stages:
  - ***Learning stage*** from an inital set of classified objects
  - ***Classification stage*** of objects
- **Segmentation:** to form homogeneous groups of objects depending on their attributes.

Two kinds of models of data-mining:

- **Numerical models** for numerical datas

statistical model, Markov model, bayes model, neu...

- **Symbolic models** for binary datas (context)

association rules, Galois lattice, decision tree, ...

The most usual **symbolic models** are linked with **lattice theory**

Association rules

- X={bière, saucisse, moutarde}: *item* of ***support*** 0.4
- X'={bière, saucisse}: *item* of ***support*** 0.6
- {Bière, saucisse} → {moutarde}: ***association rule*** with ***confidence*** 0.66

« if a person buy*saucisse* and *bière*, then it will by *moutarde* with a probability of *0.66* »

C/P	chips	moutarde	saucisse	boisson douce	bière
C1	x				x
C2	x	x	x	x	x
C3	x		x		
C4			x		x
C5		x	x	x	x
C6	x	x	x		x
C7	x		x	x	x
C8	x	x	x		
C9	x			x	
C1		x	x		x

Association rules

- **Association rules:** two items $A \rightarrow B$
- **Support of a rule:** support $(A \cup B)$
- **Confidence of the rule:** support $(A \cup B)$ / support (A)
- **Valid association rule:** association rule with a confidence greather than a ***minimal confidence***
- **Exact assocation rule:** association rule with 1 as confidence

Exact association rules are implications.
The two basis are used to generate all valid association rules.
The canonical direct basis is denoted as

- **Proper implications,** *Bastide et Taouil (2002)*
- **Functional dependencies,** *Maier (1983)*

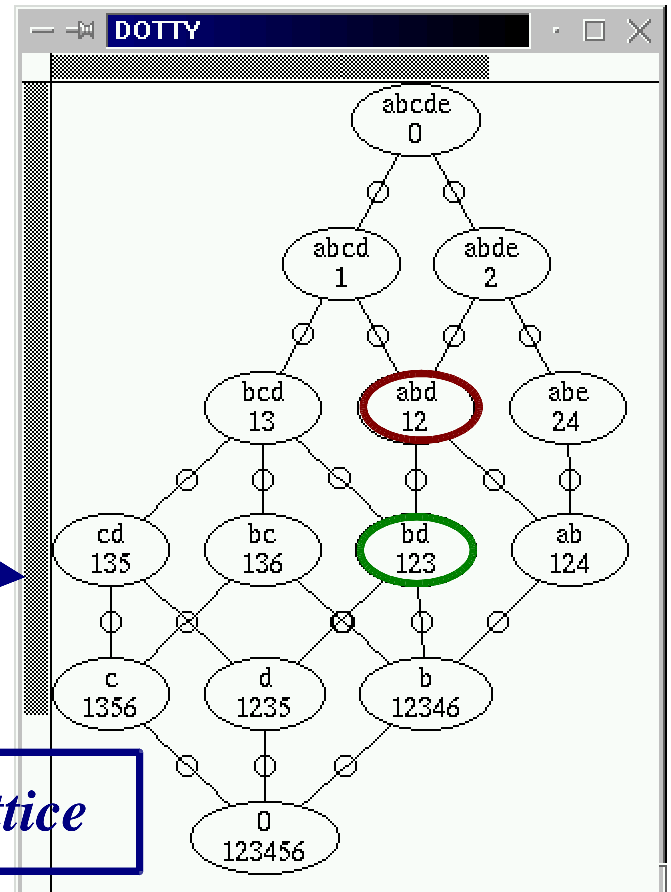
Galois / concept lattice

Example of ***concepts***:(abd,12), (bd,123),

Relation on concepts: (abd,12) ≥ (bd,123)

Context

	1	2	3	4	5	6
a	X	X		X		
b	X	X	X	X		X
c	X		X		X	X
d	X	X	X		X	
e		X		X		



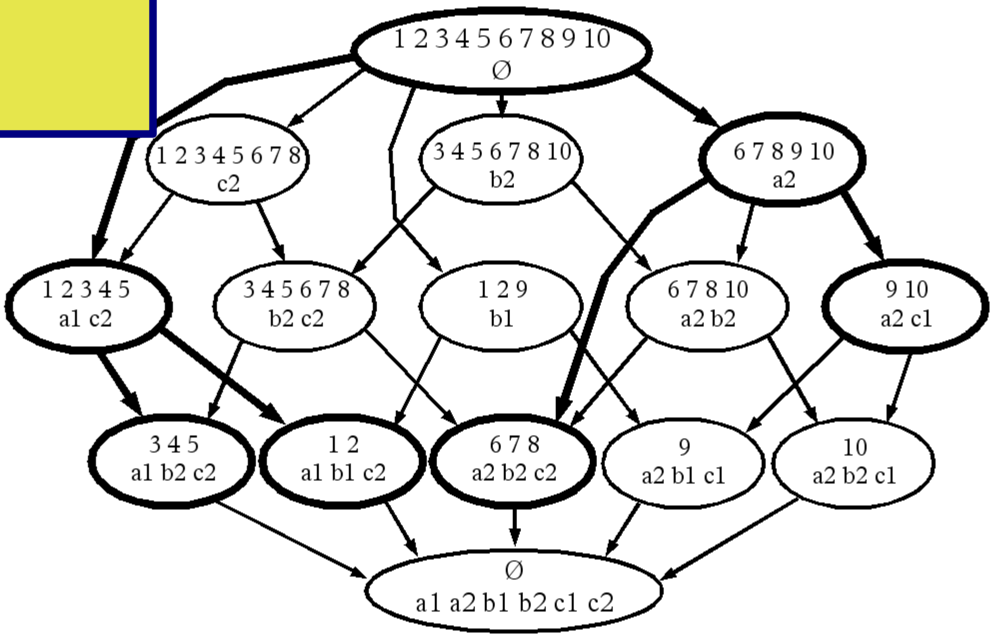
Galois lattice

Lattice and decision tree

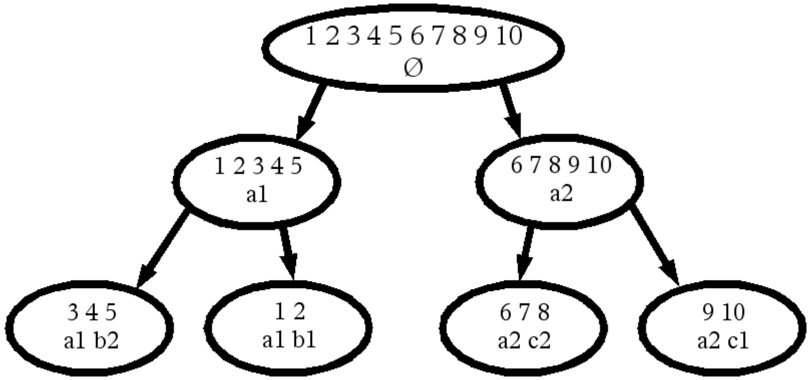
Property: (*Guillas, 2005*)

« Every decision tree issued from a context is included in the concept lattice associated to this context »

Concept lattice



Decision tree



- Smaller size
- Fast classification

⇒ approprite for exact datas

- Biger size
- Several ways of classification

⇒ appropriate for noised datas

Recognition of noised symbols

Classification with a lattice (*Guillas*, 2005):

Learning stage:

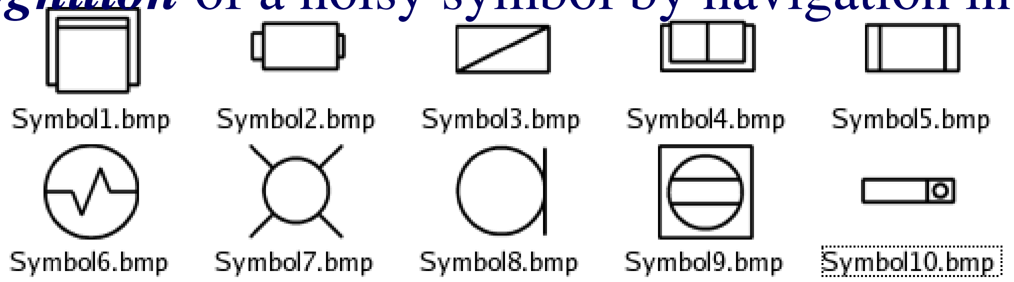
1- ***Extraction*** of a signature from images of symbols

2- ***Discretization*** of signatures according to their class

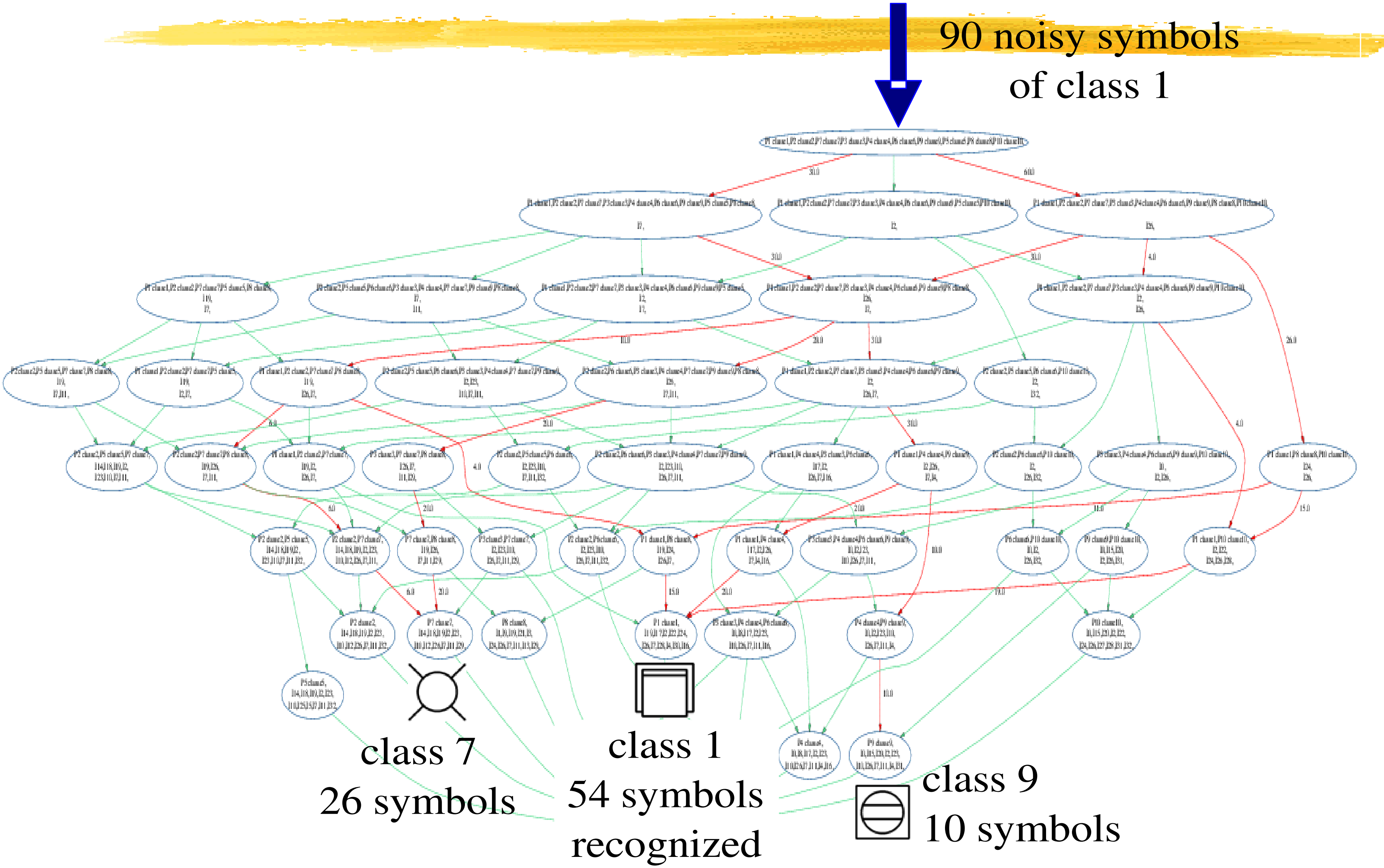
3- Generation of the ***concept lattice*** from the discretized data

Classification stage

4- ***Recognition*** of a noisy symbol by navigation into the lattice



Experimentation



Experimentation

Size of the set **S of elements**	**7**	**8**	**8**	**6**	**7**
Conceptsnumber	**20**	**42**	**24**	**25**	**23**
Recognition rate	**98,6**	**99,2**	**99,2**	**98,9**	**99,2**
Size of the **canonical basis**	**33**	**62**	**32**	**31**	**32**
Size of the **canonical direct basis**	**280**	**779**	**724**	**103**	**293**
Generation of the **canonical direct basis**					
... by the **global algorithm**	**30**	**112**	**46**	**32**	**39**
... by the**incremental algorithm**	**17**	**25**	**21**	**14**	**17**