

# Concept-based Learning Models

Sergei O. Kuznetsov

Department of Applied Mathematics and Information Science  
Higher School of Economics, Moscow;  
ABYY Chair of Pattern Recognition and Text Processing  
Moscow Institute for Physics and Technology

# Outline

1. **Lattices in Machine Learning**
2. Learning implications and association rules
3. JSM-hypotheses
4. Decision trees
5. Version spaces
6. Learning with Pattern Structures
7. Conclusions

# Lattices

Let  $(M, \leq)$  be an ordered set and  $A \subseteq M$ . A **lower bound** of  $A$  is an element  $s$  of  $M$  with  $s \leq a$  for all  $a \in A$ . Dually for a **upper bound** of  $A$ .

If there is a largest element in the set of all lower bounds of  $A$ , it is called the **infimum** of  $A$ . Dually for **supremum** of  $A$ .

An ordered set  $(L, \leq)$  is a **lattice** if for any two elements  $x, y \in L$  supremum  $x \vee y$  and infimum  $x \wedge y$  always exist.

# Lattices in machine learning. Antiunification

**Antiunification**, in the finite term case, was introduced by G. Plotkin and J. C. Reynolds.

The antiunification algorithm was studied in

J. C. Reynolds, Transformational systems and the algebraic structure of atomic formulas, *Machine Intelligence*, vol. 5, pp. 135-151, Edinburgh University Press, 1970.

as the least upper bound operation in a lattice of terms.

## Example:

If  $A = P(a, x, f(x))$  and  $B = P(y, f(b), f(f(b)))$ , then  $\wedge(A, B) = P(z_1, z_2, f(z_2))$ .

Antiunification was used by Plotkin

G.D. Plotkin, A Note on inductive generalization, *Machine Intelligence*, vol. 5, pp. 153-163, Edinburgh University Press, 1970.

as a method of generalization and later this work was extended to form a theory of inductive generalization and hypothesis formation.

# Outline

1. Lattices in Machine Learning
- 2. Learning implications and association rules**
3. JSM-hypotheses
4. Decision trees
5. Version spaces
6. Learning with Pattern Structures
7. Conclusions

# Formal Concept Analysis

[Wille 1982, Ganter, Wille 1996]

- $M$ , a set of **attributes**
- $G$ , a set of **objects**
- relation  $I \subseteq G \times M$  such that  $(g, m) \in I$  if and only if object  $g$  has the attribute  $m$ .
- $\mathbb{K} := (G, M, I)$  is a **formal context**.

**Derivation operators:**

$$A' \stackrel{\text{def}}{=} \{m \in M \mid gIm \text{ for all } g \in A\}, B' \stackrel{\text{def}}{=} \{g \in G \mid gIm \text{ for all } m \in B\}$$

A **formal concept** is a pair  $(A, B)$ :  $A \subseteq G$ ,  $B \subseteq M$ ,  $A' = B$ , and  $B' = A$ .

- $A$  is the **extent** and  $B$  is the **intent** of the concept  $(A, B)$ .
- The concepts, ordered by  $(A_1, B_1) \geq (A_2, B_2) \iff A_1 \supseteq A_2$  form a complete lattice, called **the concept lattice**  $\underline{\mathfrak{B}}(G, M, I)$ .

# Implications and attribute exploration

- **Implication**  $A \rightarrow B$  for  $A, B \subseteq M$  holds if  $A' \subseteq B'$ , i.e., every object that has all attributes from  $A$  also has all attributes from  $B$ .
- Implications obey **Armstrong rules**:

$$\frac{A \rightarrow B}{A \cup C \rightarrow B} \quad , \quad \frac{A \rightarrow B, A \rightarrow C}{A \rightarrow B \cup C} \quad , \quad \frac{A \rightarrow B, B \rightarrow C}{A \rightarrow C}$$

## Learning aspects

- **Next Closure** an incremental algorithm for constructing implication bases.
- **Attribute exploration** is an interactive learning procedure.

# Association rules with FCA

In mid 1990s partial implications of FCA (M. Luxenburger, 1990) were rediscovered in Data Mining under the name **association rules** (Algorithm **Apriori** by Agrawal, Imielinski).

$A \rightarrow_{c,s} B$  is a partial implication (association rule) of the context  $(G, M, I)$  iff

- $c, s \in [0, 1]$ ;
- $c = \frac{|(A \cup B)'|}{|A'|}$ , called **confidence**;
- $s = \frac{|(A \cup B)'|}{|G'|}$ , called **support**.

# Association rules with FCA

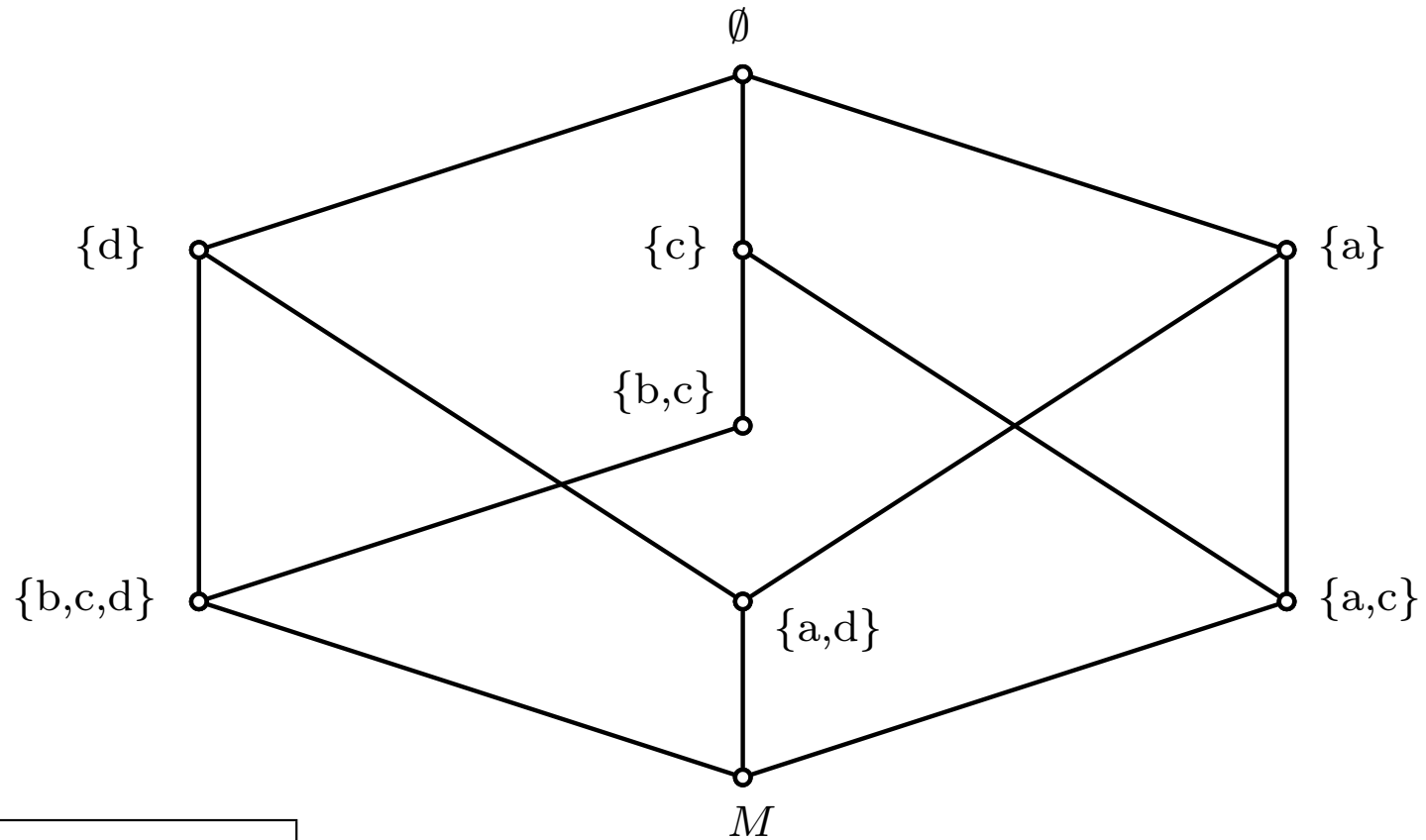
What are “most interesting” association rules  $A \rightarrow_{c,s} B$  with given confidence  $c$  and support  $s$ ?



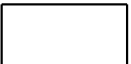

Those with possible smallest  $A$  and largest  $B$ .

We increase  $B$  by taking sets  $B \subseteq B_1 \subseteq \dots \subseteq B_k$  while  $(A \cup B_k)' = (A \cup B)'$  still holds. The largest such  $B_k$  is the one for which  $(A \cup B_k)'' = (A \cup B_k)$ .

Thus, one should look for closed sets of attributes  $(A \cup B_k)$ . If support is maximized, one should look for sets  $(A \cup B_k)$  with large  $|(A \cup B_k)'|$ , i.e., for **frequent** sets of attributes.

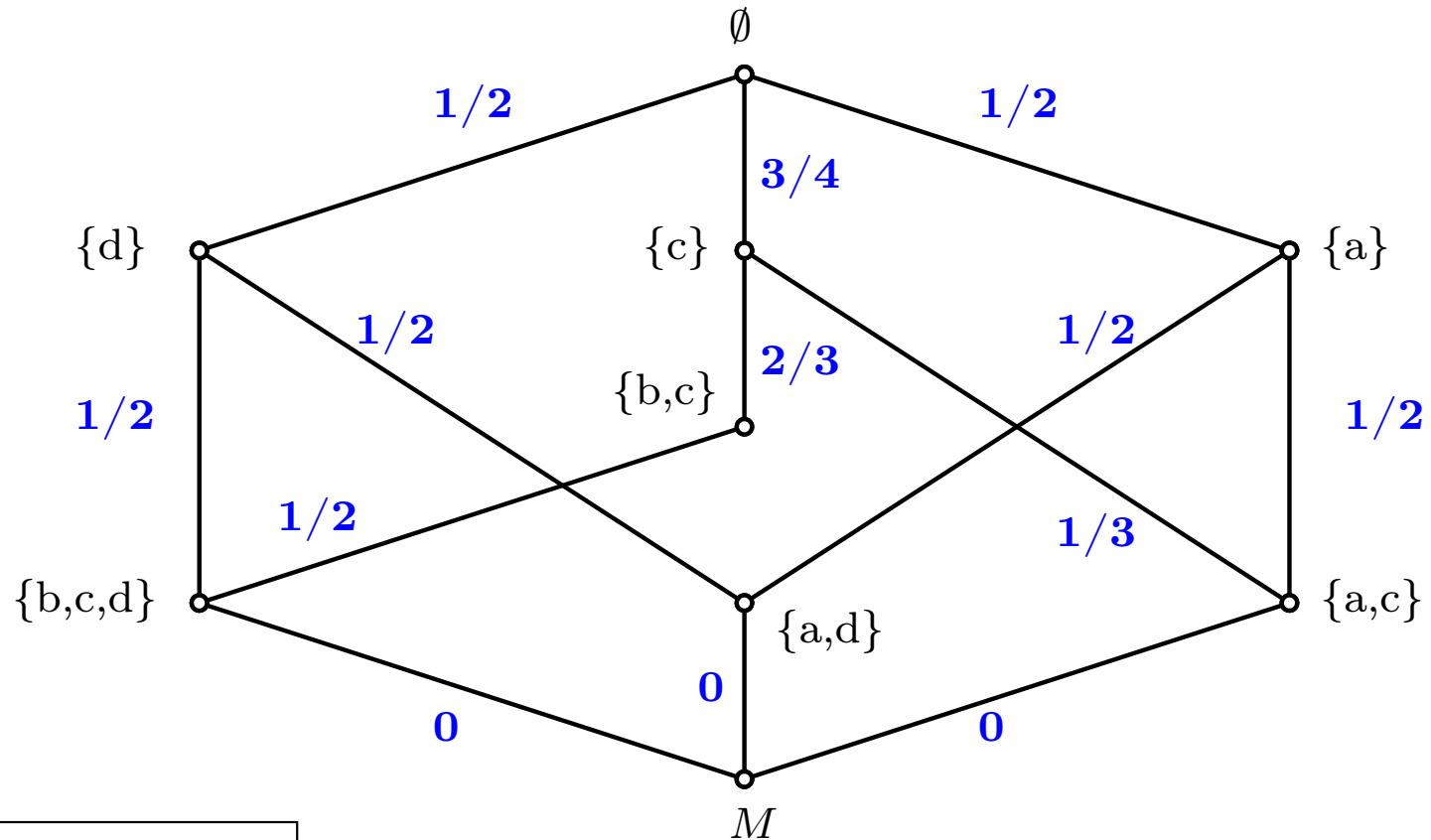
# Association rules with FCA



	$G \setminus M$	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

- a stays for “to have exactly three vertices”
- b stays for “to have exactly four vertices”
- c stays for “to have a right angle”
- d stays for “to have equal sizes”

# Association rules with FCA



	$G \setminus M$	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

## Good rules

with  $\text{sup} \geq 1/2$  and  $\text{conf} \geq 2/3$ :

1.  $\emptyset \rightarrow c$ ,  $\text{sup}(\emptyset \rightarrow c) = \text{conf}(\emptyset \rightarrow c) = 3/4$ ;
2.  $c \rightarrow b$ ,  $\text{sup}(c \rightarrow b) = 1/2$ ,  $\text{conf}(c \rightarrow b) = 2/3$ .

# Outline

1. Lattices in Machine Learning
2. Learning implications and association rules
- 3. JSM-hypotheses**
4. Decision trees
5. Version spaces
6. Learning with Pattern Structures
7. Conclusions

# JSM-method. 1

One of the first models of machine learning that used lattices (closure systems) was the JSM-method by V. Finn.

V. K. Finn, On Machine-Oriented Formalization of Plausible Reasoning in the Style of F. Backon – J. S. Mill, *Semiotika Informatika*, **20** (1983), 35-101 [in Russian]

**Method of Agreement** (First canon of inductive logic):

“ If two or more instances of the phenomenon under investigation have only one circumstance in common, ... [it] is the cause (or effect) of the given phenomenon.”

John Stuart Mill, *A System of Logic, Ratiocinative and Inductive*, London, 1843

In the JSM-method positive hypotheses are sought among intersections of positive example given as sets of attributes, same for negative hypotheses. Various additional conditions can be imposed on these intersections.

# JSM-method. 2

**Logical means of the JSM-method:** Many-valued many-sorted extension of the First-Order Predicate Logic with quantifiers over tuples of variable length (weak Second Order).

**Example:** Formalization of the Mill's Method of Agreement:

$$\begin{aligned} \mathcal{M}_{a,n}^+(V, W) &:= \exists k \widetilde{\mathcal{M}}_{a,n}^+(V, W, k), \\ \widetilde{\mathcal{M}}_{a,n}^+(V, W, k) &:= \exists Z_1 \dots \exists Z_k \exists U_1 \dots \exists U_k \left( \bigwedge_{i=1}^k J_{\langle 1,n \rangle}(Z_i \Rightarrow_1 U_i) \& \right. \\ &\& \forall U (J_{\langle 1,n \rangle}(Z_i \Rightarrow_1 U) \rightarrow \mathcal{U} \subseteq U_i) \& \& (Z_1 \cap \dots \cap Z_k) = V \& V \neq \emptyset \& W \neq \emptyset \& \\ &\& \forall i \forall j ((i \neq j) \& 1 \leq i, j \leq k) \rightarrow Z_i \neq Z_j \& \& \forall X \forall Y ((J_{\langle 1,n \rangle}(X \Rightarrow_1 Y) \& \\ &\& \forall \mathcal{U} (J_{\langle 1,n \rangle}(X \Rightarrow_1 U) \rightarrow \mathcal{U} \subseteq Y) \& \& V \subseteq X) \rightarrow (W \subseteq Y \& \left( \bigvee_{i=1}^k (X = Z_i) \right))) \& k \geq 2). \end{aligned}$$

The predicate defines a closure system (w.r.t.  $\bigcap$ ) generated by descriptions of positive examples. At the same time,  $\bigcap$  is a means of expressing “similarity” of objects given by attribute sets.

# FCA translation

[Ganter, Kuznetsov 2000]

A target attribute  $w \notin M$ ,

- **positive examples:** Set  $G_+ \subseteq G$  of objects known to have  $w$ ,
- **negative examples:** Set  $G_- \subseteq G$  of objects known not to have  $w$ ,
- **undetermined examples:** Set  $G_\tau \subseteq G$  of objects for which it is unknown whether they have the target attribute or do not have it.

Three subcontexts of  $\mathbb{K} = (G, M, I)$ :  $\mathbb{K}_\varepsilon := (G_\varepsilon, M, I_\varepsilon)$ ,  $\varepsilon \in \{-, +, \tau\}$ .

A **positive hypothesis**  $H \subseteq M$  is an intent of  $\mathbb{K}_+$  not contained in the intent  $g^-$  of any negative example  $g \in G_-$ :

$$\forall g \in G_- \quad H \not\subseteq g^-$$

# Example of a learning context

G \ M	color	firm	smooth	form	fruit
apple	yellow	no	yes	round	+
grapefruit	yellow	no	no	round	+
kiwi	green	no	no	oval	+
plum	blue	no	yes	oval	+
toy cube	green	yes	yes	cubic	-
egg	white	yes	yes	oval	-
tennis ball	white	no	no	round	-

# Natural scaling of the context

G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
apple		×				×	×		×		+
grapefruit		×				×		×	×		+
kiwi			×			×		×		×	+
plum				×		×	×			×	+
toy cube			×		×		×			×	-
egg	×				×		×			×	-
tennis ball	×					×		×	×		-

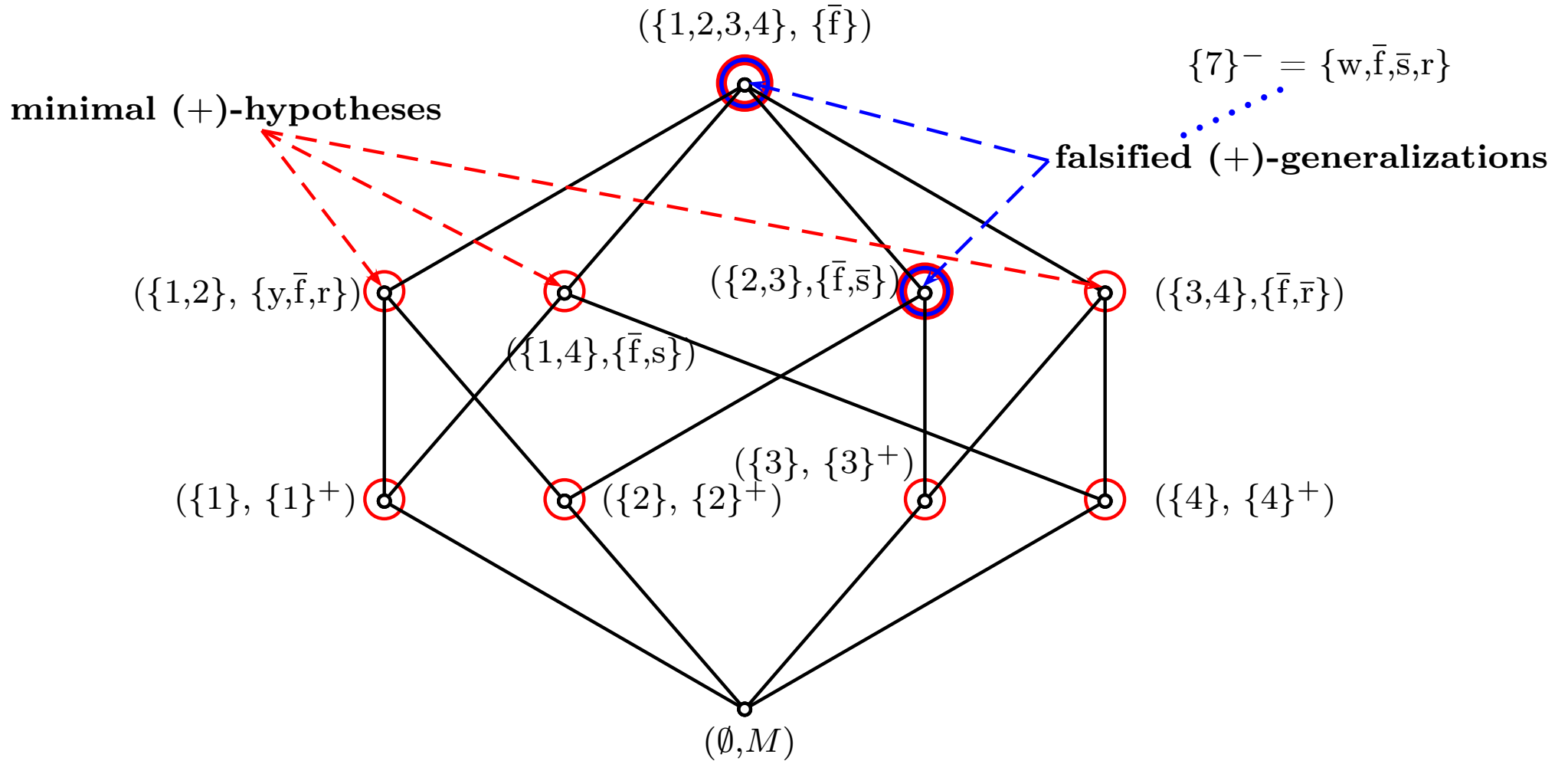
Abbreviations:

“g” for green, “y” for yellow, “w” for white, “f” for firm, “ $\bar{f}$ ” for nonfirm,

“s” for smooth, “ $\bar{s}$ ” for nonsmooth, “r” for round,

“ $\bar{r}$ ” for nonround.

# Positive Concept Lattice



G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
apple		×				×	×		×		+
grapefruit		×				×		×	×		+
kiwi			×			×		×		×	+
plum				×		×	×			×	+
toy cube			×		×		×			×	-
egg	×				×		×			×	-
tennis ball	×					×		×	×		-

# Classification of undetermined example $g_\tau$

- If  $g'_\tau$  contains a positive and no negative hypothesis,  $g_\tau$  is **classified positively** (predicted to have  $w$ ).
- If  $g'_\tau$  contains a negative and no positive hypothesis,  $g_\tau$  is **classified negatively**.
- If  $g'_\tau$  contains hypotheses of both kinds, or if  $g'_\tau$  contains no hypothesis at all, then the classification is **contradictory** or **undetermined**, respectively.

For classification purposes it suffices to have all **minimal** (w.r.t.  $\subseteq$ ) hypotheses

# Classifying undetermined example **mango**

	G\M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	-
6	egg	×				×		×			×	-
7	tennis ball	×					×		×	×		-
8	mango		×				×	×			×	$\tau$

The object **mango** is classified positively:

- $\{\bar{r}, \bar{f}\}$  is a (+)-hypothesis,  
 $\{\bar{r}, \bar{f}\} \subseteq \text{mango}' = \{y, \bar{f}, s, \bar{r}\}$ ;
- for (-)-hypotheses  $\{w\}$  and  $\{f, s, \bar{r}\}$ :  
 $\{w\} \not\subseteq \text{mango}'$ ,  
 $\{\bar{f}, s, \bar{r}\} \not\subseteq \text{mango}'$ .

# Variations of the learning model

- allowing for  $\alpha\%$  of counterexamples (for hypotheses and/or classifications),
- imposing other logical conditions (e.g. of the “Difference method” of J. S. Mill):  
**Finn’s “lattice of methods”**,
- nonsymmetric classification (applying only (+)-hypotheses),
- and so on.

**The invariant:** hypotheses are sought among positive and negative intents.

# Analysis of wheel chains quality

Data from [ADAC Magazine, 1999, 11]

#	system	mount	price	con	snow	ice	dur	grade
1	SK	F	206	1.9	1.4	1.8	2.7	1.8
2	SRK	F or R	520	2.1	0.8	3.8	2.3	1.9
3	SK	F	160	1.7	1.9	1.6	3.7	2.1
4	SK	F	213	1.7	2.0	2.4	3.4	2.1
5	SMS	F or R	598	1.6	2.4	2.7	2.8	2.2
6	SK	F	109	2.0	1.9	2.4	3.7	2.3
7	SRK	F or R	325	2.0	2.1	3.2	2.8	2.3
8	SMS	F or R	498	1.5	3.3	3.5	2.0	2.4
9	SRK	F or R	396	2.8	2.1	3.1	2.5	2.6
10	SRK	F or R	325	2.2	2.2	4.6	3.2	2.6
11	SRK	F or R	389	2.0	2.2	3.3	4.3	2.6
12	SRK	F	298	2.5	2.3	3.3	2.8	2.6
13	SK	F	149	1.9	2.5	4.0	3.8	2.6
14	SMS	F or R	684	1.7	3.3	4.4	2.2	2.6
15	SK	F	99	2.8	2.2	2.5	4.0	2.7
16	SK	F	140	2.6	2.3	3.3	3.4	2.7
17	SK	F	215	2.3	3.8	4.8	2.3	3.1

# Target property: general quality estimate grade

We consider that a good chain has the estimate not larger than 2.1.

## Positive context

#	system	mount	price	con	snow	ice	dur	(grade)
1	SK	F	206	1.9	1.4	1.8	2.7	(1.8)
2	SRK	F or R	520	2.1	0.8	3.8	2.3	(1.9)
3	SK	F	160	1.7	1.9	1.6	3.7	(2.1)
4	SK	F	213	1.7	2.0	2.4	3.4	(2.1)

# Target property: general quality estimate grade

We consider that a bad chain has the estimate not smaller than 2.6

## Negative context

No	system	mount	price	con	snow	ice	dur	(grade)
9	SRK	F or R	396	2.8	2.1	3.1	2.5	(2.6)
10	SRK	F or R	325	2.2	2.2	4.6	3.2	(2.6)
11	SRK	F or R	389	2.0	2.2	3.3	4.3	(2.6)
12	SRK	F	298	2.5	2.3	3.3	2.8	(2.6)
13	SK	F	149	1.9	2.5	4.0	3.8	(2.6)
14	SMS	F or R	684	1.7	3.3	4.4	2.2	(2.6)
15	SK	F	99	2.8	2.2	2.5	4.0	(2.7)
16	SK	F	140	2.6	2.3	3.3	3.4	(2.7)
17	SK	F	215	2.3	3.8	4.8	2.3	(3.1)

# A possible scaling

<b>system</b>	SK	SRK	SMS	
<b>mount</b>	F	F or R		
<b>price</b>	$\leq 160$	$\leq 215$	$\leq 520$	$> 520$
<b>con</b>	$\leq 2.1$	$\leq 2.5$	$> 2.5$	
<b>snow</b>	$\leq 2.0$	$> 2.0$		
<b>ice</b>	$\leq 2.4$	$\leq 3.0$	$\leq 4.0$	$> 4.0$
<b>dur</b>	$\leq 3$	$\leq 3.7$	$> 3.7$	

# Hypotheses vs. minimal-premise implications

A minimal positive hypothesis:

$\{\mathbf{con} \leq 2.1, \mathbf{snow} \leq 2.0, \mathbf{ice} \leq 4, \mathbf{dur} \leq 3.7\}$

Interpretation: a good chain from the customer's viewpoint.

Minimal-premise implications:

$$\{\mathbf{snow} \leq 2.0\} \rightarrow \{\text{good quality}\},$$
$$\{\mathbf{con} \leq 2.1, \mathbf{ice} \leq 4, \mathbf{dur} \leq 3.7\} \rightarrow \{\text{good quality}\},$$

Interpretation: a good chain from the producer's viewpoint.

Nonminimal hypotheses: taxonomy of positive examples. For example, hypothesis

$$\{\text{SK, F, price} \leq 215, \mathbf{con} \leq 2.1, \mathbf{snow} \leq 2.0, \mathbf{ice} \leq 4.0, \mathbf{dur} \leq 3.7\}$$

describes a certain consumer class of chains (cheap, with good behavior on snow and satisfiable behavior on ice).

# Toxicology analysis by means of the JSM-method

*Bioinformatics, 19(2003)*

V. G. Blinova, D. A. Dobrynin, V. K. Finn, S. O. Kuznetsov and E. S. Pankratova

**Predictive Toxicology Challenge:** (PTC) Workshop at the joint 5th European Conference on Knowledge Discovery in Databases (KDD'2001) and the 12th European Conference on Machine Learning (ECML'2001), Freiburg.

**Organizers:** Machine Learning groups of the Freiburg University, Oxford University, University of Wales.

**Toxicology experts:** US Environmental Protection Agency, US National Institute of Environmental and Health Standards.

# Toxicology analysis by means of the JSM-method

*Bioinformatics, 19(2003)*

**Training Sample:** Data of the National Toxicology Program (NTP) with 120 to 150 positive examples and 190 to 230 negative examples of toxicity: molecular graphs with indication of whether a substance is toxic for four sex/species groups: {male, female}  $\times$  {mice, rats}.

**Testing Sample:** Data of Food and Drug Administration (FDA): about 200 chemical compounds with known molecular structures, whose (non)toxicity, known to organizers, was to be predicted by participants.

**Participants:** 12 research groups (world-wide), each with up to 4 prediction models for every sex/species group.

**Evaluation:** ROC diagrams

**Stages of the Competition:**

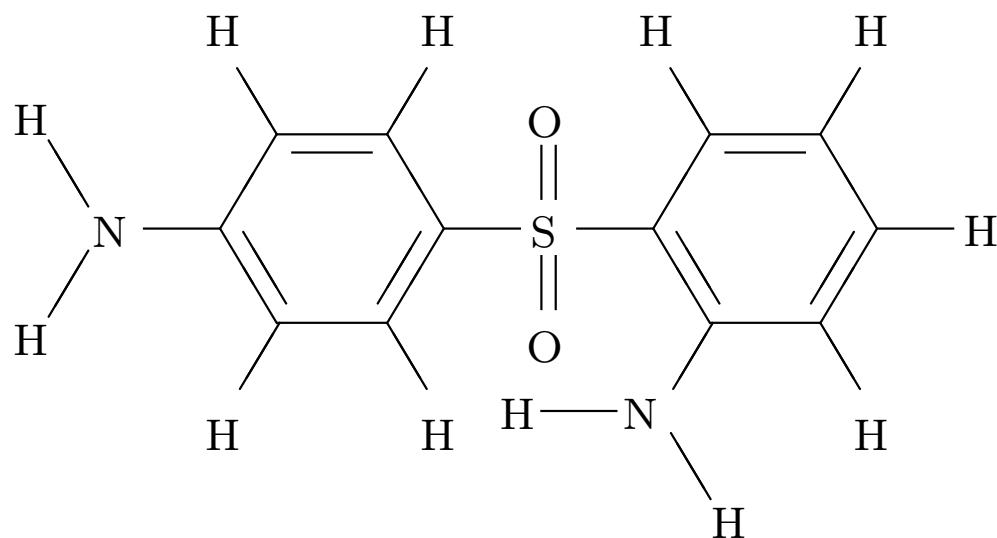
1. Encoding of chemical structures in terms of attributes,
2. Generation of classification rules,
3. Prediction by means of classification rules.

Results of each stage were made public (put on a web site).

# Example of Coding with FCSS

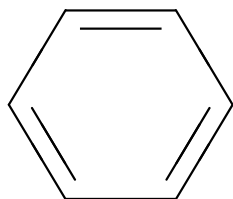
Chemical structure

Complete list of descriptors

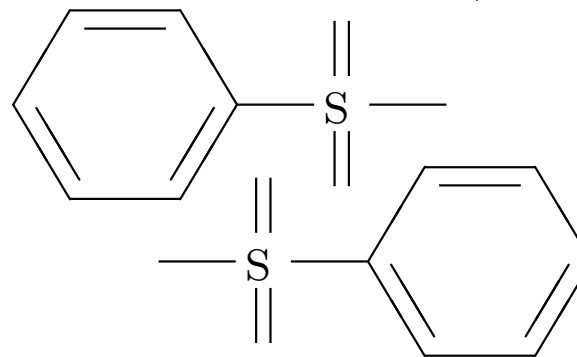


6,06	×2
0200331	×2
1300241	×2
2400331	×2
0264241	
0262241	

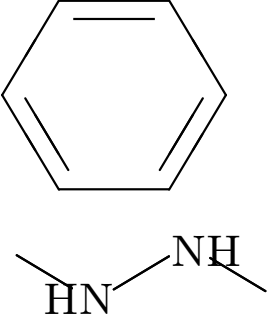
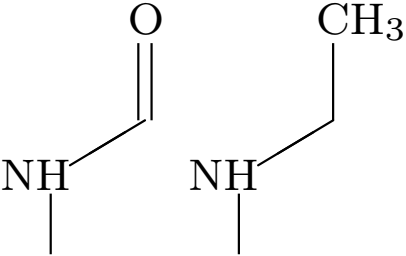
6,06 (cyclic descriptors)



0200331 (linear descriptors)

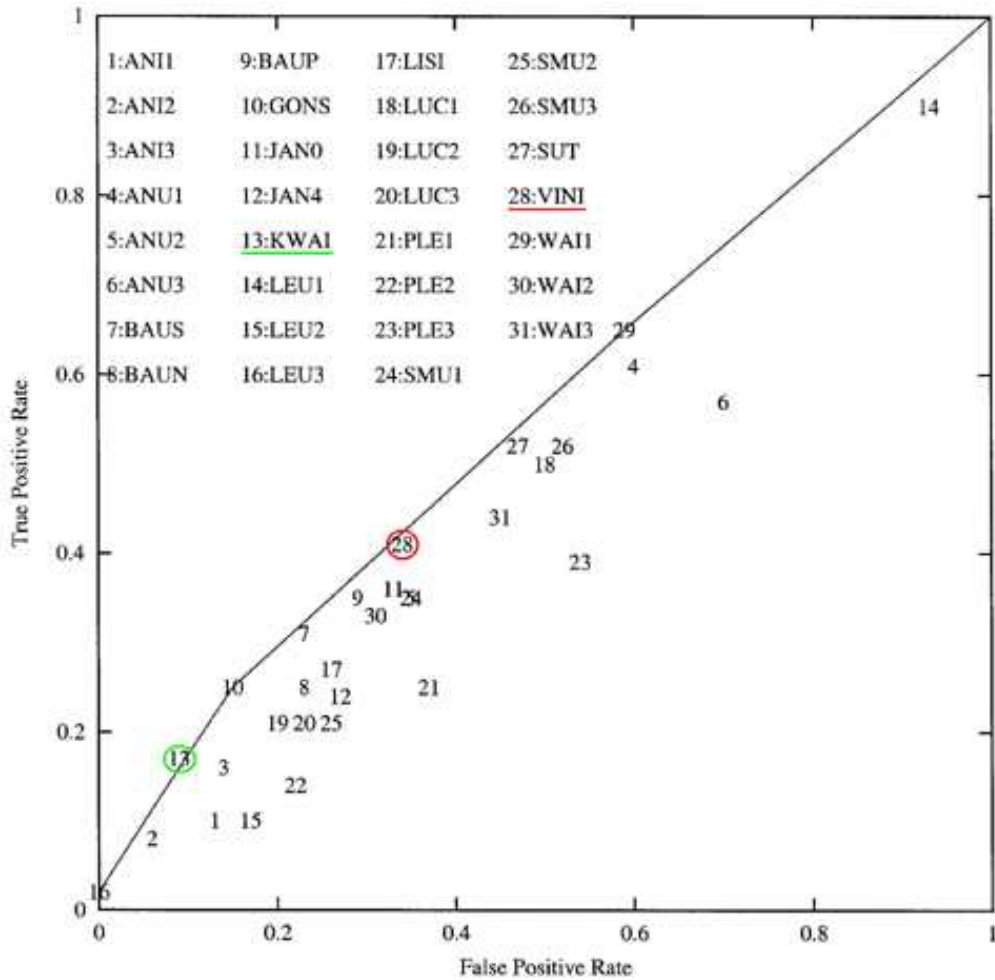


# Some positive hypotheses

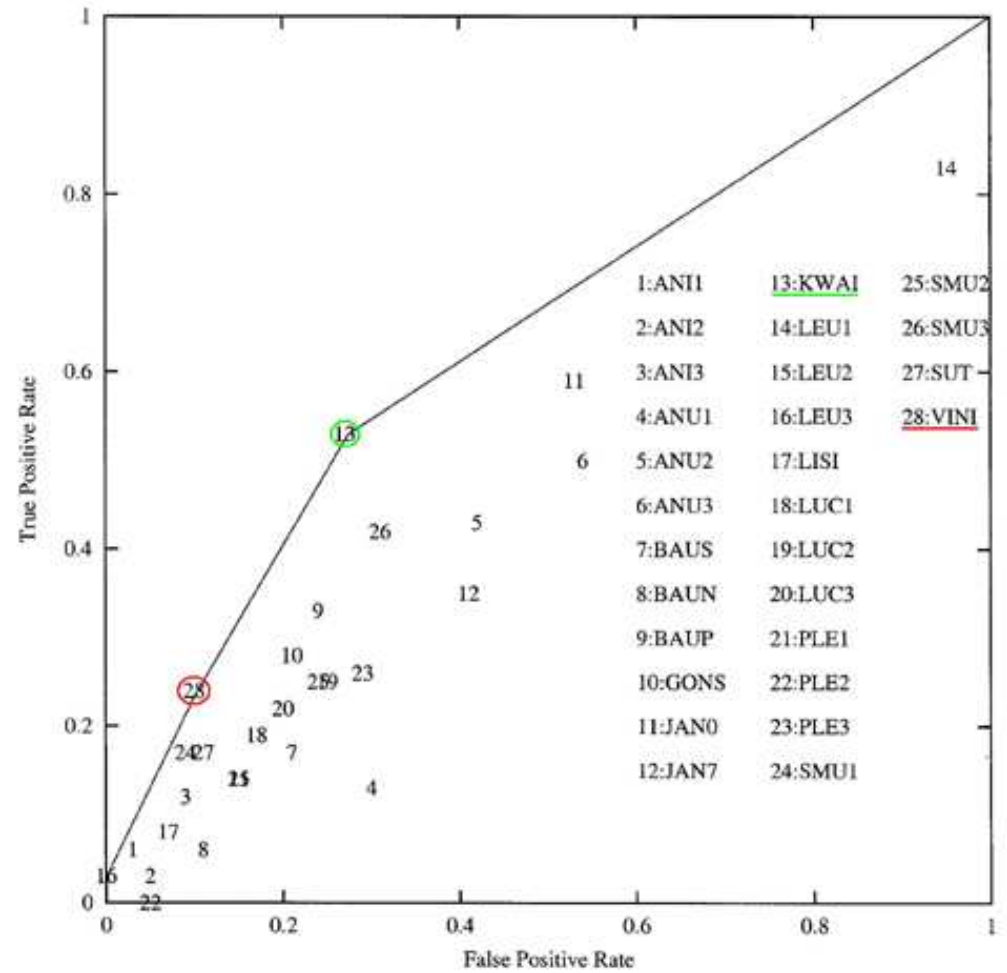
Molecular graph	FCCS descriptors (encoding)	# of predictions in sex/species group(s)
 <p>Chemical structure showing a benzene ring (hexagon with three double bonds) and an ethylamino group (-NH-CH<sub>2</sub>-CH<sub>3</sub>) attached to one of the carbons.</p>	6,06 0200021	2FR
 <p>Chemical structure showing a carbonyl group (C=O) bonded to an NH group, and a methyl group (CH<sub>3</sub>) bonded to another NH group.</p>	0201131 0202410	1FR 1MM

# ROC diagrams : Rats

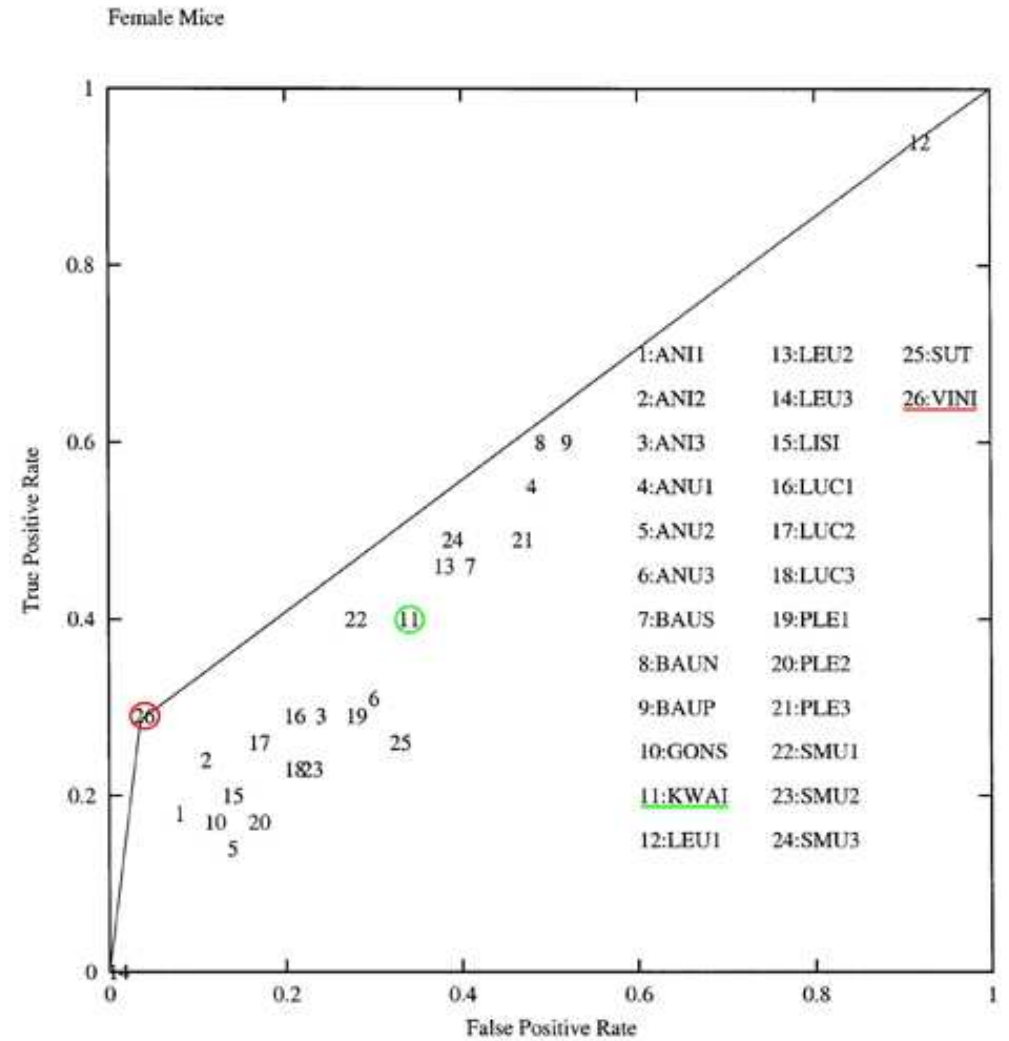
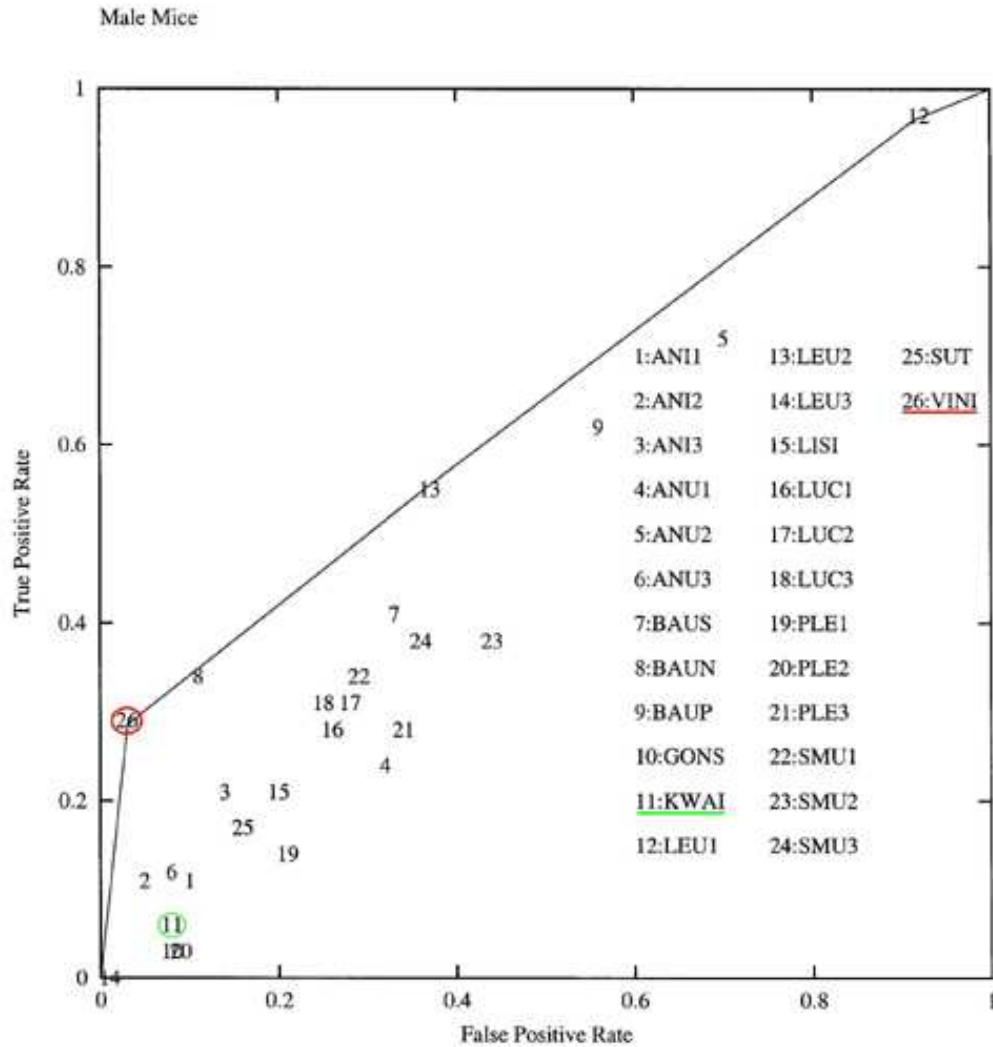
Male Rats



Female Rats



# ROC diagrams : Mice



# Outline

1. Lattices in Machine Learning
2. Learning implications and association rules
3. JSM-hypotheses
- 4. Decision trees**
5. Version spaces
6. Learning with Pattern Structures
7. Conclusions

# Decision trees

**Input:** descriptions of positive and negative examples as sets of attribute values.

All vertices (except for the root) are labeled by attributes and edges are labeled by values of the attributes (e.g., 0 or 1 in case of binary attributes), each leaf is additionally labeled by a class + or -: examples with all attribute values in the path leading from the root to the leaf belong to a certain class, either + or -.

Systems like **ID3** [R. Quinlan 86] compute the value of the *information gain* (IG), for each vertex and each attribute not chosen in the branch above.

The algorithm sequentially extends branches by choosing attributes with the highest information gain (that “most correctly separates” objects from classes + and -).

Extension of a branch terminates when a next attribute value together with attribute values chosen before uniquely classify examples into one of the classes + or -. An algorithm can stop earlier to avoid *overfitting*.

# Entropy

In systems like **ID3**, **C4.5** a next chosen attribute should maximize some information functional, e.g., information gain (IG), based on the entropy w.r.t. the target attribute

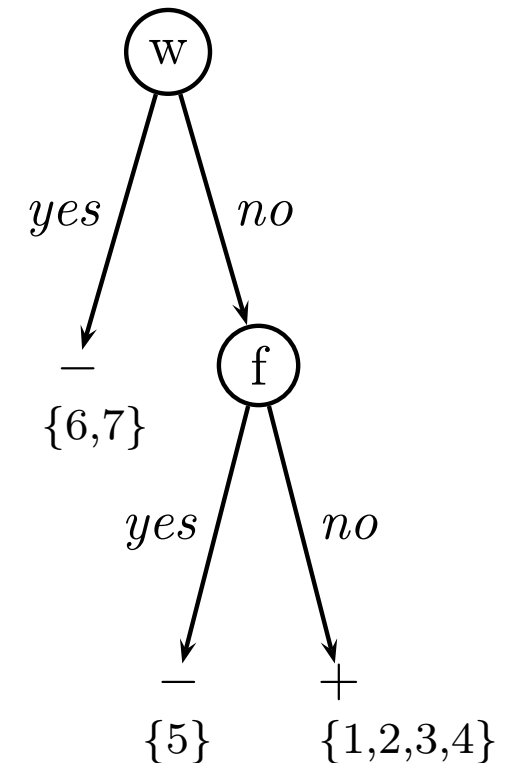
$$\text{Ent}(A) := - \sum_{\varepsilon \in \{+, -\}} p(\varepsilon | A) \cdot \log_2 p(\varepsilon | A),$$

$\{+, -\}$  are values of the target attribute  $p(\varepsilon | A)$  is the conditional sample probability (for the training set) that an object having a set of attributes  $A$  belongs to a class  $\varepsilon \in \{+, -\}$ .

# An example of a decision tree

Decision tree obtained by the IG-based algorithm:

G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
apple		×				×	×		×		+
grapefruit		×				×		×	×		+
kiwi			×			×		×		×	+
plum				×		×	×			×	+
toy cube			×		×		×			×	-
egg	×				×		×			×	-
tennis ball	×					×		×	×		-

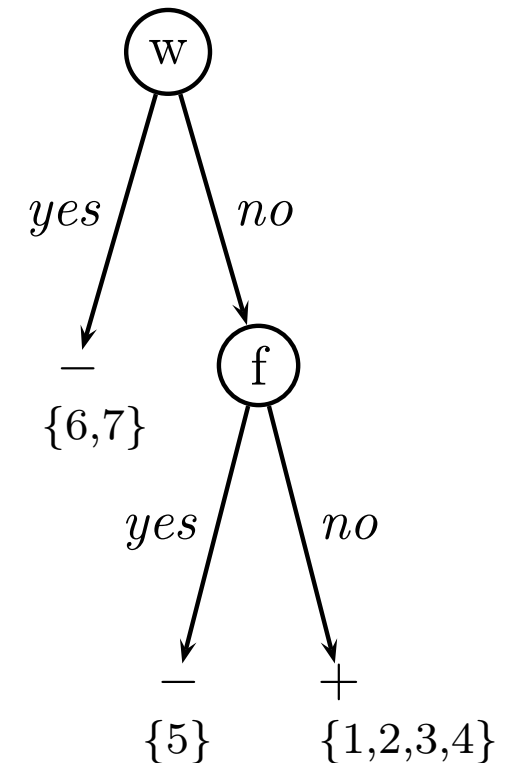


- Note that attributes **f** and **w** has the same IG value (a similar tree with **f** at the root is also optimal), IG-based algorithms usually take the first attribute with the same value of IG.
- The tree corresponds to three implications  $\{w\} \rightarrow -$ ,  $\{\bar{w}, f\} \rightarrow -$ ,  $\{\bar{w}, \bar{f}\} \rightarrow +$ .

# An example of a decision tree

Decision tree obtained by the IG-based algorithm:

G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
apple		×				×	×		×		+
grapefruit		×				×		×	×		+
kiwi			×			×		×		×	+
plum				×		×	×			×	+
toy cube			×		×		×			×	-
egg	×				×		×			×	-
tennis ball	×					×		×	×		-



- The closures of the implication premises make the corresponding negative and positive hypotheses.
- Note that the hypothesis  $\{\bar{w}, f\}''$  is not minimal, since there is a minimal hypothesis  $\{f\}''$  contained in it. The minimal hypothesis  $\{f\}''$  corresponds to a decision path of the IG-optimal tree with the attribute  $f$  at the root.

# Decision trees in FCA terms

Training data is given by the context  $\mathbb{K}_{+-} = (G_+ \cup G_-, M, I_+ \cup I_-)$  with the derivation operator  $(\cdot)'$ . In FCA terms  $\mathbb{K}_{+-}$  is the *subposition* of  $\mathbb{K}_+$  and  $\mathbb{K}_-$ .

**Assumption.** The set of attributes  $M$  is *dichotomized*: For each attribute  $m \in M$  there is an attribute  $\bar{m} \in M$ , a “negation” of  $m$ :  $\bar{m} \in g'$  iff  $m \notin g'$ .

- A subset of attributes  $A \subseteq M$  is *noncontradictory* if  $m \notin A$  or  $\bar{m} \notin A$ .
- A subset of attributes  $A \subseteq M$  is *complete* if for every  $m \in M$  one has  $m \in A$  or  $\bar{m} \in A$ .

The construction of an arbitrary decision tree proceeds by sequentially choosing attributes. First we ignore the optimization aspect related to the information gain.

A sequence of attributes  $\langle m_1, \dots, m_k \rangle$  is called a *decision path* if  $\{m_1, \dots, m_k\}$  is noncontradictory and there exists an object  $g \in G_+ \cup G_-$  such that  $\{m_1, \dots, m_k\}' \subseteq g'$  (i.e., there is an example with this set of attributes).

# Decision trees in FCA terms

- A decision path  $\langle m_1, \dots, m_i \rangle$  is a (*proper*) *subpath* of a decision path  $\langle m_1, \dots, m_k \rangle$  if  $i \leq k$  ( $i < k$ , respectively).
- A decision path  $\langle m_1, \dots, m_k \rangle$  is called *full* if objects having attributes  $\{m_1, \dots, m_k\}$  are all either positive or negative examples.
- A full decision path is *irredundant* if none of its subpaths is a full decision path. The set of all chosen attributes in a full decision path can be considered as a sufficient condition for an object to belong to a class  $\varepsilon \in \{+, -\}$ .
- A **decision tree** is a set of full decision paths.
- The *closure of a decision path*  $\langle m_1, \dots, m_k \rangle$  is the closure of the corresponding set of attributes, i.e.,  $\{m_1, \dots, m_k\}''$ .
- A sequence of concepts with decreasing extents is called a *descending chain*.
- A chain starting at the top element of the lattice is called *rooted*.

# Semiproduct of dichotomic scales

The *semiproduct* of two contexts  $\mathbb{K}_1$  and  $\mathbb{K}_2$  is defined by

$\mathbb{K}_1 \boxtimes \mathbb{K}_2: = (G_1 \times G_2, M_1 \dot{\cup} M_2, \nabla)$ , where

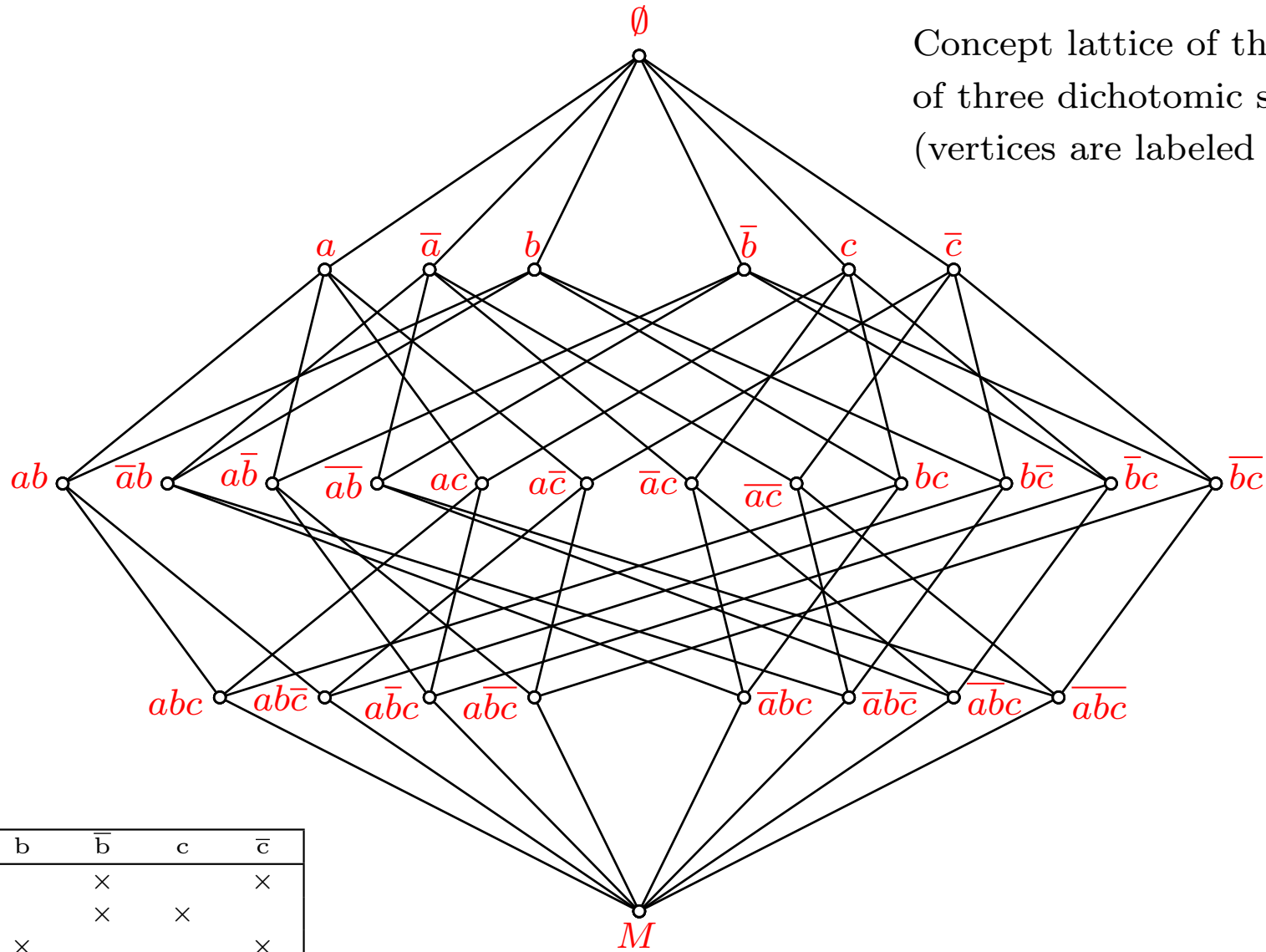
$$(g_1, g_2) \nabla (j, m): \longleftrightarrow g_j I_j m \quad \text{for } j \in \{1, 2\}.$$

**Example.** The semiproduct of three dichotomic scales  $(\{0, 1\}, \{0, 1\}, =)$ :

	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$
1		×		×		×
2		×		×	×	
3		×	×			×
4		×	×		×	
5	×			×		×
6	×			×	×	
7	×		×			×
8	×		×		×	

# Semiproduct of dichotomic scales

Concept lattice of the semiproduct  
of three dichotomic scales  
(vertices are labeled by intents)



$\mathbb{D}_1 \times \mathbb{D}_2 \times \mathbb{D}_3$

	a	$\bar{a}$	b	$\bar{b}$	c	$\bar{c}$
1		×		×		×
2		×		×	×	
3		×	×			×
4		×	×		×	
5	×			×		×
6	×			×	×	
7	×		×			×
8	×		×		×	

# Decision trees vs. semiproducts of dichotomic scales

Consider the following context  $\mathbb{K} = (G, M, I)$ :

The set of objects  $G$  is of size  $2^{|M|/2}$  and the relation  $I$  is such that the set of object intents is exactly the set of complete noncontradictory subsets of attributes.

In terms of FCA the context  $\mathbb{K}$  is the *semiproduct* of  $|M|/2$  *dichotomic scales* or  $\mathbb{K} = D_1 \times \dots \times D_{|M|/2}$  (denoted by  $\times_M D$  for short), where each dichotomic scale  $D_i$  stays for the pair of attributes  $(m, \bar{m})$ .

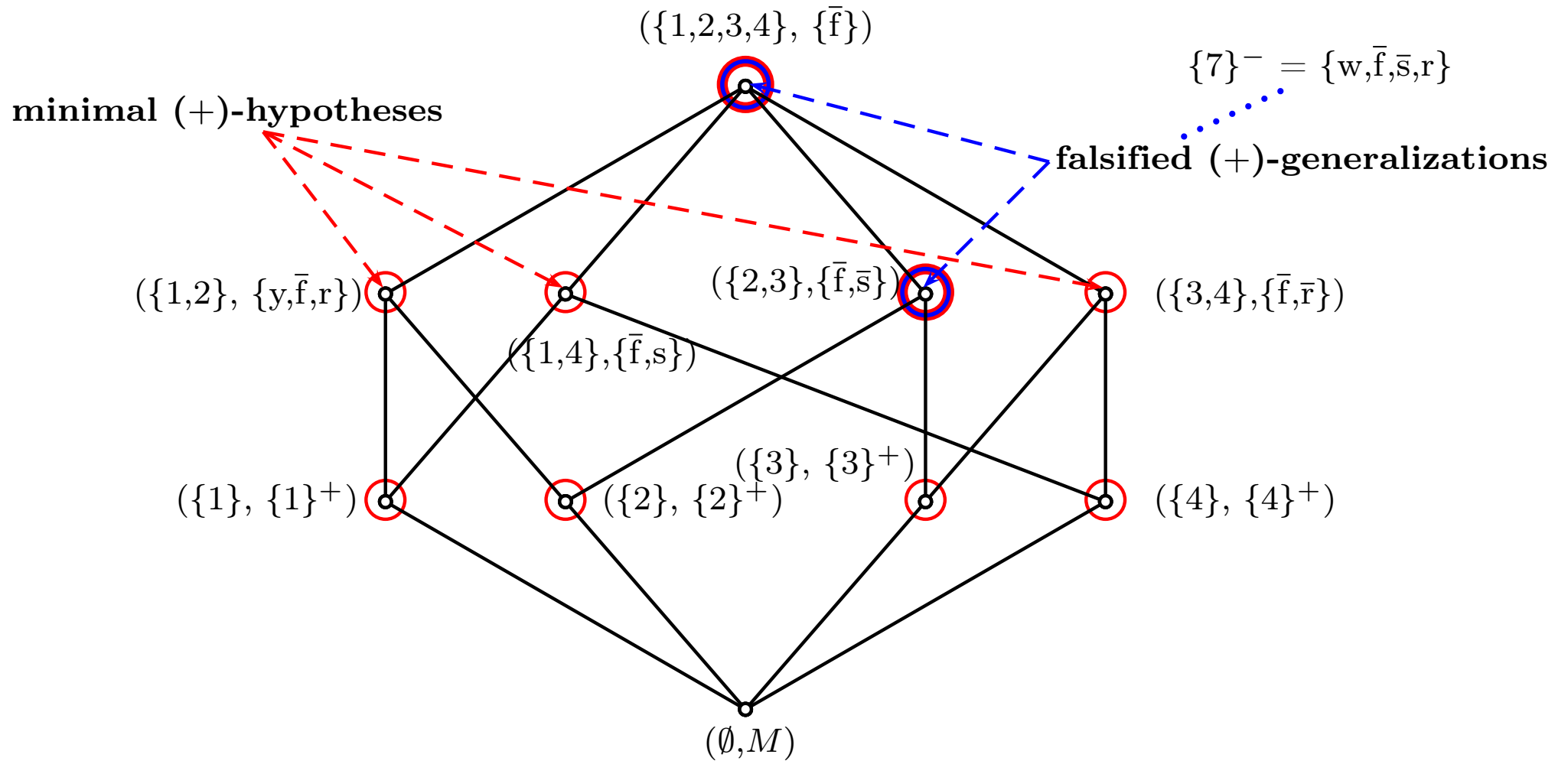
**Proposition.** *Every decision path is a rooted descending chain in  $\underline{\mathfrak{B}}(\times_M D)$  and every rooted descending chain consisting of concepts with nonempty extents in  $\underline{\mathfrak{B}}(\times_M D)$  is a decision path.*

# Decision trees vs. semiproducts of dichotomic scales

To relate decision trees to hypotheses introduced above we consider again the contexts  $\mathbb{K}_+ = (G_+, M, I_+)$ ,  $\mathbb{K}_- = (G_-, M, I_-)$ , and  $\mathbb{K}_{+-} = (G_+ \cup G_-, M, I_+ \cup I_-)$ . The context  $\mathbb{K}_{+-}$  can be much smaller than  $\bigotimes_M D$  because the latter always has  $2^{|M|/2}$  objects while the number of objects in the former is the number of examples. Also the lattice  $\underline{\mathfrak{B}}(\mathbb{K}_{+-})$  can be much smaller than  $\underline{\mathfrak{B}}(\bigotimes_M D)$ .

**Proposition.** *A full decision path  $\langle m_1, \dots, m_k \rangle$  corresponds to a rooted descending chain  $\langle (m''_1, m'_1), \dots, (\{m_1, \dots, m_k\}'', \{m_1, \dots, m_k\}') \rangle$  of the line diagram of  $\underline{\mathfrak{B}}(\mathbb{K}_{+-})$  and the closure of each full decision path  $\langle m_1, \dots, m_k \rangle$  is a hypothesis, either positive or negative. Moreover, for each minimal hypothesis  $h$ , there is a full irredundant path  $\langle m_1, \dots, m_k \rangle$  such that  $\{m_1, \dots, m_k\}'' = h$ .*

# Positive Concept Lattice



	G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	-
6	egg	×				×		×			×	-
7	tennis ball	×					×		×	×		-
8	mango			×			×	×			×	$\tau$

# Discussion of the propositions

The propositions show the difference between hypotheses and irredundant decision paths.

- Hypotheses correspond to “most cautious” (most specific) classifier consistent with the data: they are least general generalizations of descriptions of positive examples (object intents).
- The shortest decision paths (in no decision tree there exist full subpaths) correspond to the “most courageous” (“most discriminant”): they are most general generalizations of positive example descriptions.
- It is not guaranteed that for a given training set there is a decision tree such that minimal hypotheses are among closures of its paths.
- In general, to obtain all minimal hypotheses as closures of decision paths one needs to consider not only paths optimal w.r.t. the information gain.

The issues of generality of generalizations are naturally captured in terms of version spaces.

# Recalling the Information Gain

For dichotomized attributes the information gain is natural to define for a pair of attributes  $m, \bar{m} \in M$ .

For a decision path  $\langle m_1, \dots, m_k \rangle$

$$\text{IG}(m) := -\frac{|A'_m|}{|G|} \text{Ent}(A_m) - \frac{|A'_{\bar{m}}|}{|G|} \text{Ent}(A_{\bar{m}}),$$

where  $A_m := \{m_1, \dots, m_k, m\}$ ,  $A_{\bar{m}} := \{m_1, \dots, m_k, \bar{m}\}$ , and for  $A \subseteq M$

$$\text{Ent}(A) := - \sum_{\varepsilon \in \{+, -\}} p(\varepsilon | A) \cdot \log_2 p(\varepsilon | A),$$

$\{+, -\}$  are values of the target attribute  $p(\varepsilon | A)$  is the conditional sample probability (for the training set) that an object having a set of attributes  $A$  belongs to a class  $\varepsilon \in \{+, -\}$ .

# Information Gain is nonsensitive to closure

If the derivation operator  $(\cdot)'$  is associated with the context  $(G_+ \cup G_-, M, I_+ \cup I_-)$ , then

$$p(\varepsilon \mid A) = \frac{|A' \cap G_\varepsilon|}{|A'|} = \frac{|(A'')' \cap G_\varepsilon|}{|(A'')'|} = p(\varepsilon \mid A'')$$

by the property of the derivation operator  $(\cdot)'$ :  $(A'')' = A'$ .

Hence,

- Instead of considering decision paths, one can consider their closures without affecting the values of the information gain.

In FCA terms: instead of the concept lattice  $\underline{\mathfrak{B}}(\bigotimes_M D)$  one can consider the concept lattice  $\underline{\mathfrak{B}}(\mathbb{K}_{+-}) = \underline{\mathfrak{B}}(G_+ \cup G_-, M, I_+ \cup I_-)$ , which can be much smaller.

- If implication  $m \rightarrow n$  holds in the context  $\mathbb{K}_{+-} = (G_+ \cup G_-, M, I_+ \cup I_-)$ , then an IG-based algorithm will not choose attribute  $n$  in the branch below chosen  $m$  and will not choose  $\bar{m}$  in the branch below chosen  $\bar{n}$ .

# Outline

1. Lattices in Machine Learning
2. Learning implications and association rules
3. JSM-hypotheses
4. Decision trees
- 5. Version spaces**
6. Learning with Pattern Structures
7. Conclusions

# Version spaces

T. Mitchell, Generalization as Search, *Artificial Intelligence* **18**, no. 2, 1982.

T. Mitchell, Machine Learning, The McGraw-Hill Companies, 1997.

- An **example language**  $L_e$  that describes a **set  $E$  of examples**;
- A **classifier language**  $L_c$  that describes a set  $C$  of **classifiers** (elsewhere called *concepts*);
- A **matching predicate**  $M(c, e)$ : We have  $M(c, e)$  iff  $e$  is an example of classifier  $c$  or  $c$  **matches**  $e$ . The set of classifiers is (partially) ordered by a **subsumption order**: for  $c_1, c_2 \in L_c$ ,

$$c_1 \leq c_2 : \iff \forall_{e \in E} M(c_1, e) \rightarrow M(c_2, e).$$

- Sets  $E_+$  and  $E_-$  of **positive** and **negative examples** of a **target attribute** with  $E_+ \cap E_- = \emptyset$ .

# Version spaces

- **Consistency predicate**  $\text{cons}(c)$ :

$\text{cons}(c)$  holds if for every  $e \in E_+$  the matching predicate  $M(c, e)$  holds and for every  $e \in E_-$  the negation  $\neg M(c, e)$  holds.

- **Version space** is the set of all consistent classifiers:  $\text{VS}(L_c, L_e, M(c, e), E_+, E_-)$ .
- **Learning problem:**

**Given**  $L_c, L_e, M(c, e), E_+, E_-$ .

**Find** the version space  $\text{VS}(L_c, L_e, M(c, e), E_+, E_-)$ .

- **Classification:**

A classifier  $c \in \text{VS}$  **classifies** an example positively if  $c$  matches  $e$ , otherwise it classifies it negatively.

An example  $e$  is  $\alpha\%$ -**classified** if no less than  $\frac{\alpha}{100} \cdot |\text{VS}|$  classifiers classify it positively.

# Version spaces in terms of boundary sets

T. Mitchell, Generalization as Search, *Artificial Intelligence* **18**, no. 2, 1982.

T. Mitchell, Machine Learning, The McGraw-Hill Companies, 1997.

If every chain in the subsumption order has a minimal and a maximal element, a version space can be described by sets of most specific  $S(VS)$  and most general  $G(VS)$  elements:

$$G(VS) := \text{MIN}(VS) := \{c \in VS \mid \neg \exists c_1 \in VS c_1 > c\},$$

$$S(VS) := \text{MAX}(VS) := \{c \in VS \mid \neg \exists c_1 \in VS c < c_1\}.$$

# Heroes of *Euryanthe*

K.M. von Weber, H. von Chezy, 1824

(Semperoper Dresden, 29.06.2006)

G \ M	sex	age	jealous	envious	brave	faithful	positive hero
Adolar	m	young	yes	no	yes	yes	+
Euryanthe	f	young	no	no	yes	yes	+
King	m	middle	no	no	yes	yes	+
Lysiart	m	middle	yes	yes	yes	no	-
Eglantine	f	young	yes	yes	no	no	-

Languages  $L_e$  and  $L_c$  are given by sets of attributes (we consider them binary, dichotomically scaled)

$M(c, e) = \text{true}$  iff  $c \subseteq e$ .

The order on classifiers  $c_1 \geq c_2 : \iff c_1 \subseteq c_2$ .

# Heroes of *Euryanthe*

Opera of K.M. von Weber, H. libretto of von Chezy, 1824

G \ M	m	f	y	ny	j	nj	e	ne	b	nb	f	nf	positive hero
Adolar	×		×		×			×	×		×		+
Euryanthe		×	×			×		×	×		×		+
King	×			×		×		×	×		×		+
Lysiart	×			×	×		×		×			×	-
Eglantine		×	×		×		×			×		×	-

Languages  $L_e$  and  $L_c$  are given by sets of attributes (we consider them binary, dichotomically scaled)

$M(c, e) = \text{true}$  iff  $c \subseteq e$ .

The order on classifiers  $c_1 \geq c_2 : \iff c_1 \subseteq c_2$ .

# Version space for *Euryanthe*

G \ M	sex	age	jealous	envious	brave	faithful	<b>positive hero</b>
Adolar	m	young	yes	no	yes	yes	+
Euryanthe	f	young	no	no	yes	yes	+
King	m	middle	no	no	yes	yes	+
Lysiart	m	middle	yes	yes	yes	no	-
Eglantine	f	young	yes	yes	no	no	-

VS = {{faithful}, {not envious}, {brave, faithful}, {not envious, faithful}, {not envious, brave}, {not envious, brave, faithful}}.

# Boundary sets of the version space

G \ M	sex	age	jealous	envious	brave	faithful	positive hero
Adolar	m	young	yes	no	yes	yes	+
Euryanthe	f	young	no	no	yes	yes	+
King	m	middle	no	no	yes	yes	+
Lysiart	m	middle	yes	yes	yes	no	-
Eglantine	f	young	yes	yes	no	no	-

VS = {{faithful}, {not envious}, {brave, faithful}, {not envious, faithful}, {not envious, brave}, {not envious, brave, faithful}}.

GVS = {{faithful}, {not envious}}

SVS = {{not envious, brave, faithful}}

# Boundary sets of the version space: Interpretation

G \ M	sex	age	jealous	envious	brave	faithful	positive hero
Adolar	m	young	yes	no	yes	yes	+
Euryanthe	f	young	no	no	yes	yes	+
King	m	middle	no	no	yes	yes	+
Lysiart	m	middle	yes	yes	yes	no	-
Eglantine	f	young	yes	yes	no	no	-

VS = {{faithful}, {not envious}, {brave, faithful}, {not envious, faithful}, {not envious, brave}, {not envious, brave, faithful}}.

GVS = {{faithful}, {not envious}}: **smallest sufficient conditions for being “positive”**

SVS = {{not envious, brave, faithful}}: **Gestalt of a romantic hero**

# Version spaces in terms of Galois connections

Formal context  $(E, C, I)$ :

- $E$  is the set of examples containing disjoint sets of observed positive and negative examples of a target attribute:  $E \supseteq E_+ \cup E_-$ ,  $E_+ \cap E_- = \emptyset$ ;
- $C$  is the set of classifiers;
- relation  $I$  corresponds to the matching predicate  $M(c, e)$ : for  $c \in C$ ,  $e \in E$  the relation  $eIc$  holds iff  $M(c, e) = 1$ ;
- $\bar{I}$  is the complementary relation:  $e\bar{I}c$  holds iff  $M(c, e) = 0$ .

**Утверждение.**

$$VS(E_+, E_-) = E_+^I \cap E_-^{\bar{I}}.$$

# Corollary: Merging version spaces

H. Hirsh, Generalizing Version Spaces, *Machine Learning* **17**, 5-46, 1994.

For fixed  $L_c$ ,  $L_e$ ,  $M(c, e)$  and two sets  $E_{+1}, E_{-1}$  and  $E_{+2}, E_{-2}$  of positive and negative examples one has

$$\text{VS}(E_{+1} \cup E_{+2}, E_{-1} \cup E_{-2}) = \text{VS}(E_{+1}, E_{-1}) \cap \text{VS}(E_{+2}, E_{-2}).$$

**Important application:** One can construct the version space  $\text{VS}(E_+, E_-)$  by intersecting “individual” version spaces related to each positive example:

$$\text{VS}(E_+, E_-) = \bigcap_{e \in E_+} \text{VS}(\{e\}, E_-).$$

# Corollary: Merging version spaces

H. Hirsh, Generalizing Version Spaces, *Machine Learning* **17**, 5-46, 1994.

For fixed  $L_c$ ,  $L_e$ ,  $M(c, e)$  and two sets  $E_{+1}, E_{-1}$  and  $E_{+2}, E_{-2}$  of positive and negative examples one has

$$\text{VS}(E_{+1} \cup E_{+2}, E_{-1} \cup E_{-2}) = \text{VS}(E_{+1}, E_{-1}) \cap \text{VS}(E_{+2}, E_{-2}).$$

**Proof.** By the property  $(A \cup B)' = A' \cap B'$ ,

$$\begin{aligned} \text{VS}(E_{+1} \cup E_{+2}, E_{-1} \cup E_{-2}) &= (E_{+1} \cup E_{+2})^I \cap (E_{-1} \cup E_{-2})^{\bar{I}} = \\ &E_{+1}^I \cap E_{+2}^I \cap E_{-1}^{\bar{I}} \cap E_{-2}^{\bar{I}} = (E_{+1}^I \cap E_{-1}^{\bar{I}}) \cap (E_{+2}^I \cap E_{-2}^{\bar{I}}) = \\ &\text{VS}(E_{+1}, E_{-1}) \cap \text{VS}(E_{+2}, E_{-2}). \end{aligned}$$

# More corollaries: Classifications and closed sets

**Утверждение.** The set of all 100%-classified examples defined by the version space  $VS(E_+, E_-)$  is given by

$$(E_+^I \cap E_-^{\bar{I}})^I.$$

# More corollaries: Classifications and closed sets

**Утверждение.** The set of all 100%-classified examples defined by the version space  $VS(E_+, E_-)$  is given by

$$(E_+^I \cap E_-^{\bar{I}})^I.$$

Interpretation of a closed set of examples:

**Утверждение.** If  $E_+^{II} = E_+$  and  $E_- = \emptyset$ , then there cannot be any 100%-classified undetermined example.

# More corollaries: Classifications and closed sets

**Утверждение.** The set of all 100%-classified examples defined by the version space  $VS(E_+, E_-)$  is given by

$$(E_+^I \cap E_-^{\bar{I}})^I.$$

Interpretation of a closed set of examples:

**Утверждение.** If  $E_+^{II} = E_+$  and  $E_- = \emptyset$ , then there cannot be any 100%-classified undetermined example.

**Утверждение.** The set of examples that are classified positively by at least one element of the version space  $VS(E_+, E_-)$  is given by

$$E \setminus (E_+^I \cap E_-^{\bar{I}})^{\bar{I}}.$$

# Classifier semilattices

**Утверждение.** If the classifiers, ordered by subsumption, form a complete semilattice, then the version space is a complete subsemilattice for any sets of examples  $E_+$  and  $E_-$ .

We return to this later when we consider pattern structures

B. Ganter and S. O. Kuznetsov, Pattern Structures and Their Projections, *Proc. 9th Int. Conf. on Conceptual Structures, ICCS'01*, G. Stumme and H. Delugach, Eds., Lecture Notes in Artificial Intelligence, **2120**, 2001, pp. 129-142.

# Hypotheses vs. version spaces

[Ganter, Kuznetsov 2003]

**Definition** A positive example  $e_+$  is hopeless iff  $(e_+)'' \cap E_- \neq \emptyset$

**Interpretation:**  $e_+$  has a negative counterpart  $e_- \in E_-$  such that every classifier which matches  $e_+$  also matches  $e_-$ .

**Theorem 1.** Suppose that the classifiers are given by subsets of the set  $M$  for some context  $(G, M, I)$

Then the following are equivalent:

1. The version space  $VS(E_+, E_-)$  is not empty.
2.  $(E_+)'' \cap E_- = \emptyset$ .
3. There are no hopeless positive examples and there is a unique minimal positive hypothesis  $h_{\min}$ . In this case,  $h_{\min} = (E_+)'$ , and the version space is an order ideal (wrt to generality order) of the powerset  $\mathcal{P}(h_{\min})$ ,  $SVS = h_{\min}$ .

# Hypotheses vs. version spaces

$B \subseteq M$  is a **proper (positive) predictor** if

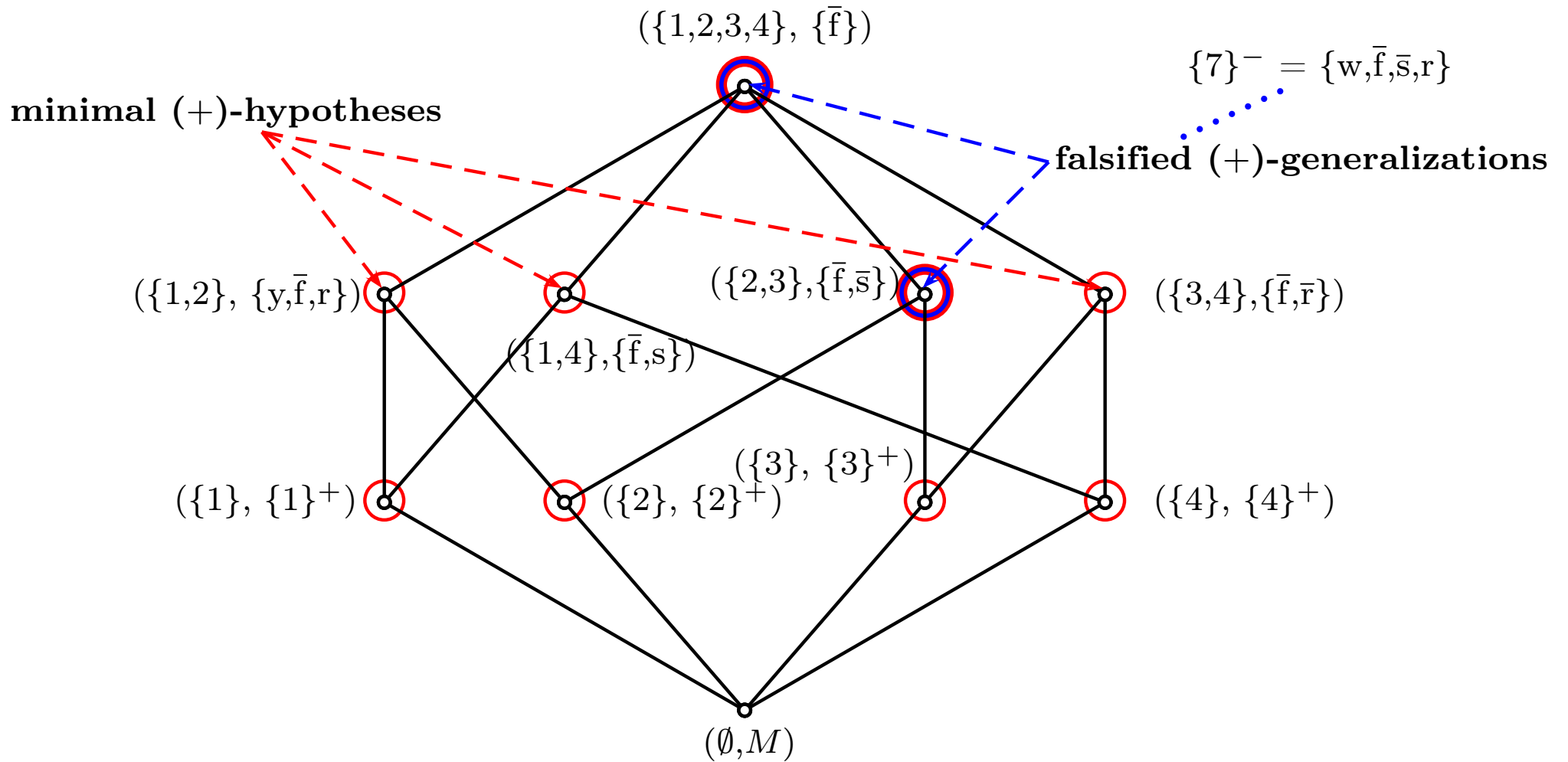
$$B' \cap E_+ \neq \emptyset, \quad B' \cap E_- = \emptyset, \quad \forall C \subset B \quad C' \cap E_- \neq \emptyset$$

**Theorem 2.** *Let  $E_+, E_-$  be sets of positive and negative examples,  $\mathcal{H}_m^+(E_+, E_-)$ ,  $PP_+(E_+, E_-)$  denote sets of minimal positive hypotheses and proper positive predictors, respectively. Then*

$$\mathcal{H}_m^+(E_+, E_-) = \min_{\subseteq} \bigcup_{F_+ \subseteq E_+} S(\text{VS})(F_+, E_-)$$

$$PP_+(E_+, E_-) = \bigcup_{e_+ \in E_+} G(\text{VS})(E_+, E_-)$$

# Positive Concept Lattice



	G \ M	w	y	g	b	f	$\bar{f}$	s	$\bar{s}$	r	$\bar{r}$	fruit
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	-
6	egg	×				×		×			×	-
7	tennis ball	×					×		×	×		-
8	mango			×			×	×			×	$\tau$

# Computing a Version Space

We order the set  $G = E_+ \cup E_- \cup E_\tau$  of all examples as follows:

$E_+$	$E_-$	$E_\tau$
$p_1, \dots, p_{max}$	$n_1, \dots, n_{max}$	$\tau_1, \dots, \tau_{max}$

The following notation is adapted from the standard formulation of the **NextClosure** algorithm:

- For  $A, B \subseteq G$  and  $i \in G$  define

$$A <_i B : \iff i \in B, i \notin A, \text{ and } (j \in A \iff j \in B \text{ for all } j < i).$$

- For  $A \subseteq G$  and  $i \in G, i \notin A$ , define

$$A \oplus i := (\{j \in A \mid j < i\} \cup \{i\})''.$$

# An algorithm

Consider a context  $(E_+ \cup E_-, M, I)$ . If classifiers are given by subsets of  $M$ , then the version space can be computed as follows:

1. **If**  $(E_+)'' \cap E_- \neq \emptyset$  **then** the version space is empty **else**
2. The first element is  $h_{\min} := (E_+)'$ ;
3. **If**  $h$  is an element of the version space, **then** the “next” element is  $h_{\text{next}} := (A \oplus i)'$ , where  $A := h'$ , and  $i$  is the largest element that is greater than  $n_{\max}$  and that satisfies  $A <_i A \oplus i$ .

# Outline

1. Lattices in Machine Learning
2. Learning implications and association rules
3. JSM-hypotheses
4. Decision trees
5. Version spaces
- 6. Learning with Pattern Structures**
7. Conclusions

# Pattern Structures

[Ganter, Kuznetsov 2001]

$(G, \underline{D}, \delta)$  is a **pattern structure** if

- $G$  is a set (“set of objects”);
- $\underline{D} = (D, \sqcap)$  is a meet-semilattice;
- $\delta : G \rightarrow D$  is a mapping;
- the set  $\delta(G) := \{\delta(g) \mid g \in G\}$  generates a complete subsemilattice  $(D_\delta, \sqcap)$  of  $(D, \sqcap)$ .

Possible origin of  $\sqcap$  operation:

- A set of objects  $G$ , each with description from  $P$ ;
- Partially ordered set  $(P, \leq)$  of “descriptions” ( $\leq$  is a “more general than” relation);
- The (distributive) lattice of order ideals of the ordered set  $(P, \leq)$ .

# Pattern Structures

**Pattern structure** is a tuple  $(E, (C, \sqcap), \delta)$ , where

- $E$  is a set of “examples”,
- $\delta$  is a mapping of examples to “descriptions”,  $\delta : E \rightarrow C$ ,  $\delta(E) := \{\delta(e) \mid e \in E\}$ .

The subsumption order:  $c \sqsubseteq d \iff c \sqcap d = c$ .

**Derivation operators:**

$$A^\diamond := \sqcap_{e \in A} \delta(e) \text{ for } A \subseteq E$$

$$c^\diamond := \{e \in E \mid c \sqsubseteq \delta(e)\} \text{ for } c \in C.$$

A pair  $(A, c)$  is a **pattern concept** of  $(E, (C, \sqcap), \delta)$  if

$$A \subseteq E, c \in C, A^\diamond = c, c^\diamond = A$$

$A$  is **extent** and  $c$  is **pattern intent**.

# Pattern-based Hypotheses

$E_+$  and  $E_-$  are positive and negative examples for some target attribute,

$$E_+, E_- \subseteq E, \quad E_+ \cap E_- = \emptyset$$

A **positive hypothesis**  $h$  is a pattern intent of  $(E_+, (C, \sqcap), \delta)$  not subsumed by any negative example:

$$h^\diamond \cap E_- = \emptyset \quad \text{and} \quad \exists_{A \subseteq E_+} A^\diamond = h.$$

# An application

Symbiosis between some fungi or micro-organisms and trees positively influences temperate forest productivity.

 **INRA**: How does symbiosis work at the cellular level ?

- *Laccaria bicolor* is a model fungus to study symbiosis in temperate forest ecosystem: genome sequenced
- Search for particular biological processes
- Find the functions of unknown genes involved in symbiosis



By analysing Gene Expression Data (GED)  
obtained with microarrays

# An example of Gene Expression Data (GED)

A **gene expression data (GED)** is a table with

- genes in rows
- biological situations in columns
- table entries: “expression value” of a gene in row for the situation in column.  
A whole row: **gene expression profile** of a gene, or its behaviour through situations.

	$s_1$	$s_2$	$s_3$
$g_1$	5	7	6
$g_2$	6	8	4
$g_3$	4	8	5
$g_4$	4	9	8
$g_5$	5	8	5

A group of genes having a similar expression profile interact together within the same biological process.

# Similarity Operation $\sqcap$ for Intervals

**Interval pattern structures for analyzing GED [M.Keytoue et al., ICFCA'2009]**

Given  $a_1, b_1, a_2, b_2 \in \mathbb{R}$ ,

$$[a_1, b_1] \sqcap [a_2, b_2] = [\min(a_1, a_2), \max(b_1, b_2)] \quad (1)$$

The similarity  $\sqcap$  of two intervals is the smallest interval “containing” them.

$$[4, 5] \sqcap [5, 8] = [4, 8]$$

$$[3, 4] \sqcap [1, 2] = [1, 4]$$

$\sqcap$  is idempotent, commutative and associative.

# Interval Ordering

**Partial order  $\sqsubseteq$  on intervals** Given  $i_1 = [a_1, b_1]$  and  $i_2 = [a_2, b_2]$  two intervals, the order on intervals is given by:

$$\begin{aligned} i_1 &\sqsubseteq i_2 \\ \Leftrightarrow i_1 \cap i_2 &= i_1 \\ \Leftrightarrow [a_1, b_1] \cap [a_2, b_2] &= [a_1, b_1] \\ \Leftrightarrow a_1 \leq a_2 \text{ and } b_1 &\geq b_2 \end{aligned} \tag{2}$$

Intuitively smaller intervals subsume larger intervals “containing” them.

$$\begin{aligned} [4, 8] &\sqsubseteq [6, 8] \text{ as } 4 \leq 6 \text{ and } 8 \geq 8 \\ [2, 5] &\not\sqsubseteq [1, 8] \text{ as } 2 \not\leq 1 \text{ or } 5 \not\geq 8 \end{aligned}$$

# Interval Pattern

For representing GED we need multi-dimensional descriptions of objects, where each dimension represents a situation.

An **interval pattern**  $e$  is a  $p$ -dimensional vector of intervals.

$$e = \langle [a_i, b_i] \rangle_{i \in [1, p]} \quad (3)$$

Each gene expression profile is an interval pattern,

$$\langle [5, 5], [7, 7], [6, 6] \rangle, \text{ with } p = 3,$$

where  $[a, a]$  stays for any number  $a$ .

# Similarity of Interval Patterns and Order

For two interval patterns  $e$  and  $f$ ,

$$e = \langle [a_i, b_i] \rangle_{i \in [1, p]}$$

$$f = \langle [c_i, d_i] \rangle_{i \in [1, p]}$$

$\sqcap$  operation and order of interval patterns

$$\begin{aligned} e \sqcap f &= \langle [a_i, b_i] \rangle_{i \in [1, p]} \sqcap \langle [c_i, d_i] \rangle_{i \in [1, p]} \\ e \sqcap f &= \langle [a_i, b_i] \sqcap [c_i, d_i] \rangle_{i \in [1, p]} \end{aligned} \tag{4}$$

$$e \sqsubseteq f$$

$$\Leftrightarrow e \sqsubseteq f = e \tag{5}$$

$$\Leftrightarrow [a_i, b_i] \sqsubseteq [c_i, d_i], \forall i \in [1, p]$$

# Interval pattern structures. An Example

	$s_1$	$s_2$	$s_3$
$g_1$	5	7	6
$g_2$	6	8	4
$g_3$	4	8	5
$g_4$	4	9	8
$g_5$	5	8	5

$$G = \{g_1, \dots, g_5\}$$

$$\delta(g_1) = \langle [5, 5], [7, 7], [6, 6] \rangle$$

$$D = \{\delta(g_1), \dots, \delta(g_5)\}$$

$(D, \sqcap)$  is a meet-semi lattice of interval patterns.

# Interval pattern structures. An Example.

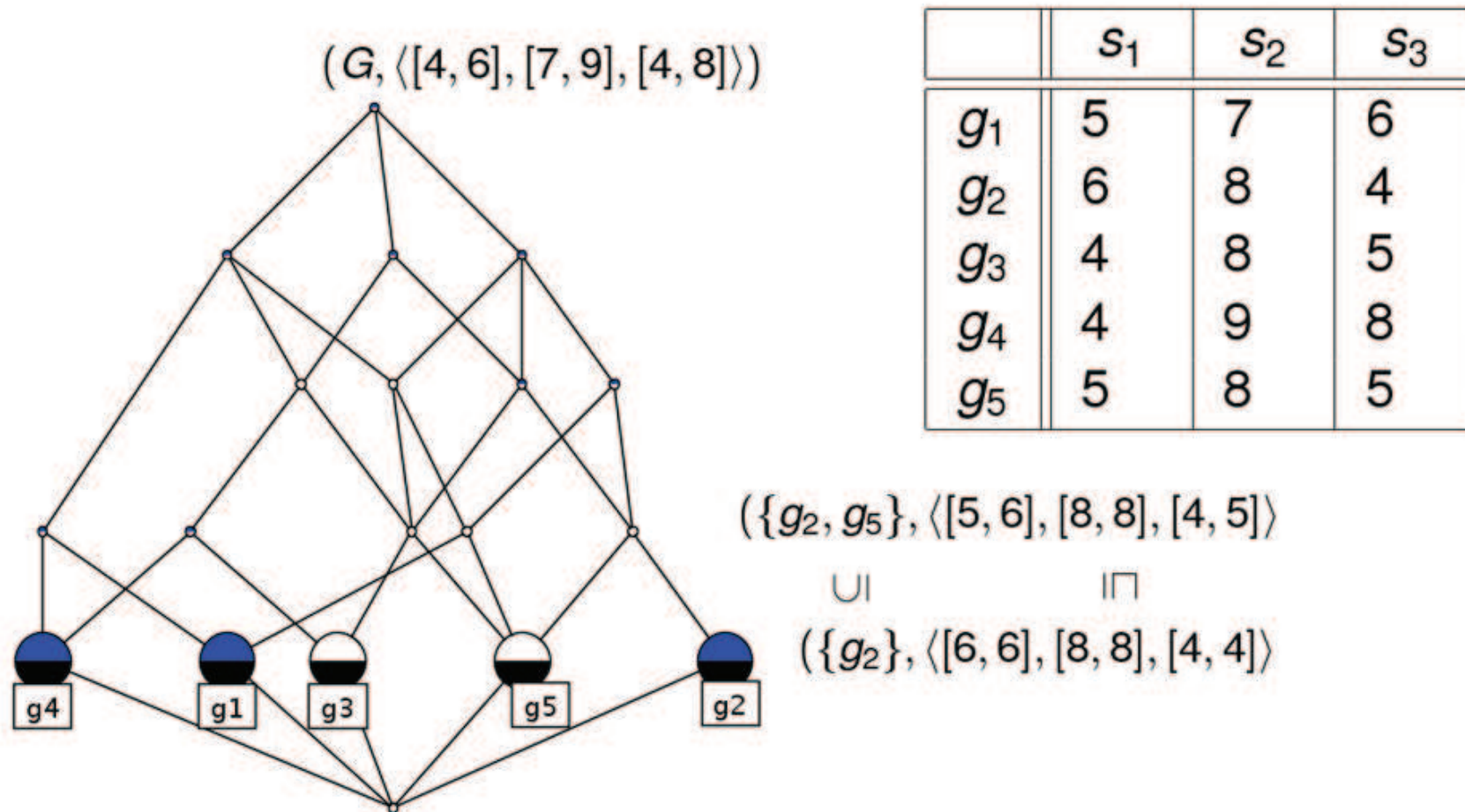
	$s_1$	$s_2$	$s_3$
$g_1$	5	7	6
$g_2$	6	8	4
$g_3$	4	8	5
$g_4$	4	9	8
$g_5$	5	8	5

$$\begin{aligned} \{g_4, g_5\}^\diamond &= \bigcap_{\delta \in \{g_4, g_5\}} \delta(g) = \delta(g_4) \sqcap \delta(g_5) = \langle [4, 4], [9, 9], [8, 8] \rangle \sqcap \langle [5, 5], [8, 8], [5, 5] \rangle \\ &= \langle [4, 4] \sqcap [5, 5], [9, 9] \sqcap [8, 8], [8, 8] \sqcap [5, 5] \rangle = \langle [4, 5], [8, 9], [5, 8] \rangle \end{aligned}$$

$$\{[4, 5], [8, 9], [5, 8]\}^\diamond = \{g \in G \mid \langle [4, 5], [8, 9], [5, 8] \rangle \sqsubseteq \delta(g)\} = \{g_3, g_4, g_5\}$$

Then the pair  $(\{g_3, g_4, g_5\}, \langle [4, 5], [8, 9], [5, 8] \rangle)$  is a pattern concept, FCA algorithms need slight adaptations

# Pattern concept lattice



# Interestingness of an interval pattern

## Not all pattern concepts are interesting

- Biologists look for homogeneous groups of genes described by patterns with “small” intervals.
- Recall: The pattern of the top concept is composed of largest intervals.

## A solution: introduce a $max\_size$ parameter

Given  $d = \langle [a_i, b_i] \rangle_{i \in [1, p]}$ , two constraints are defined:

- $\exists i \in [1, p] (b_i - a_i) \leq max\_size$
- $\forall i \in [1, p] (b_i - a_i) \leq max\_size.$

## Another solution

When computing  $\sqcap$ , replace any  $[a, b]$  with  $b - a > max\_size$  by a \*-value. Then,  $* \sqsubseteq [a, b] \Leftrightarrow * \sqcap [a, b] = *$  for any  $[a, b]$ .

# Data



Gene expression data<sup>a</sup> of *Laccaria bicolor* composed of:

- 22,294 genes
- 5 biological situations reflecting cells of the organism in various stages of its biological cycle: free living mycelium, symbiotic tissues and fruiting bodies.

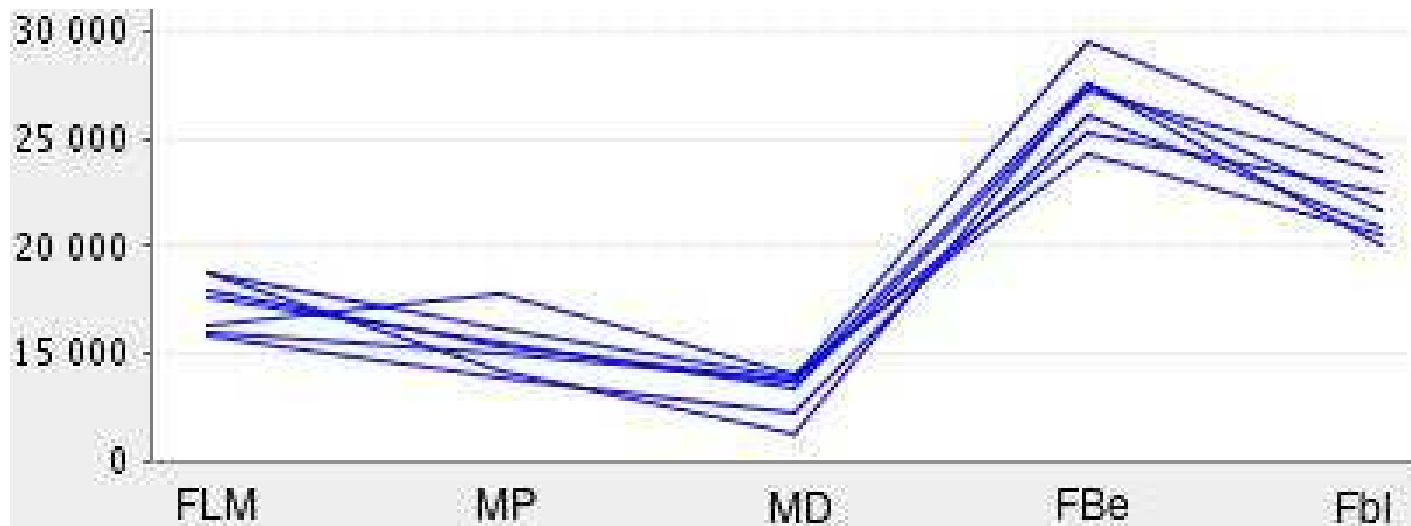
---

<sup>a</sup>[Martin et al., Nature 2008]

# An example of biological result

## An example of extracted pattern

- Genes present a high expression level in the fruit-body situations
- Some of these genes encode metabolic enzymes in remobilization of fungal resources towards the new organ in development
- Other genes are unknown but specific to *Laccaria Bicolor*: it requires biological experiments



# Interval Patterns vs. Conceptual Scaling

- Scaling gets a formal context  $(G, M, I)$  from a many-valued context  $(G, N, W, J)$ ;
- Turns many-valued attributes of  $N$  in binary attributes of  $M$ ;
- **Interordinal scale** allows to represent intervals of attribute values
- Each attribute is scaled separately.
- $W_s$  is the set of attribute values of many-valued attribute  $s$ .
- **Interordinal scale**  $I_{W_s} = (W_s, W_s, \leq) | (W_s, W_s, \geq)$ ,  $W_s$  is the set of attribute values of attribute  $s$ .
- The **Interordinal scale** creates  $2|W_s| - 1$  binary attributes for each many-valued attribute  $s$ .

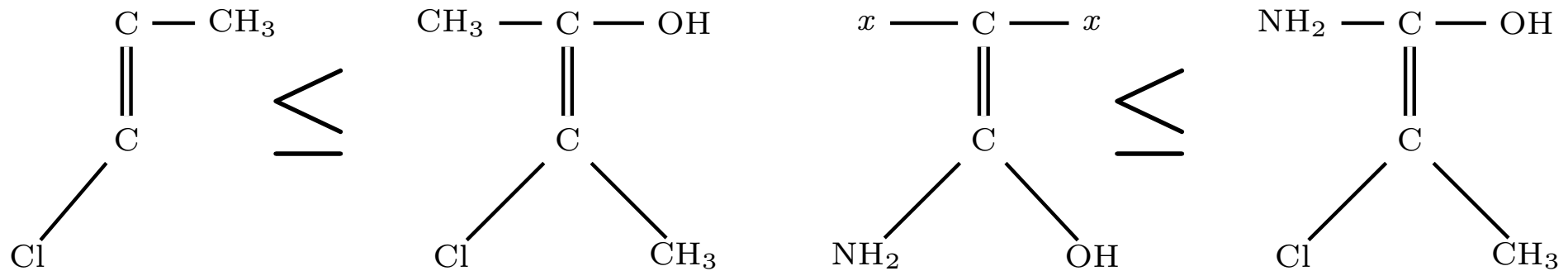
# Pattern structures on labeled graphs

$\Gamma_1 := ((V_1, l_1), E_1)$  **dominates**  $\Gamma_2 := ((V_2, l_2), E_2)$  or  $\Gamma_2 \leq \Gamma_1$

if there exists a one-to-one mapping  $\varphi: V_2 \rightarrow V_1$  such that

- respects edges:  $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$ ,
- fits under labels:  $l_2(v) \leq l_1(\varphi(v))$ .

**Example:**



vertex labels are unordered

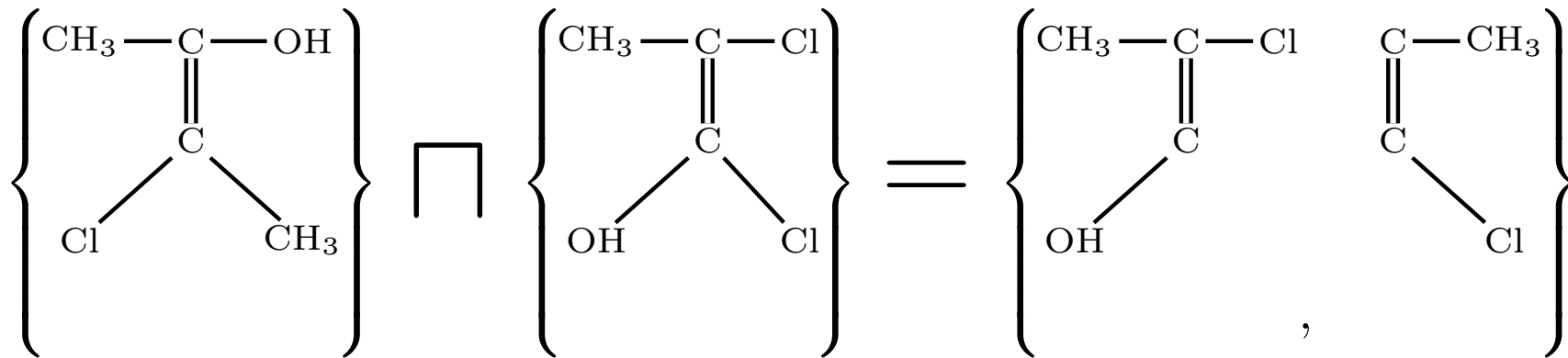
$x \preceq A$  for any vertex label  $A \in \mathcal{L}$

# Semilattice on graph sets

$$\{X\} \sqcap \{Y\} := \{Z \mid Z \leq X, Y, \quad \forall Z_* \leq X, Y \quad Z_* \not\leq Z\}$$

= The set of all maximal common subgraphs of  $\Gamma_1$  and  $\Gamma_2$ .

**Example:**



# Meet of graph sets

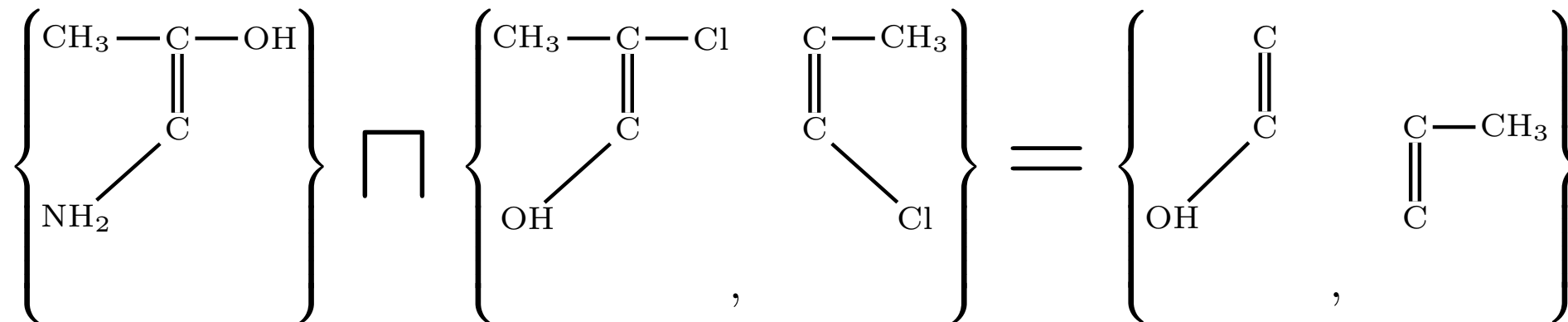
For sets of graphs

$$\mathcal{X} = \{X_1, \dots, X_k\} \text{ and } \mathcal{Y} = \{Y_1, \dots, Y_n\}$$

$$\mathcal{X} \sqcap \mathcal{Y} := \text{MAX} (\cup_{i,j} (\{X_i\} \sqcap \{Y_j\}))$$

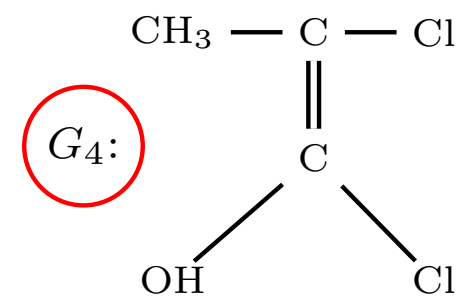
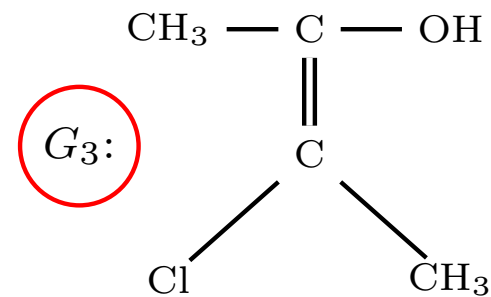
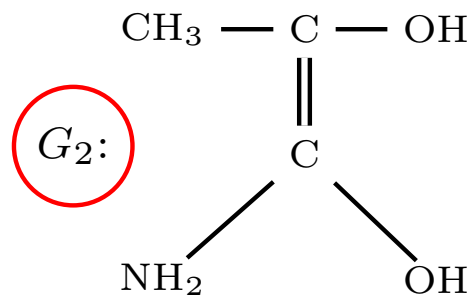
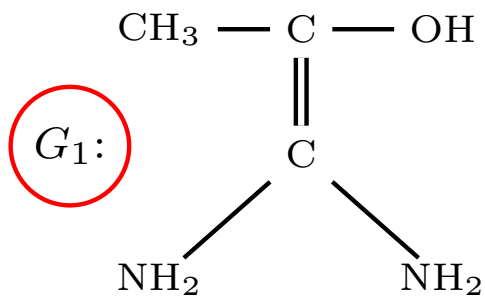
$\sqcap$  is idempotent, commutative, and associative.

**Example:**

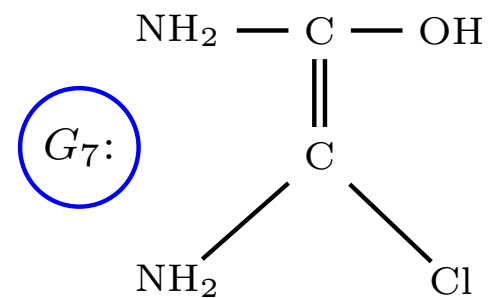
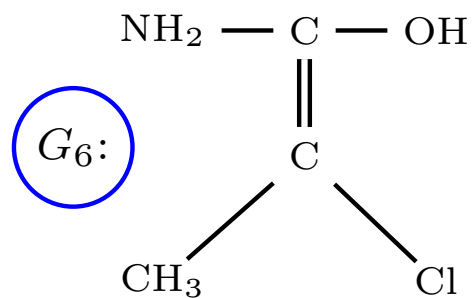
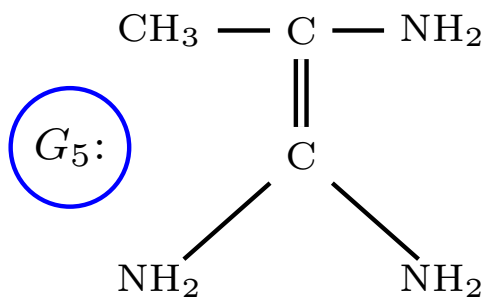


# Examples

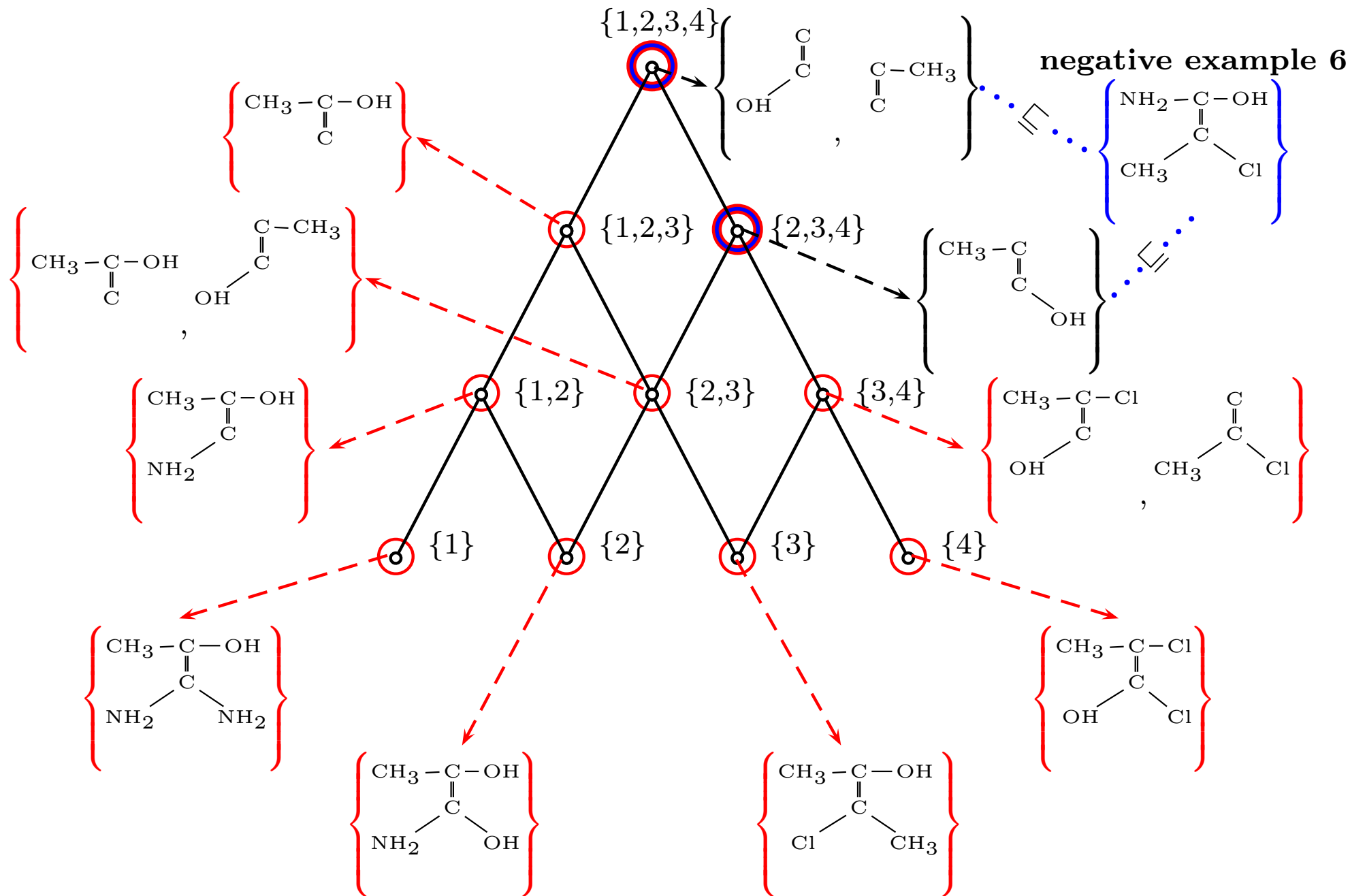
## Positive examples:



## Negative examples:

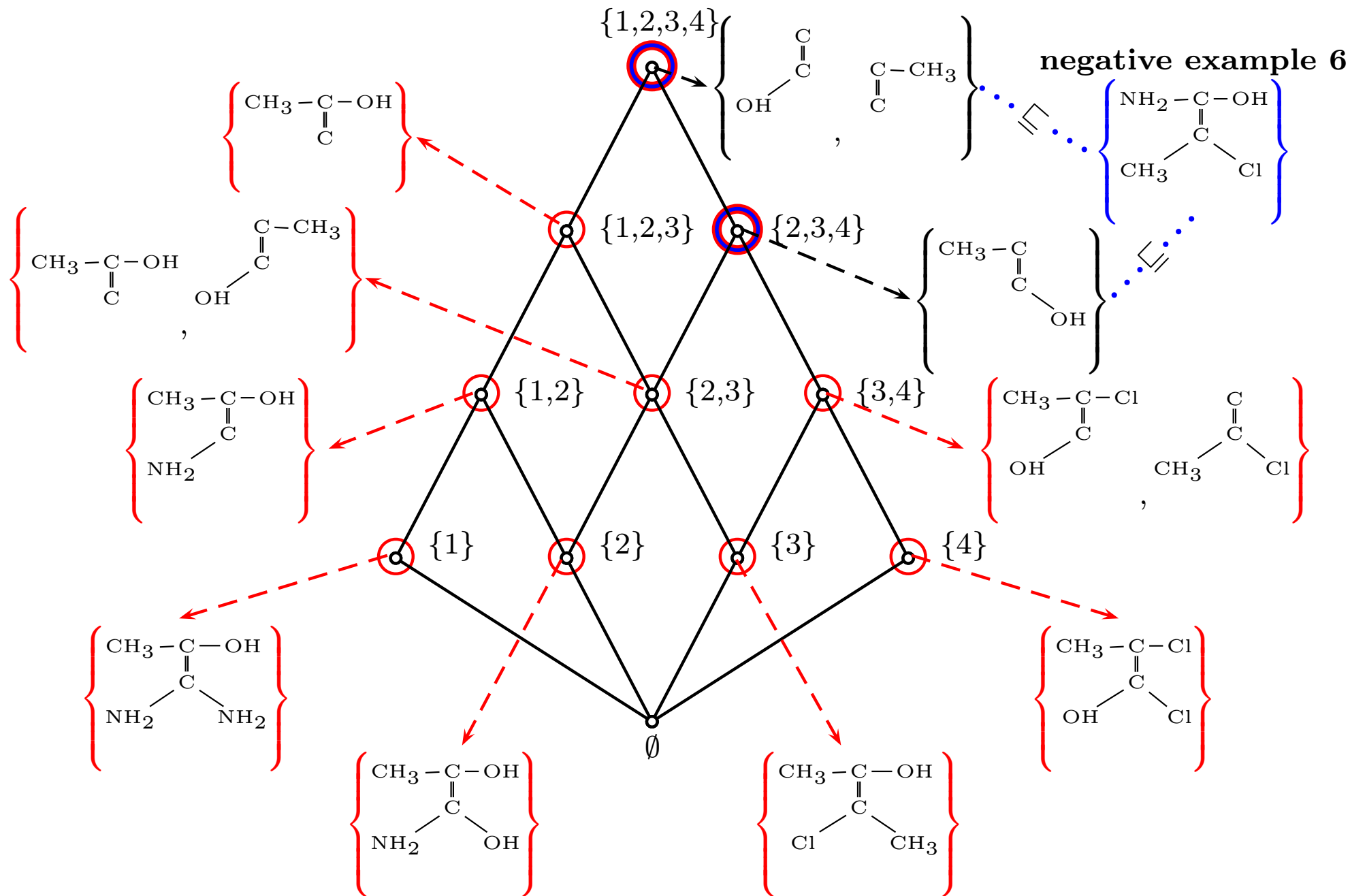


# Positive (semi)lattice



positive examples 1, 2, 3, 4

# Positive lattice



positive examples 1, 2, 3, 4

# Projections as Approximation Tool

**Motivation:** Complexity of computations in  $(G, \underline{D}, \delta)$ , e.g.,  
SUBGRAPH ISOMORPHISM, i.e., testing  $\leq$  for graphs is NP-complete.

$\psi$  is **projection** (kernel operator) on an ordered set  $(D, \sqsubseteq)$  if  $\psi$  is

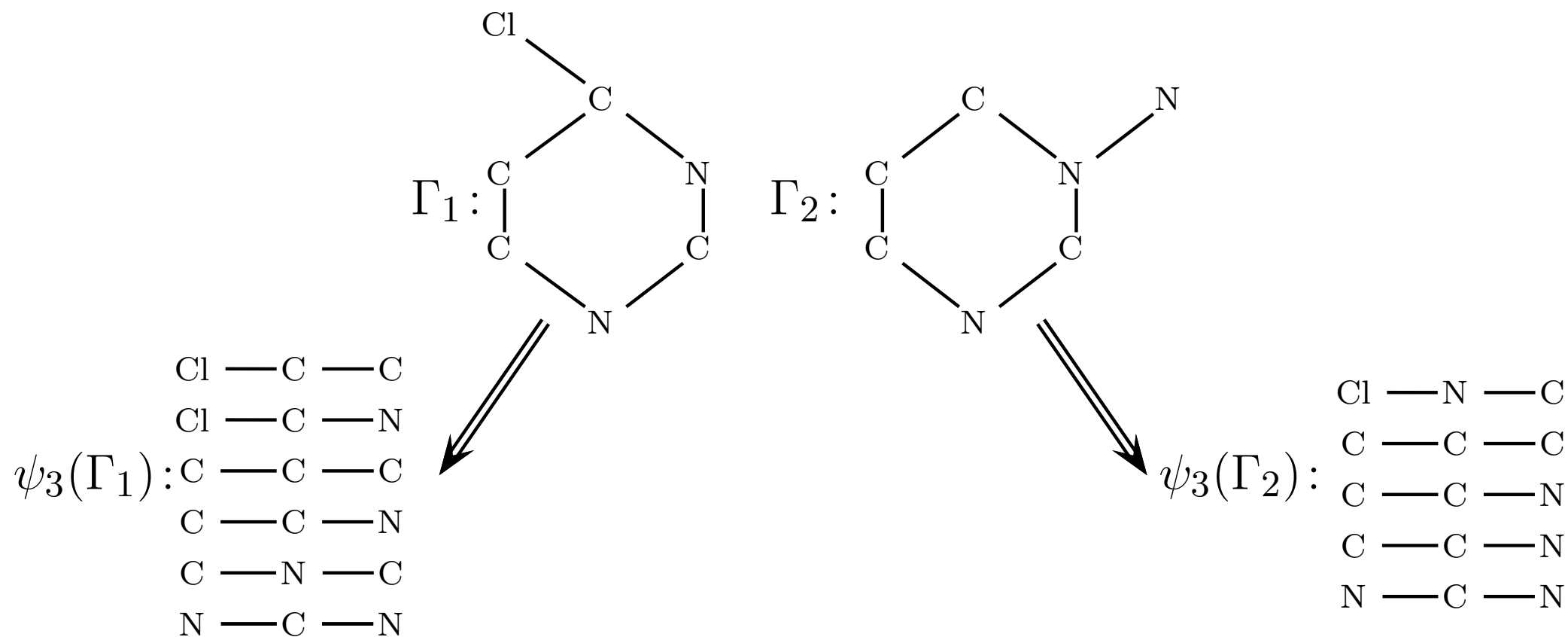
**monotone:** if  $x \sqsubseteq y$ , then  $\psi(x) \sqsubseteq \psi(y)$ ,

**contractive:**  $\psi(x) \sqsubseteq x$ ,

**idempotent:**  $\psi(\psi(x)) = \psi(x)$ .

# Projections as Approximation Tool

**Example.** A projection for labeled graphs:  $\psi_n(\Gamma)$  takes  $\Gamma$  to the set of its  $n$ -chains not dominated by other  $n$ -chains. Here  $n = 3$ .

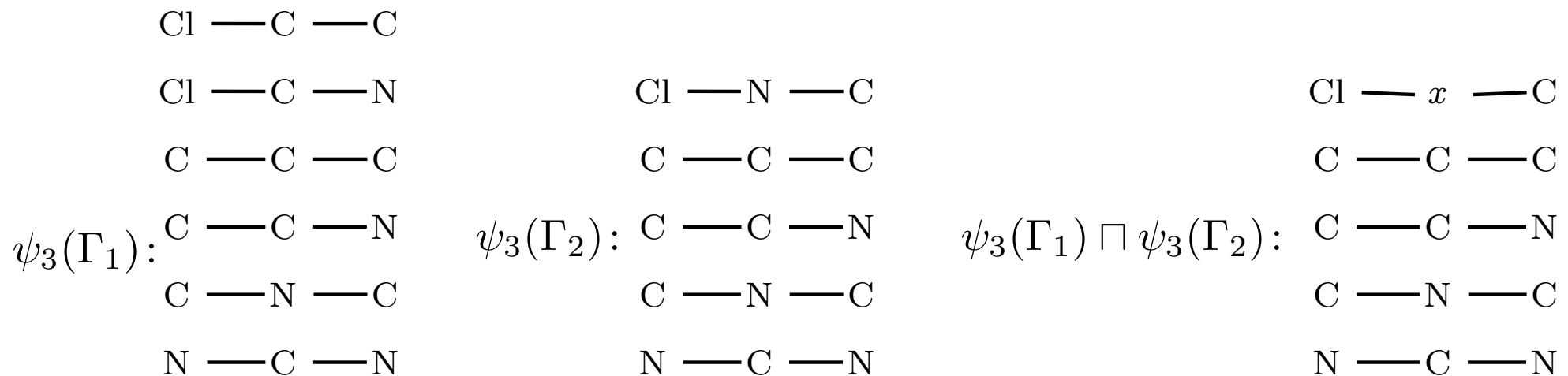


# Property of projections

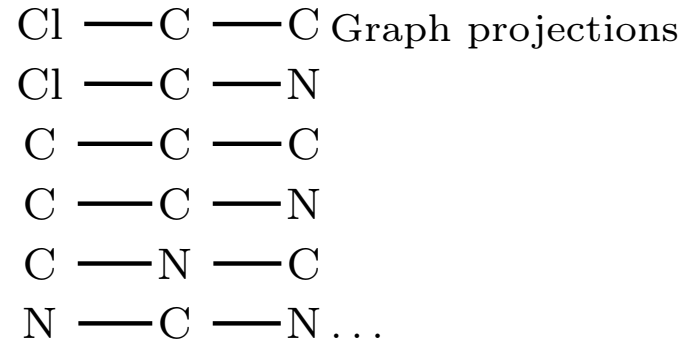
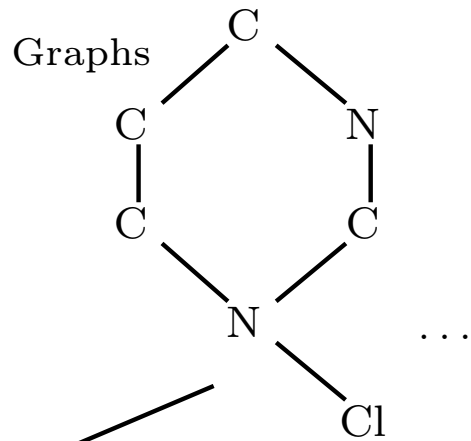
Any projection of a complete semilattice  $(D, \sqcap)$  is  $\sqcap$ -preserving, i.e., for any  $X, Y \in D$

$$\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y).$$

**Example.** A projection for labeled graphs:  $\psi_n(\Gamma)$  takes  $\Gamma$  to the set of its  $n$ -chains not dominated by other  $n$ -chains. Here  $n = 3$ .



# Projections and Representation Context



Lattice of graph sets

Lattice of graph sets projections

Basic Theorem of FCA

Basic Theorem of FCA

Representation Context

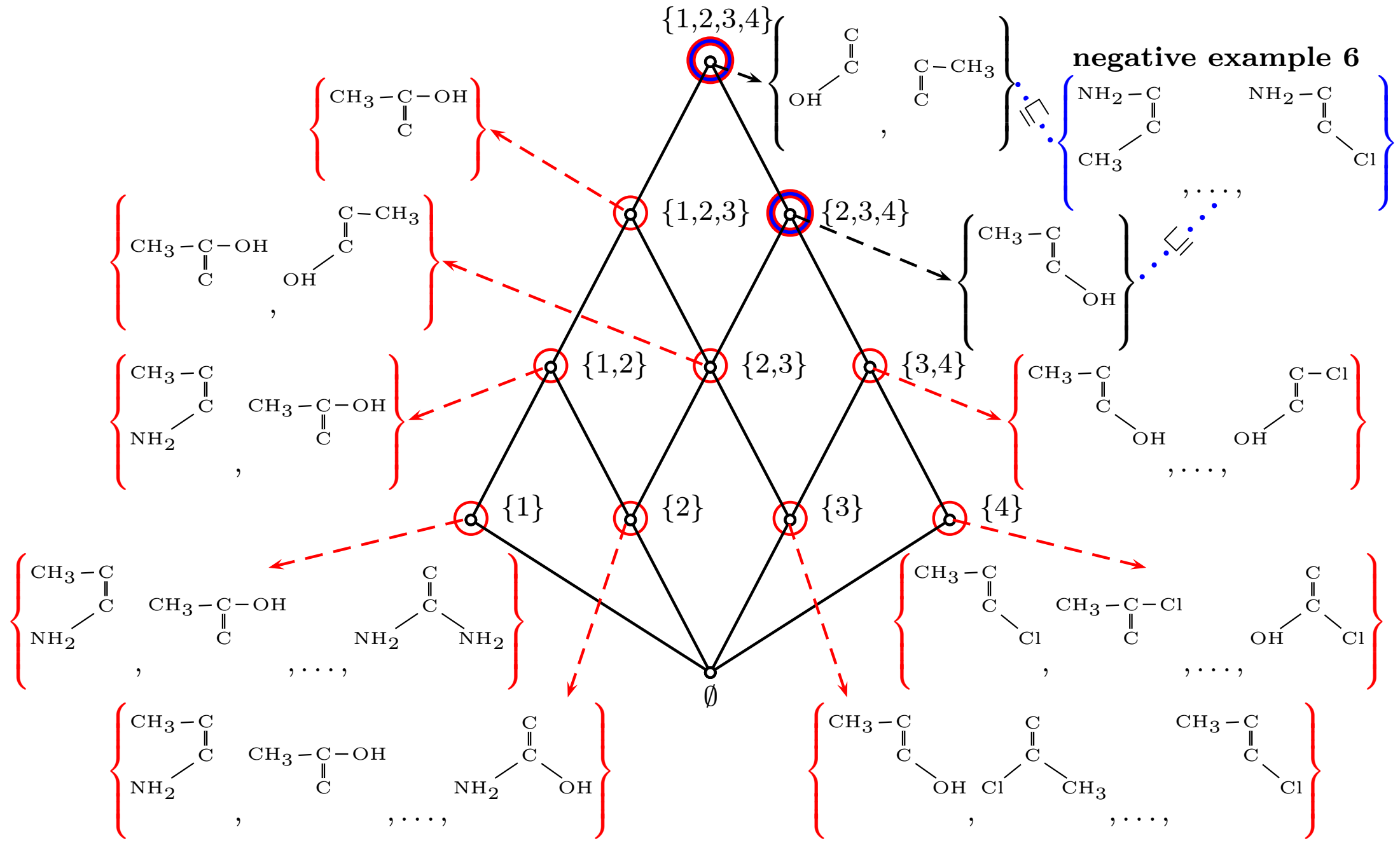
Representation Subcontext

G \ M	a	b	c	d	e	f	goal
1	x	x	x		x		+
2	x	x	x	x	x	x	+
3	x	x		x			+
4	x	x		x		x	+
5	x		x		x		-
6	x	x	x	x		x	-
7		x	x	x	x	x	-



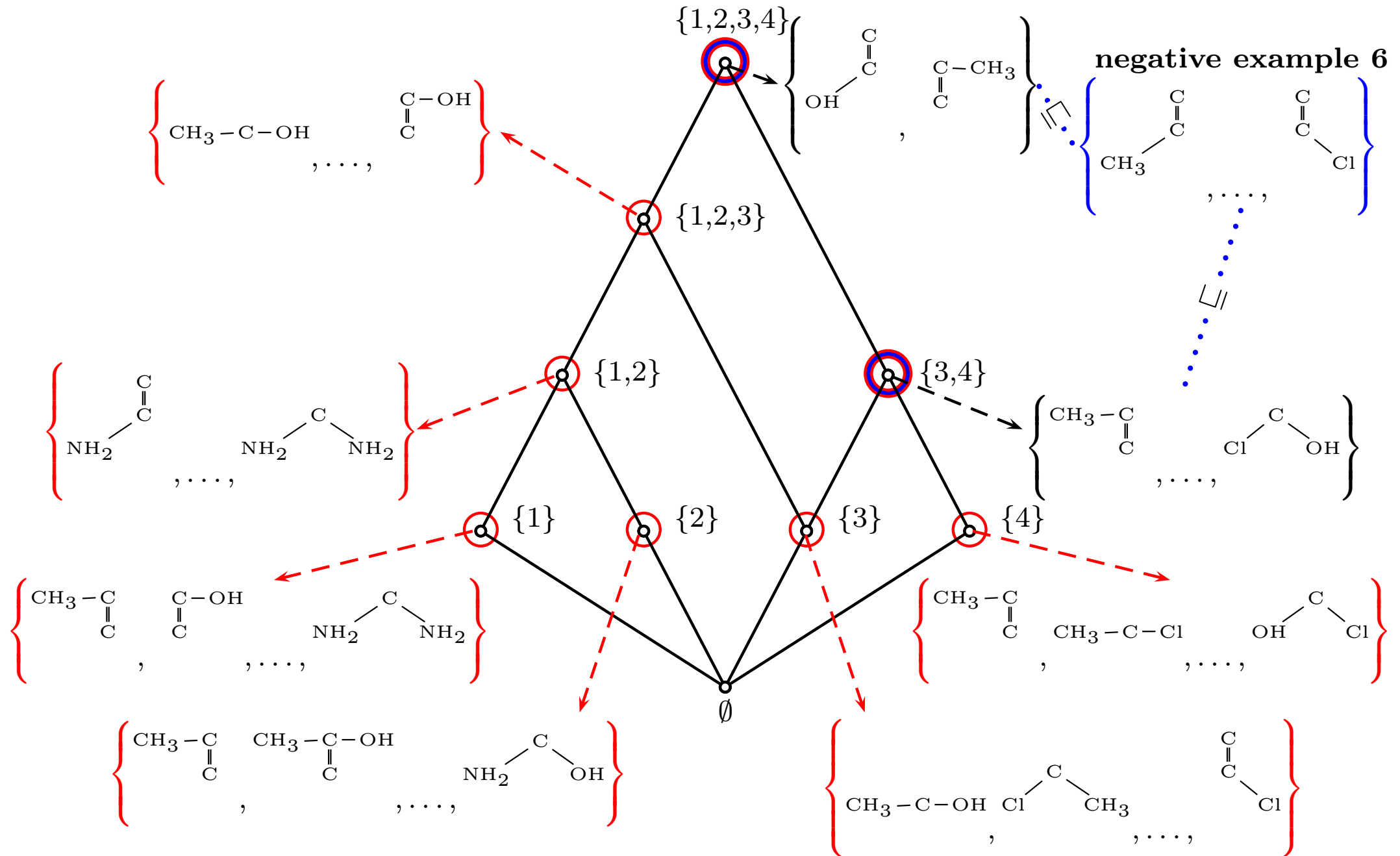
G \ M	a	b	c	d	e	f	goal
1	x	x	x		x		+
2	x	x	x	x	x	x	+
3	x	x		x			+
4	x	x		x		x	+
5	x		x		x		-
6	x	x	x	x		x	-
7		x	x	x	x	x	-

# 4-Projections

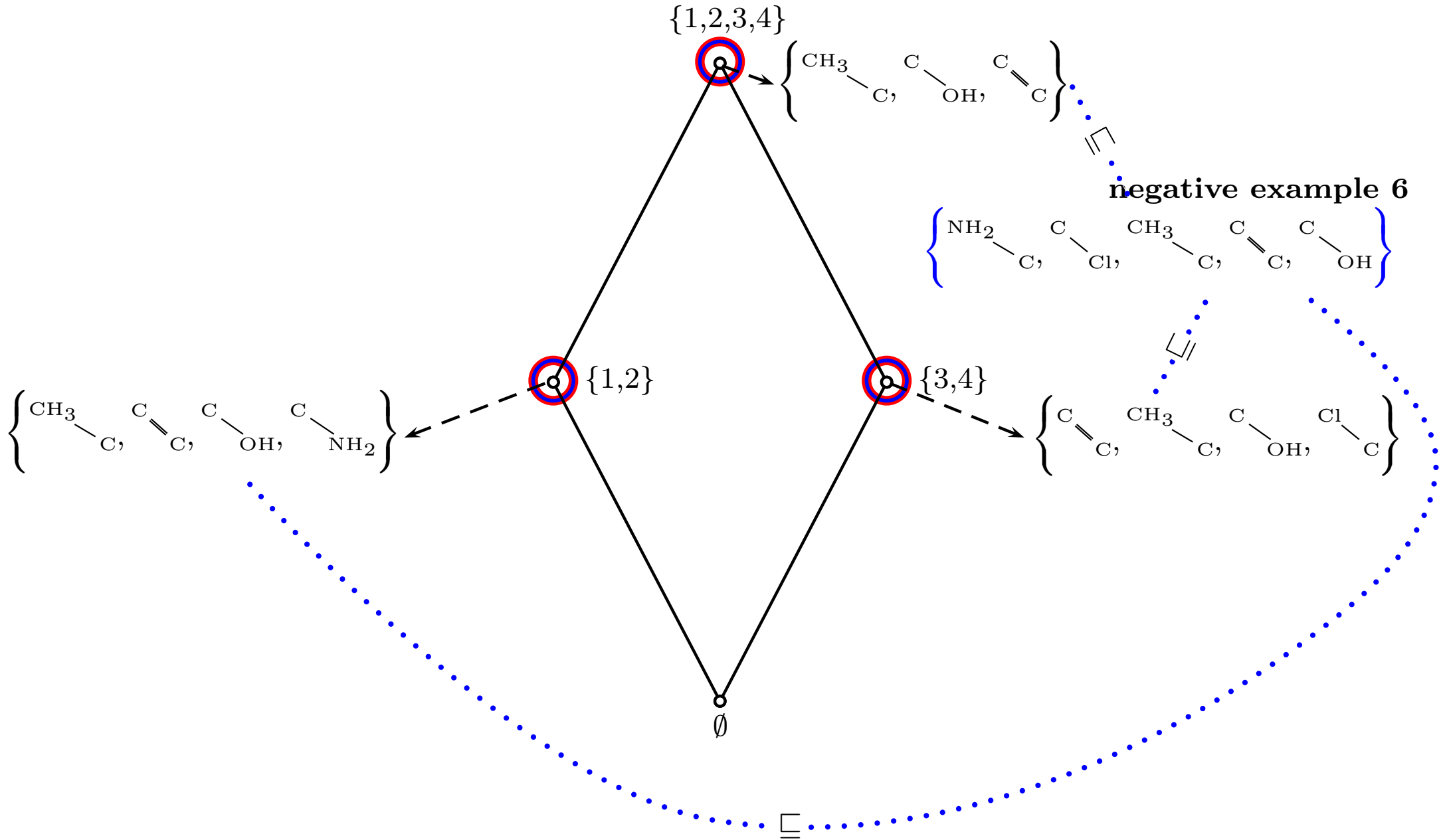


positive examples 1, 2, 3, 4

# 3-Projections



# 2-Projections



# Hypotheses vs. version spaces

$d \in D$  is a **proper (positive) predictor** if

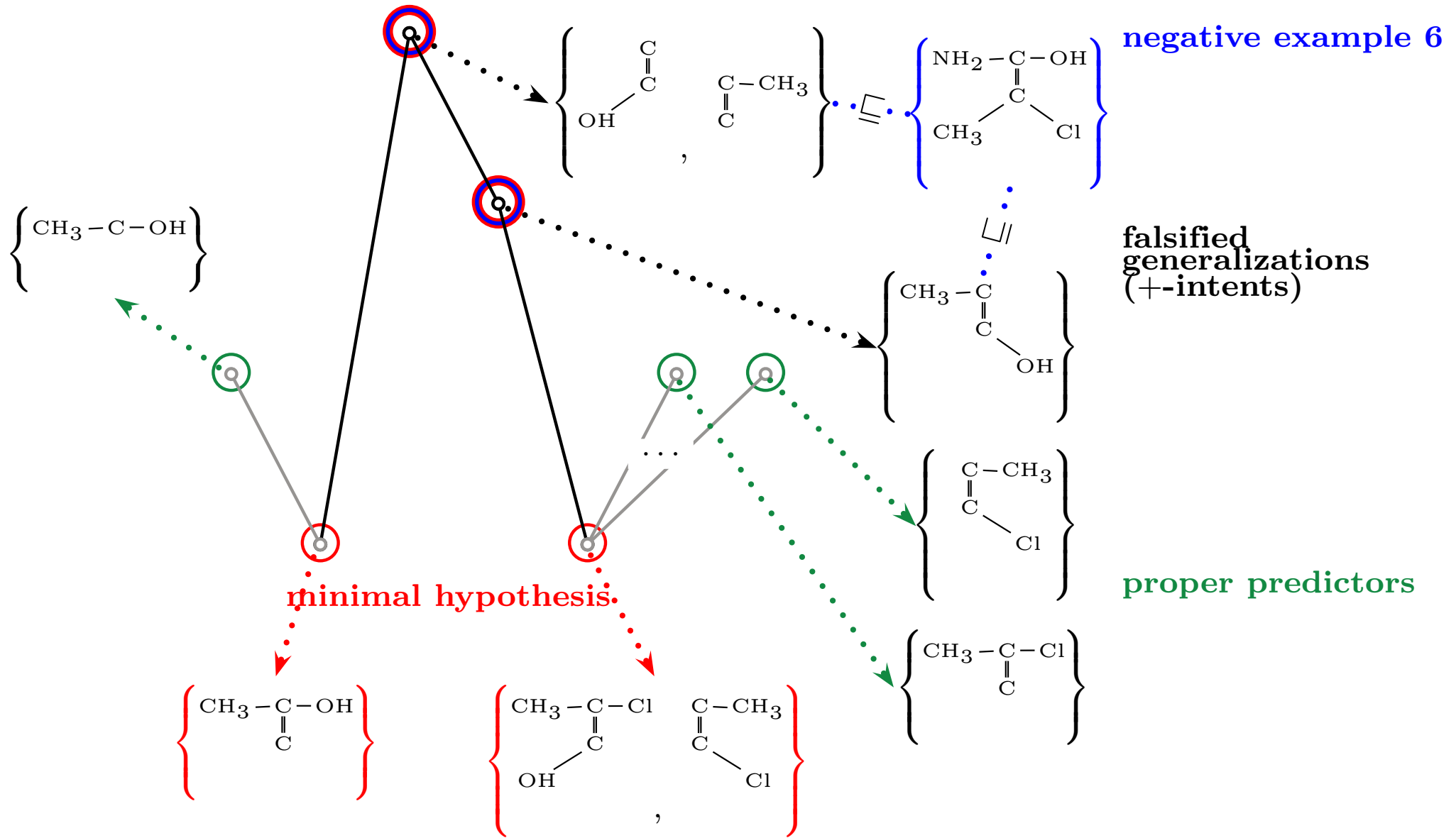
$$d^\diamond \cap E_+ \neq \emptyset, \quad d^\diamond \cap E_- = \emptyset, \quad \forall q \sqsubset d \quad q^\diamond \cap E_- \neq \emptyset$$

**Theorem 2.** *Let  $E_+, E_-$  be sets of positive and negative examples,  $\mathcal{H}_m^+(E_+, E_-)$ ,  $PP_+(E_+, E_-)$  denote sets of minimal positive hypotheses and proper positive predictors, respectively. Then*

$$\mathcal{H}_m^+(E_+, E_-) = \min_{\sqsubseteq} \bigcup_{F_+ \subseteq E_+} S(\text{VS})(F_+, E_-)$$

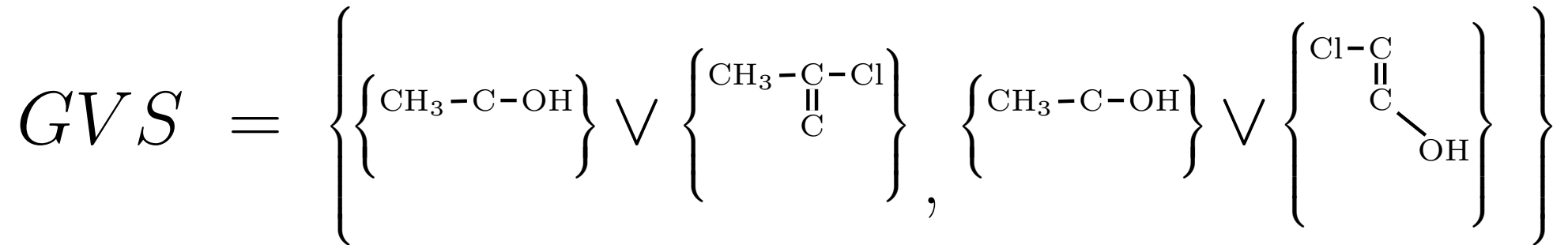
$$PP_+(E_+, E_-) = \bigcup_{e_+ \in E_+} G(\text{VS})(E_+, E_-)$$

# Proper predictors

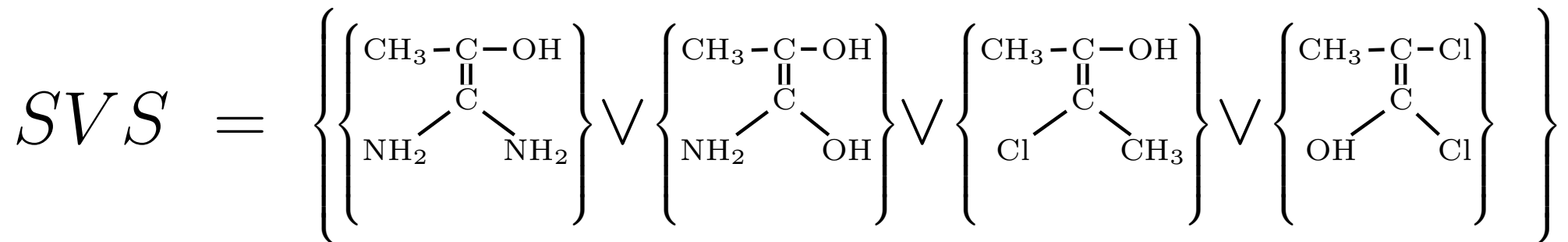


# Example. Boundaries of the Version Space

If disjunction is not allowed, then  $VS = \emptyset$ . If disjunction is allowed, then



equivalent to  $PP_+$ , but generally can be of size  $\left(\frac{|PP_+|}{k}\right)^k$ , where  $k \in \{1, \dots, |PP_+|\}$ .



trivial generalization: disjunction of all positive examples.

# Projection types used in the experiments

We used several types of projections of labeled graph sets that are natural in chemical applications:

- *k-chain* projection: a set of graphs  $X$  is taken to the set of all chains with  $k$  vertices that are subgraphs of at least one graph of the set  $X$ ;
- *k-vertex* projection: a set of graphs  $X$  is taken to the set of all subgraphs with  $k$  vertices that are subgraphs of at least one graph of the set  $X$ ;
- *k-cycles* projection: a set of graphs  $X$  is taken to the set of all subgraphs consisting of  $k$  adjacent cycles of a minimal cyclic basis of at least one graph of the set  $X$ .

Mixed projections (with same algebraical properties of simple projections) are also possible.

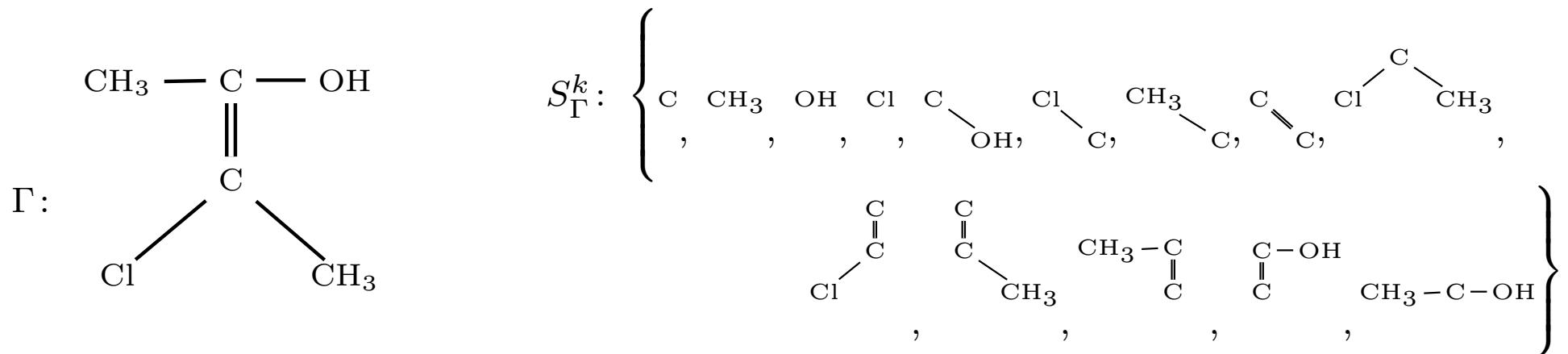
# $k$ -projections of graphs

The definition of a projection for a general semilattice can be made more precise for the case of a graph semilattice in the following way:

$S_{\Gamma}^k = \{\Gamma_* = ((V_*, l_*), (E_*, b_*)) \mid \Gamma_* \text{ is a connected graph, } \Gamma_* \leq \Gamma, |V_*| \leq k\}$  is called a  **$k$ -projection** ( $1 \leq k \leq |V|$ ) of a labeled graph  $\Gamma = ((V, l), (E, b))$ .

= The set of all connected subgraphs (up to isomorphism) of the graph  $\Gamma$  with the number of vertices not larger than  $k$ .

**Example:**  $k = 3$



## Example 2. PTC challenge: constructing a model of toxicity

[V. G. Blinova et al, 2003 // Bioinformatics, 19 (2003)]

	„mh	„ss	$P_4^{ss}$	$P_4^{mh}$	$P_5^{ss}$	$P_5^{mh}$	$P_7^{ss}$	$P_7^{mh}$
+ → +	9	9	6	1	8	2	4	2
- → +	23	17	5	1	2	2	1	1

- For the group **MR** (male rats) the representations using 5- and 7-projections lead to “new” optimal results for this group in the “posthumous analysis”. The use of the 4-projection also brings good results: the corresponding point lies above the old “ROC” curve.

	„mh	„ss	$P_4^{ss}$	$P_4^{mh}$	$P_5^{ss}$	$P_5^{mh}$	$P_6^{ss}$	$P_6^{mh}$	$P_8^{ss}$	$P_8^{mh}$
+ → +	4	3	4	3	3	3	4	3	4	3
- → +	7	0	2	3	3	1	4	1	4	1

- For the **FR** group (female rats) the results with 4-, 5-, 6-, and 8-projections are above the old ROC curve, but are not better than the old optimal models LEU3 and KWAI.

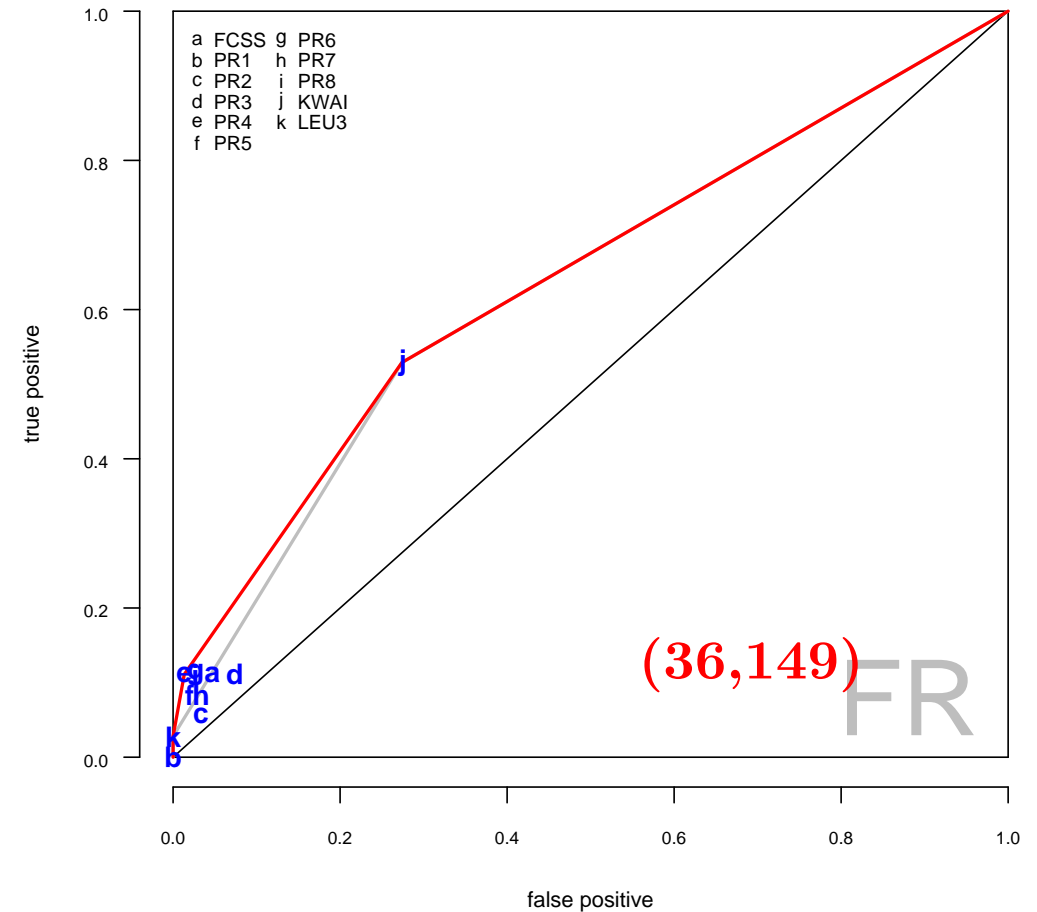
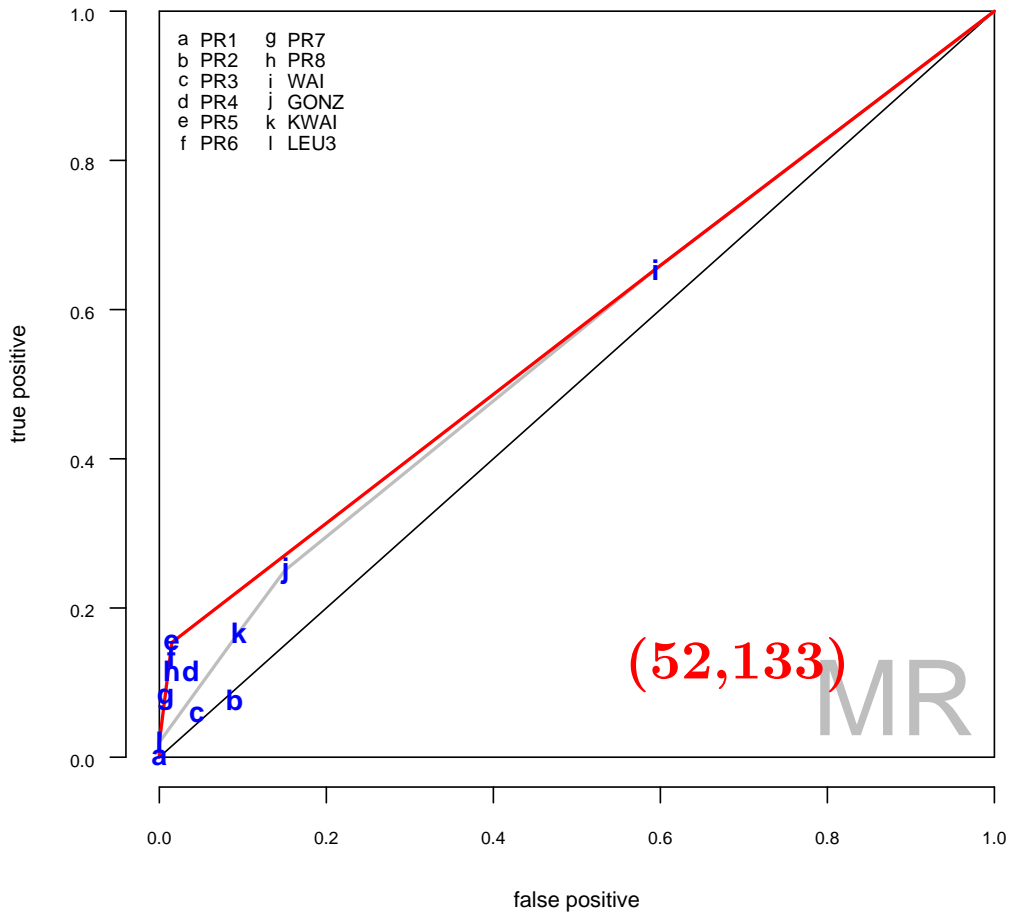
# The strategy of sequential covering

This procedure does not generate all concept-based hypotheses, but generates an irredundant covering of examples by minimal hypotheses (by a greedy algorithm).

Theoretically, this strategy has an obvious drawback: dependence on the object numbering. The obvious advantage of this strategy is its computational efficiency. In many practical cases the results of the covering strategy are very close to the results attained with the complete set of hypotheses.

# Example 2. Estimating results with ROC curves

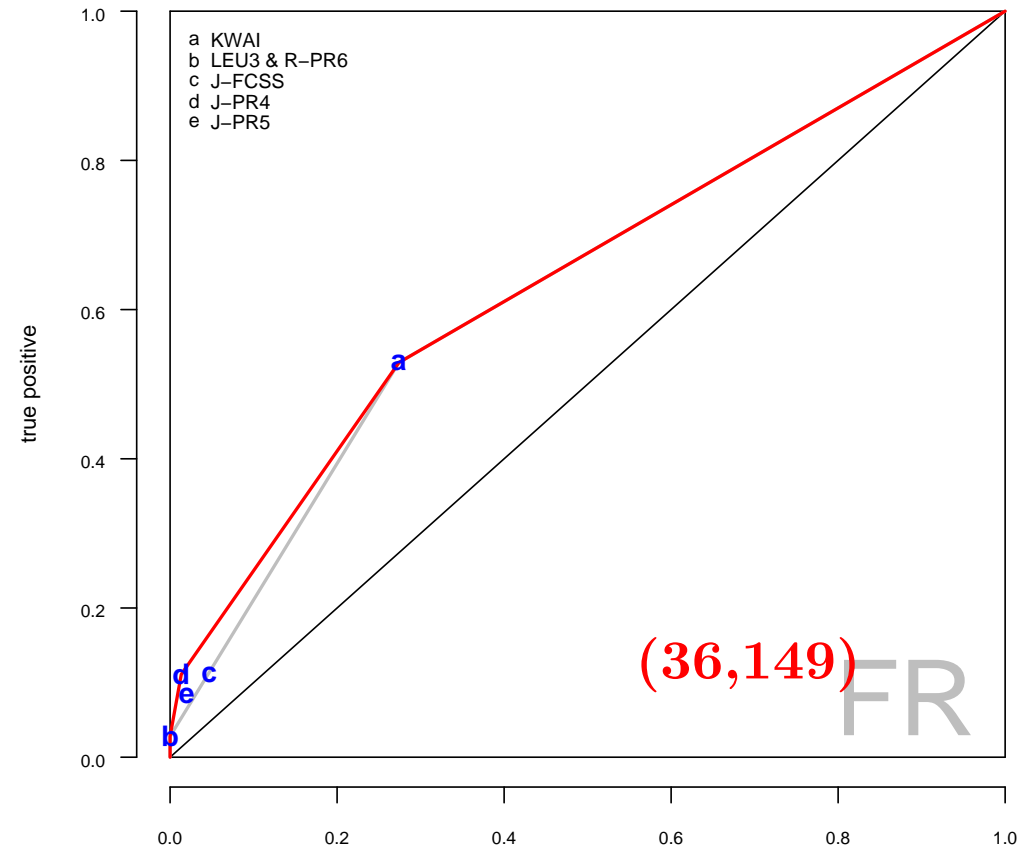
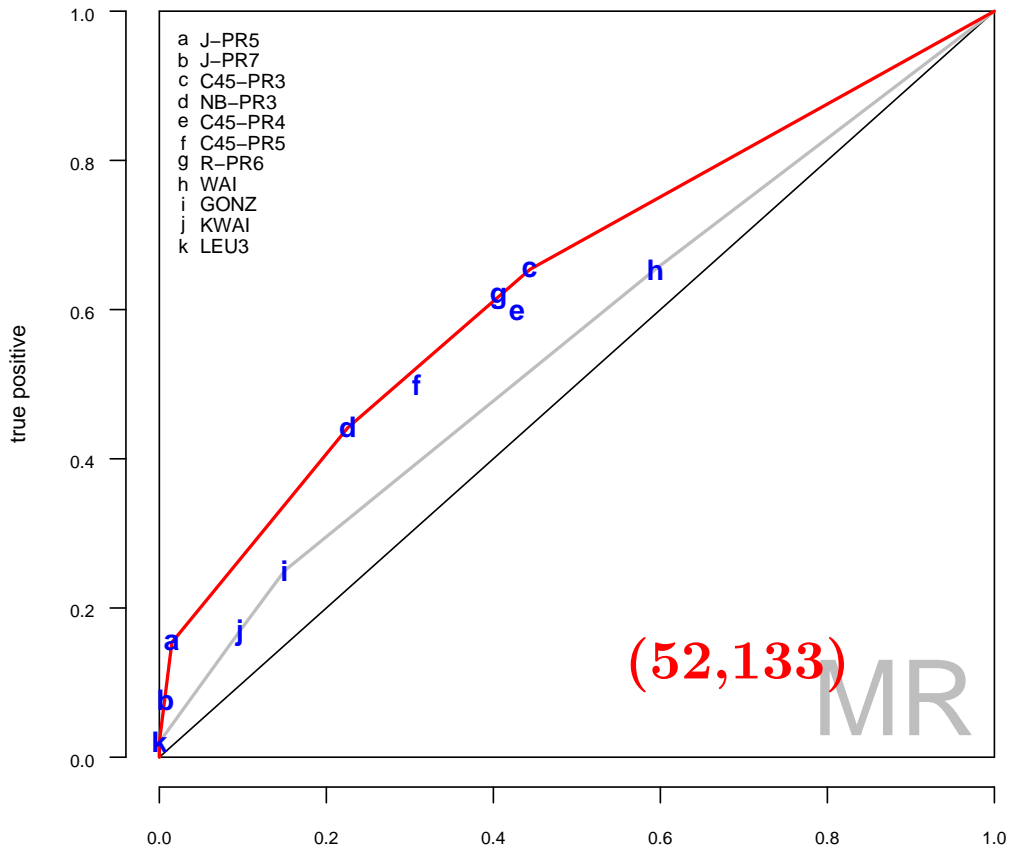
	„mh	„ss	$P_4^{SS}$	$P_4^{mh}$	$P_5^{SS}$	$P_5^{mh}$	$P_7^{SS}$	$P_7^{mh}$
+ → +	9	9	<b>6</b>	1	<b>8</b>	2	4	2
- → +	23	17	<b>5</b>	1	<b>2</b>	2	1	1



	„mh	„ss	$P_4^{SS}$	$P_4^{mh}$	$P_5^{SS}$	$P_5^{mh}$	$P_6^{SS}$	$P_6^{mh}$	$P_8^{SS}$	$P_8^{mh}$
+ → +	4	3	<b>4</b>	3	<b>3</b>	3	<b>4</b>	3	<b>4</b>	3
- → +	7	0	<b>2</b>	3	<b>3</b>	1	<b>4</b>	1	<b>4</b>	1

# Example 2a. Estimating results with ROC curves

	„mh	„ss	$P_4^{SS}$	$P_4^{mh}$	$P_5^{SS}$	$P_5^{mh}$	$P_7^{SS}$	$P_7^{mh}$
+ → +	9	9	<b>6</b>	1	<b>8</b>	2	4	2
- → +	23	17	<b>5</b>	1	<b>2</b>	2	1	1



	„mh	„ss	$P_4^{SS}$	$P_4^{mh}$	$P_5^{SS}$	$P_5^{mh}$	$P_6^{SS}$	$P_6^{mh}$	$P_8^{SS}$	$P_8^{mh}$
+ → +	4	3	<b>4</b>	3	<b>3</b>	3	<b>4</b>	3	<b>4</b>	3
- → +	7	0	<b>2</b>	3	<b>3</b>	1	<b>4</b>	1	<b>4</b>	1

# Conclusions

Concept lattices propose a unifying approach for describing important learning models:

- Attribute Exploration based on implication bases
- Generating association rules (represented by the lattice covering relation)
- Generation of JSM-hypotheses
- Induction of decision trees
- Version spaces

All the mentioned learning approaches scale to data given by complex ordered descriptions (the order staying for “more general than” or “part of” relation) within the framework of pattern structures. Pattern structures on vectors of intervals, sets of labeled graphs, and logical formulas show their usefulness in various applications.

**Thank you!**