GALACTIC : A new Plateform for Semantic Clustering

GAlois LAttices, Concept Theory, Implicational systems and Closures

Christophe Demko[1] Karell Bertet[1] Axel Jean-Caurant[1]

[1]L3I Lab, University of La Rochelle, France
prenom.nom@univ-lr.fr

Tutorial - PFIA 2026 - PlateForme Intelligence Artificielle [1]

What is GALACTIC?

Stands for

GAlois LAttices, Concept Theory, Implicational systems and Closures

Evariste Galois (1811-1831)

- French mathematician
- He died young in a pistol duel

What is GALACTIC?

- The GALACTIC project implements novel paradigms and algorithms:
  - for semantic clustering of heterogeneous and complex data
  - using a predicate-based embedding in a description space
  - supporting an interactive approach that places the analyst at the center of the analysis process
- GALACTIC is at the intersection of:
  - Pattern-mining-based hierarchical clustering
  - The theoretical framework of Formal Concept Analysis (FCA)

Outline

Clustering

Clustering

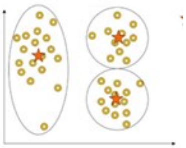
Clustering is the task of grouping a set of objects in such a way that objects in the same group/cluster are more “similar” (in some sens) to each other than to those in other groups

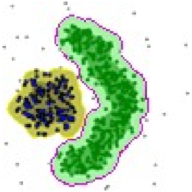
Clustering methods depends on the type of data;

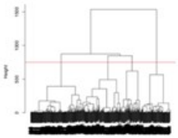
- Numerical data (distance, similarity measure)
- Categorical/discrete data (pattern, common attributes)
- Graphs (modularity, ..)

Type of clusters

- Partitioning: divise the data into k groups
  - Exclusive: each instance belongs to exactly one cluster
  - Non-exclusive: each instance may belong to multiple clusters
- Hierarchy: each group is a subgroup of its parent
  - Divisive approach (top-down): starts with a single global cluster
  - Agglomerative approach (bottom-up): starts with one cluster per instance







Clustering

Classical clustering $\{C_i\}$

object cluster C_i $\leftrightarrows$ color, (rep. object)

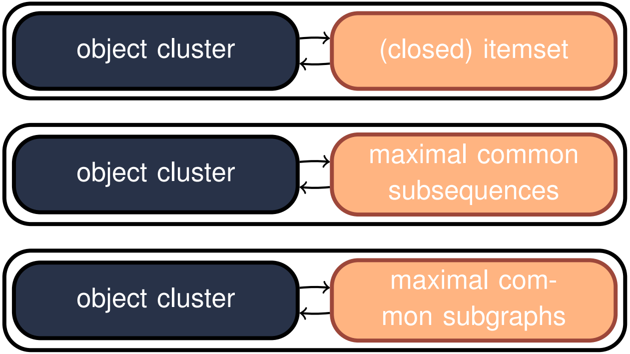
Semantic clustering $\{(C_i, P_i)\}$

object cluster C_i $\leftrightarrows$ descriptive pattern P_i

- K-means: partitionning of numerical data (distance)
- Cure: hierarchical clustering, for numerical data (distance)
- Louvain: partitionning (communities) of nodes of a graph (modularity)
- ....

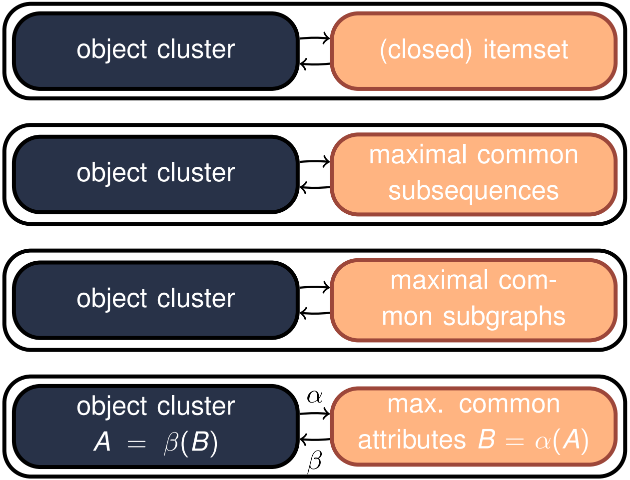
- Pattern mining: hierarchical, for categorical data, graphs and sequences
- FCA: hierarchical, for categorical data
- GALACTIC: hierarchical, for complex and heterogeneous data

Pattern Mining



- Pattern mining was first introduced for categorical data, starting with Apriori [1] and later A-close algorithms
- They are top-down hierarchical clustering methods
- They were then extended to sequence mining [2] and graph mining

Pattern Mining



Concept

- Pattern mining was first introduced for categorical data, starting with Apriori [1] and later A-close algorithms
- They are top-down hierarchical clustering methods
- They were then extended to sequence mining [2] and graph mining
- This approach can be formalized within the framework of FCA, with the (α, β) Galois connection

Formal Concept Analysis

- Formal Concept Analysis [3] (FCA) is a mathematical theory designed for applications in areas such as knowledge representation, information retrieval, data analysis and mining, and data visualization, ...
- It provides tools for understanding and visualizing data by structuring it into a hierarchy of concepts (a concept lattice) and by deriving implications between attributes (rule bases)
- These tools rely on solid theoretical foundations based on a triplet of bijective mapping between data, concept lattices, and rule bases, established through a Galois connection (α, β) between objects and binary attributes [4]

Example

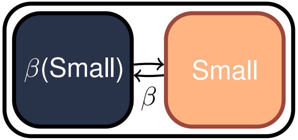
Excerpt of Animals dataset

$\alpha \nearrow\!\!\swarrow \beta$	Small	Med.	Large	Hunt	Run	Mane	Hooves
Fox		✓		✓	✓		
Dog		✓			✓		
Wolf		✓		✓	✓	✓	
Cat	✓			✓	✓		
Tiger			✓	✓	✓		
Lion			✓	✓	✓	✓	
Horse			✓		✓	✓	✓
Zebra			✓		✓	✓	✓
Cow			✓				✓

Example

Excerpt of Animals dataset

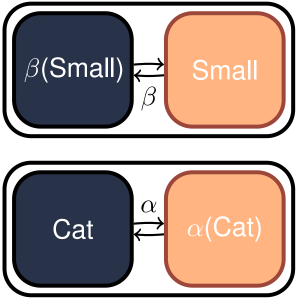
$\alpha \rightleftarrows \beta$	Small	Med.	Large	Hunt	Run	Mane	Hooves
Fox		✓		✓	✓		
Dog		✓			✓		
Wolf		✓		✓	✓	✓	
Cat	✓			✓	✓		
Tiger			✓	✓	✓		
Lion			✓	✓	✓	✓	
Horse			✓		✓	✓	✓
Zebra			✓		✓	✓	✓
Cow			✓				✓



Example

Excerpt of Animals dataset

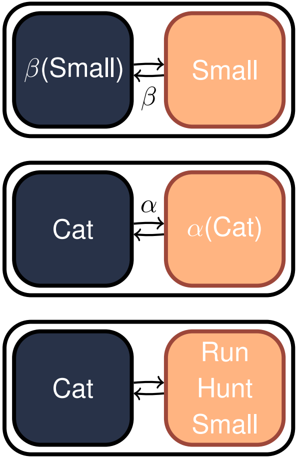
$\alpha \nearrow\!\!\swarrow \beta$	Small	Med.	Large	Hunt	Run	Mane	Hooves
Fox		✓		✓	✓		
Dog		✓			✓		
Wolf		✓		✓	✓	✓	
Cat	✓			✓	✓		
Tiger			✓	✓	✓		
Lion			✓	✓	✓	✓	
Horse			✓		✓	✓	✓
Zebra			✓		✓	✓	✓
Cow			✓				✓



Example

Excerpt of Animals dataset

$\alpha \nearrow\!\!\swarrow \beta$	Small	Med.	Large	Hunt	Run	Mane	Hooves
Fox		✓		✓	✓		
Dog		✓			✓		
Wolf		✓		✓	✓	✓	
Cat	✓			✓	✓		
Tiger			✓	✓	✓		
Lion			✓	✓	✓	✓	
Horse			✓		✓	✓	✓
Zebra			✓		✓	✓	✓
Cow			✓				✓



Concept

Definitions

Formal Context (G, M, I)

A formal context (G, M, I) is defined by a binary relation $I \subseteq G \times M$ between a set of objects G and a set of attributes M

- G stands for "Gegenstände" (objects in German) and M stands for "Merkmale" (attributes in German)
- For $a \in G$ and $b \in M$, the notation $a\,I\,b$ means that the object a has the attribute b

Galois connection (α, β)

The Galois connection (α, β) is induced by the relation I, for $A \subseteq G$ and $B \subseteq M$:

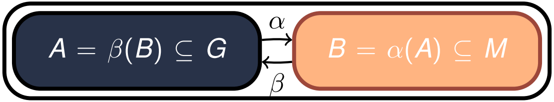
- $\alpha(A) = \{\, b \in M \mid \forall a \in A,\ I(a, b) \,\}$
- $\beta(B) = \{\, a \in G \mid \forall b \in B,\ I(a, b) \,\}$

Definitions

Formal Concept (A, B)

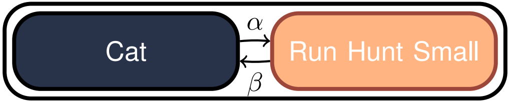
A concept is a pair (A, B) such that $A \subseteq G$, $B \subseteq M$, $A = \beta(B)$ and $B = \alpha(A)$

- A is called the extent of the concept
- B is called the intent of the concept.
- B is a closed set of attributes by the closure operator $(\alpha \circ \beta)$

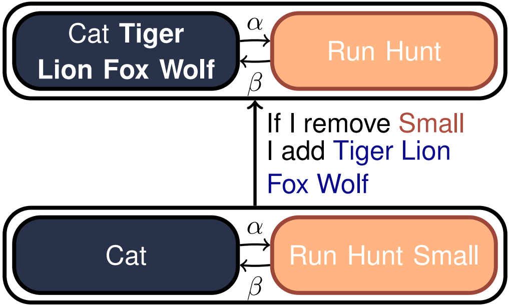


Concept $(A\ B)$

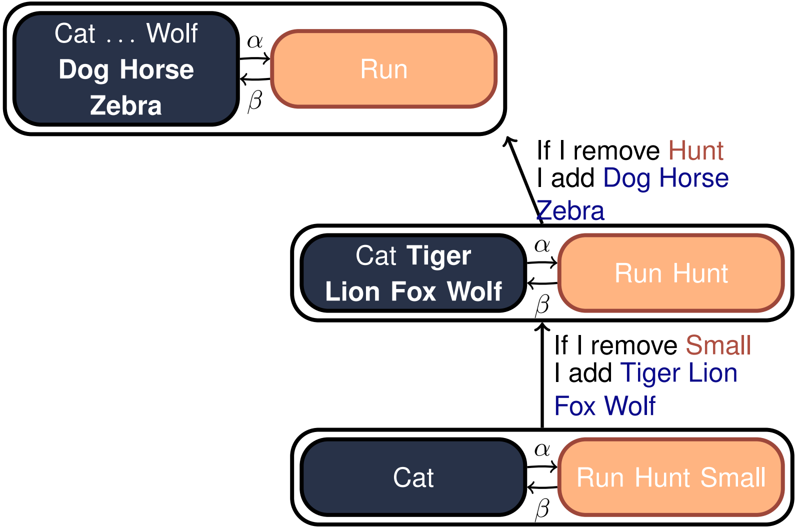
Example



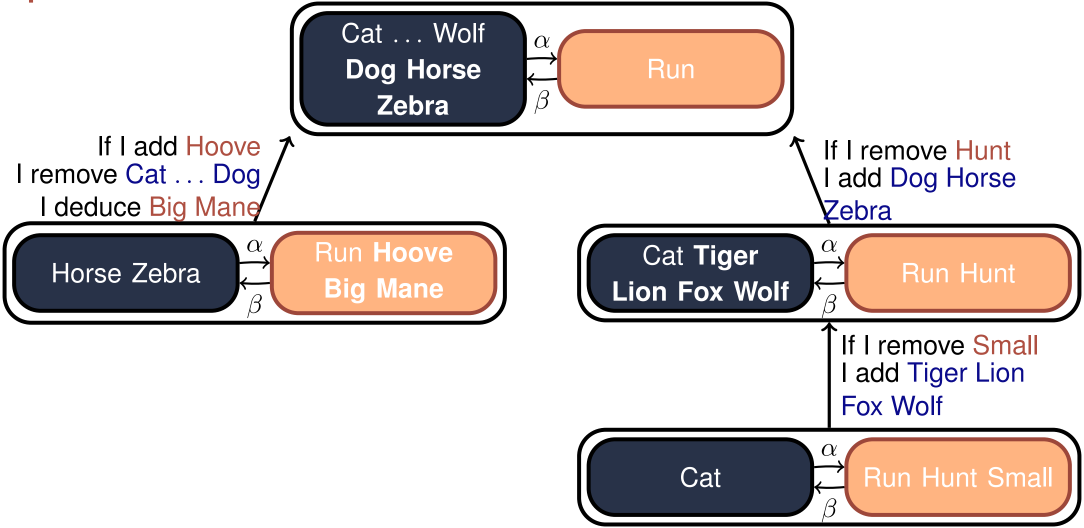
Example


Cat Tiger
Lion Fox Wolf
α
β
Run Hunt
If I remove Small
I add Tiger Lion
Fox Wolf
Cat
α
β
Run Hunt Small

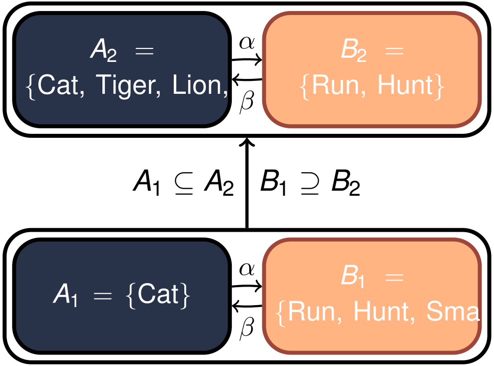
Example


Cat . . . Wolf
Dog Horse
Zebra
α
β
Run
If I remove Hunt
I add Dog Horse
Zebra
Cat Tiger
Lion Fox Wolf
α
β
Run Hunt
If I remove Small
I add Tiger Lion
Fox Wolf
Cat
α
β
Run Hunt Small

Example


Cat . . . Wolf
Dog Horse
Zebra
α
β
Run
If I add Hoove
I remove Cat . . . Dog
I deduce Big Mane
Horse Zebra
α
β
Run Hoove
Big Mane
If I remove Hunt
I add Dog Horse
Zebra
Cat Tiger
Lion Fox Wolf
α
β
Run Hunt
If I remove Small
I add Tiger Lion
Fox Wolf
Cat
α
β
Run Hunt Small

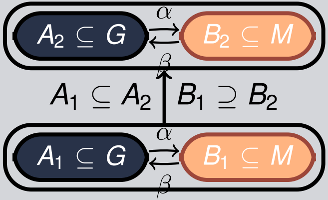
Definitions



Concept Lattice

A concept lattice consists of all concepts of a context and the subconcept–superconcept relation $\leq$ defined for (A_1, B_1) and (A_2, B_2) by:

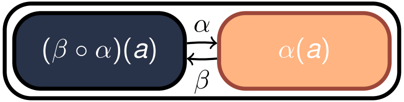
$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 \iff B_2 \subseteq B_1$$



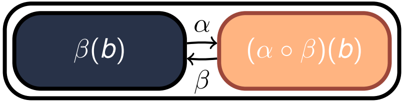
Reduced Concept Lattice

A concept lattice is usually represented in a reduced form involving:

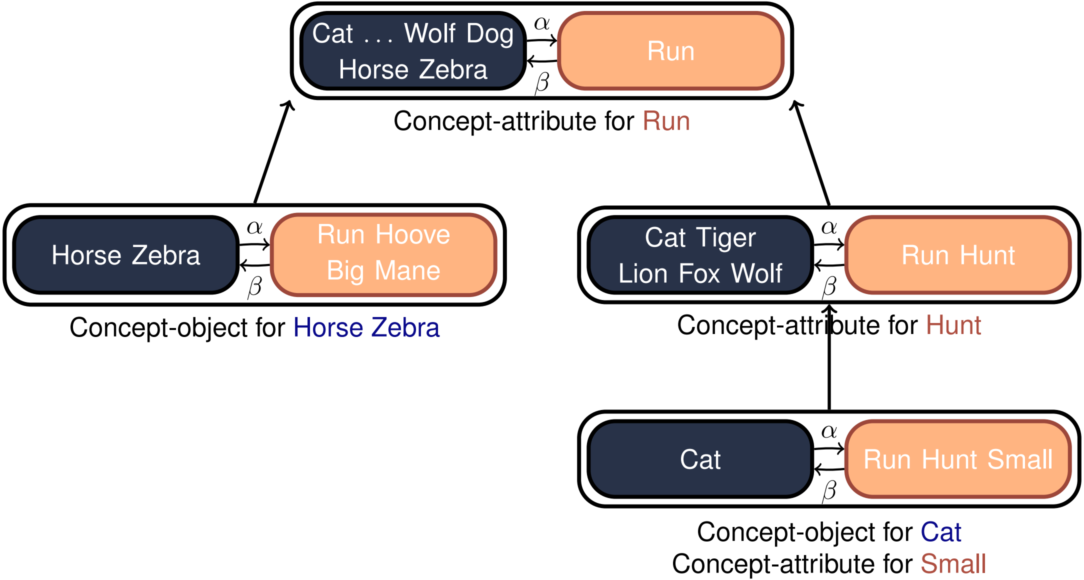
- A reduction to its Hasse diagram:
  - only the cover relations of the subconcept-superconcept order $\leq$ are displayed
  - reflexive and transitive relations of $\leq$ are omitted.
- A reduction to the new object $a \in G$ introduced by a concept-object
- A reduction to the new attribute $b \in M$ introduced by a concept-attribute
- An empty concept for the others concepts:
  - Its attributes and objects are derived from concept-objects and concept-attributes.

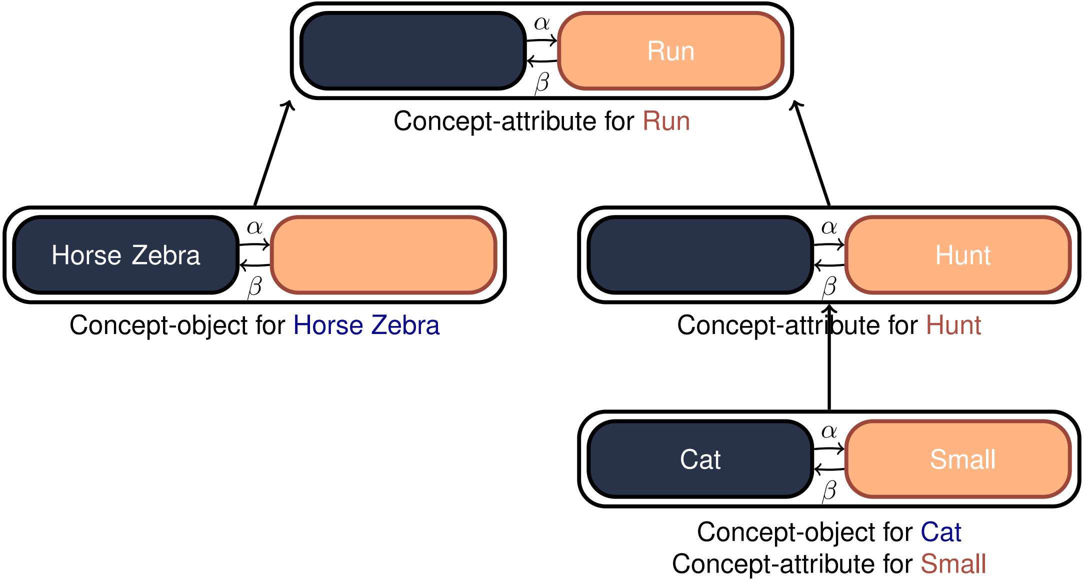


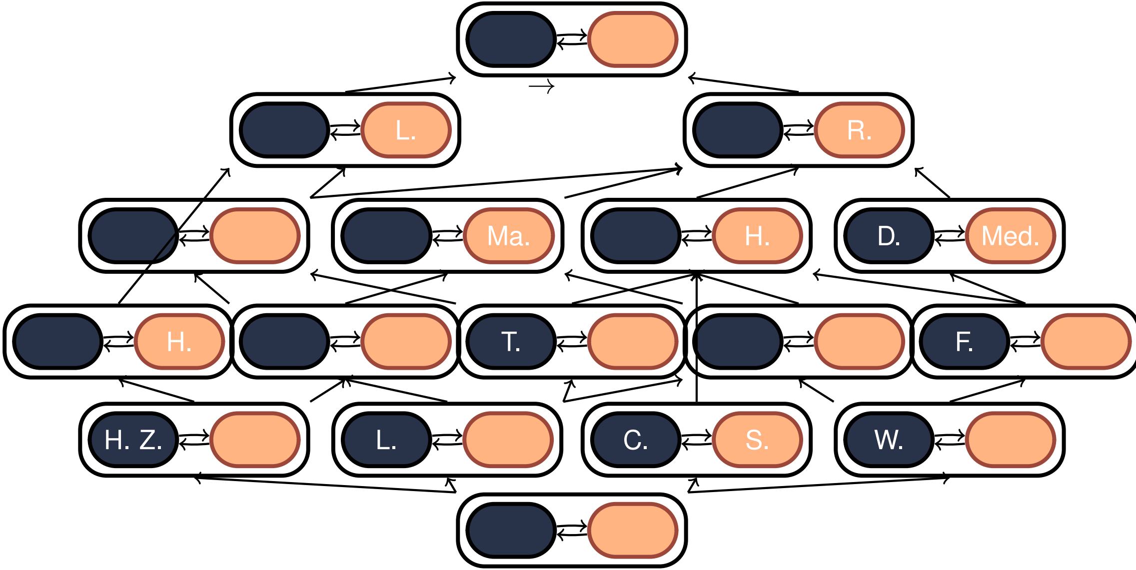
Concept-object for a

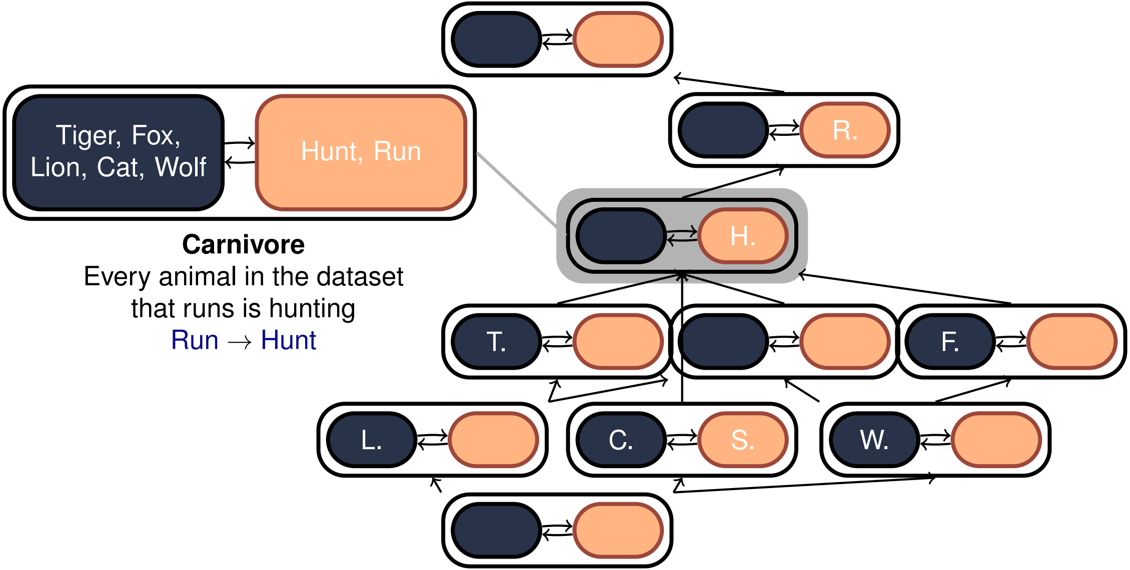


Concept-attribute for b

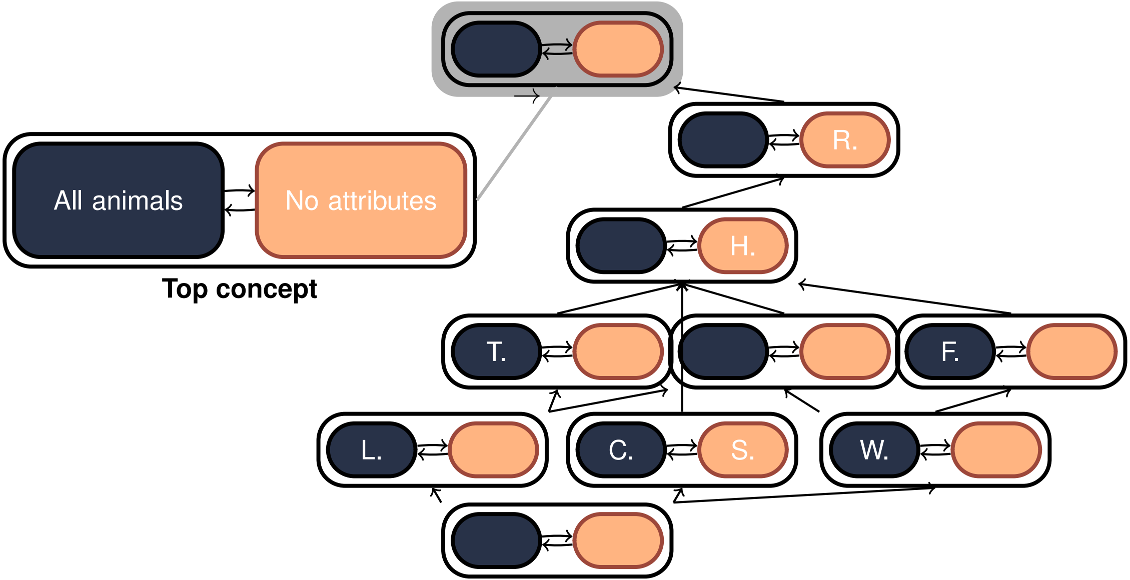

Cat . . . Wolf Dog
Horse Zebra
α
β
Run
Concept-attribute for Run
Horse Zebra
α
β
Run Hoove
Big Mane
Concept-object for Horse Zebra
Cat Tiger
Lion Fox Wolf
α
β
Run Hunt
Concept-attribute for Hunt
Cat
α
β
Run Hunt Small
Concept-object for Cat
Concept-attribute for Small

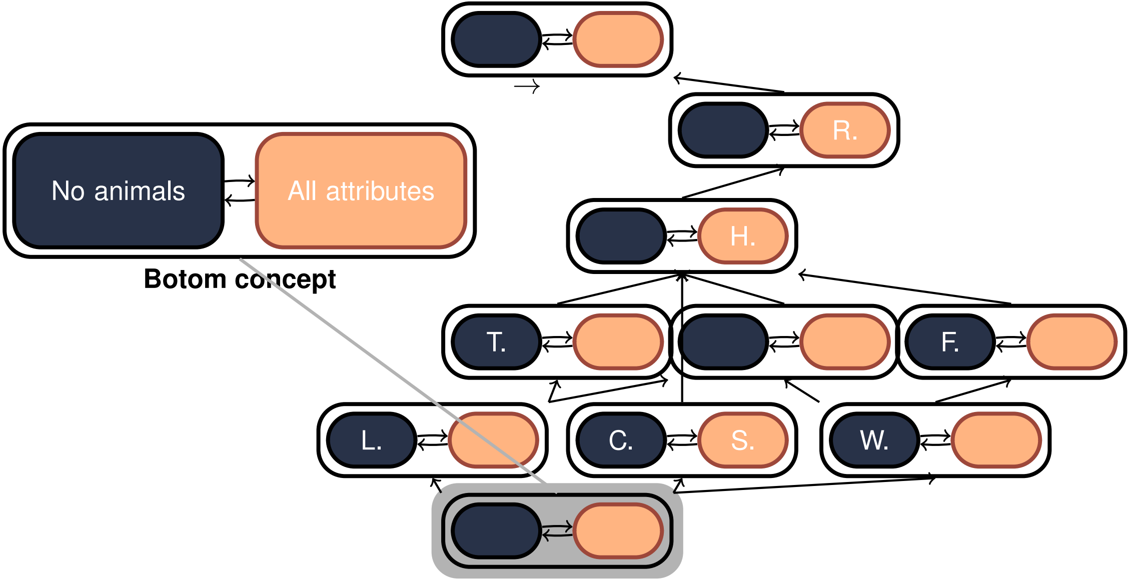

α
β
Run
Concept-attribute for Run
Horse Zebra
α
β
Concept-object for Horse Zebra
α
β
Hunt
Concept-attribute for Hunt
Cat
α
β
Small
Concept-object for Cat
Concept-attribute for Small


L.
R.
Ma.
H.
D.
Med.
H.
T.
F.
H. Z.
L.
C.
S.
W.



Carnivore
Every animal in the dataset
that runs is hunting
Run $\rightarrow$ Hunt


R.
All animals
No attributes
H.
T.
F.
L.
C.
S.
W.
Top concept


R.
No animals
All attributes
H.
Botom concept
T.
F.
L.
C.
S.
W.

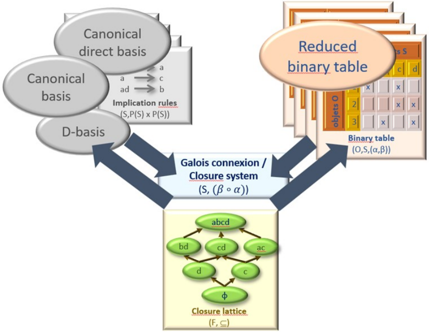
Algebraic foundations of FCA

Triple of bijective mappings

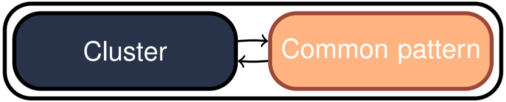
Three bijective mappings are induced by the Galois connection (α, β) through the closure operator $(\beta \circ \alpha)$ [5]:

- between (reduced) data and concepts (fundamental FCA theorem [6])
- between (reduced) data and rule bases (through concepts)

$\Rightarrow$ Deterministic generation algorithms



Data	**Algos FCA**	
Binary	NextClosure (1982)[7], . . .	✓ Bijections (Galois connection) ✗ Pattern explosion
Complex	Pattern Structures (2001)[8], LCA (2001)[9], . . .	✓ Extension of the Galois connection ✗ Botom-up generation



Hierarchy

Data	**Algos PM**	
Categ.	Apriori (1994)[1], A-Close (2003), . . .	✓ Frequent (closed) patterns ✗ Pattern explosion
Complex	**Sequences:** GSP (1996)[13], PrefixSpan (2001), CloSpan (2003), . . . **Graphs:** gSpan (2002)[14], . . .	✓ Top-down generation ✗ Dedicated algorithms

Data	Algos FCA	
Binary	NextClosure (1982)[7], ...	✓ Bijections (Galois connection) ✗ Pattern explosion
Complex	Pattern Structures (2001)[8], LCA (2001)[9], ...	✓ Extension of the Galois connection ✗ Botom-up generation

$\Longrightarrow$

Data	Algos GALACTIC	
Complex & Heterog.	NPC (2020)[10] RCC (2025)[11] LSE (2026)[12]	✓ Extension of the Galois connection ✓ Top-down and random generation ✓ Exploration strategies ✗ Recent early-stage lab work

Data	Algos PM	
Categ.	Apriori (1994)[1], A-Close (2003), ...	✓ Frequent (closed) patterns ✗ Pattern explosion
Complex	**Sequences:** GSP (1996)[13], PrefixS-pan (2001), CloSpan (2003), ... **Graphs:** gSpan (2002)[14], ...	✓ Top-down generation ✗ Dedicated algorithms

Input Data

GALACTIC handles heterogeneous data as input.

- **Simple data** (Boolean, Numerical, Categorical)
- **Complex data** (String, Sequences, Temporal Sequences, ...)

Input Data

GALACTIC considers a dataset (G, S) where G is a set of objects, S is a set of characteristics, and each characteristic $s \in S$ is a mapping from objects to a domain D_s:

$$s : G \to D_s$$

Numerical	$s : G \to \mathbb{R}$
Boolean	$s : G \to \{0, 1\}$
Categorical	$s : G \to \Sigma_s$
Sequence	$s : G \to \Sigma_s^*$

Predicates

GALACTIC introduces generic predicates for each characteristic.

- A predicate assigns one of the three values - *true*, *false*, or *none* - to each object.
- The *none* value value makes it possible to represent uncertainty

Generic Predicate

For each characteristic $s \in S$, a predicate p_s is defined as:

$$p_s : G \to \{0, 1, \text{none}\}$$

where $p_s(a) = 1$ whenever the value $s(a)$ satisfies the predicate p_s
$p_s(a) = 0$ whenever the value $s(a)$ don't satisfies the predicate p_s
$p_s(a) = ?$ whenever the value $s(a)$ is unknown

Some Predicates

Predicate p_s	Characteristic $s \in S$	Definition (for $a \in G$)
$category_s$	Categorical	$\mathrm{category}_s(a)[X] \equiv \langle s(a) \in X\ ?\rangle$ *Is $s(a)$ contained in the category set X?*
leq_s	Number	$\mathrm{leq}_s(a)[x] \equiv \langle s(a) \leq x\ ?\rangle$ *Is $s(a)$ less than or equal to x?*
geq_s	Number	$\mathrm{geq}_s(a)[x] \equiv \langle s(a) \geq x\ ?\rangle$ *Is $s(a)$ greater than or equal to x?*
$Subsequence_s$	Sequence	$\mathrm{SubSequence}_s(a)[x] \equiv \langle x \sqsubseteq s(a)\ ?\rangle$ *Does $s(a)$ contain x as a subsequence?*

Descriptions

GALACTIC introduces generic descriptions for each charactersitic

- A description is a mapping that associate a set of true predicates to a subset of objects
- The description are computed on the fly for each subset of objects

Generic Description

For a subset $A \subseteq G$, and for a characteristic $s \in S$, a description δ_s is defined by:

$$\begin{array}{rccl} \delta_s : & 2^G & \mapsto & 2^P \\ & A & \rightarrow & \{p_s \in P \mid \forall a \in A, p_s(a) = true\} \end{array}$$

Concepts

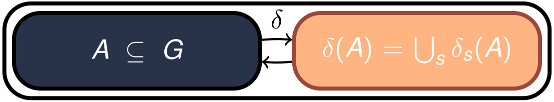
GALACTIC defines concepts as a pair of objects-predicates:

- The predicate intent contains the predicates of the descriptions of all characteristics, allowing heterogeneous data types to be combined
- The object extent contains all objects that satisfy every predicate in the intent

Concept $(A, \delta(A))$

A concept is a pair $(A, \delta(A))$ such that:

$$\delta(A) = \bigcup_{s \in S} \delta_s(A) \quad \text{and} \quad A = \{a \in G \mid \forall p_s \in \delta(A), p_s(a)\}$$



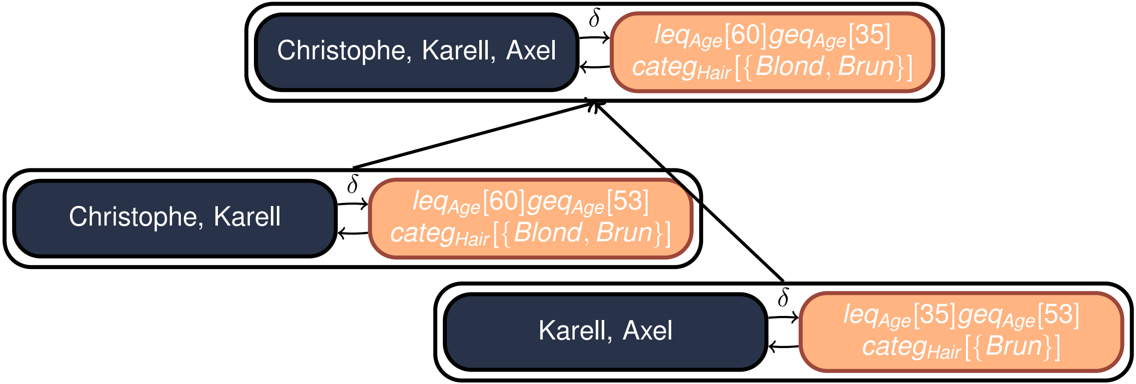
Concept $(A\,\delta(A))$

Example

	Age	Hair
Christophe	60	Blond
Karell	53	Brun
Axel	35	Brun

$$\delta_{Age}(A) = \left\{ \begin{matrix} leq_{Age}[max(Age(A))] \\ geq_{Age}[min(Age(A))] \end{matrix} \right\}$$

$$\delta_{Hair}(A) = \{\ category_{Hair}[\{Hair(A)\}]\ \}$$



Strategies

GALACTIC provides generic strategies of exploration of the data that enable the generation of strict subsets of $A \subset G$ that serve as candidates for subconcepts

Generic Strategy

For a subset $A \subseteq G$, and for a characteristic $s \in S$, a strategy σ_s is defined by:

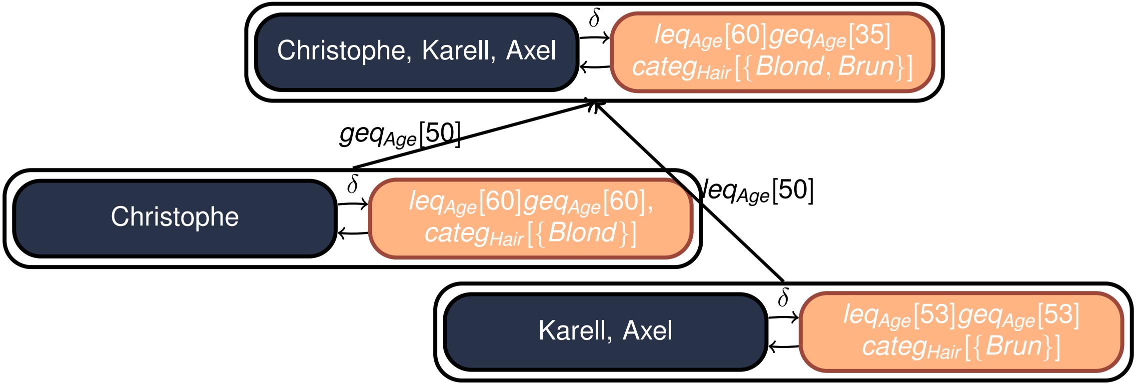
$$\begin{array}{rccc} \sigma_s: & 2^G & \mapsto & 2^G \\ & A & \rightarrow & A' \subset A \end{array}$$

A basic strategy depends on a characteristic $s \in S$, and generates predicates - called selectors - that are true only for a strict subset of $A \subset G$

Example

	Age	Hair
Christophe	60	Blond
Karell	53	Brun
Axel	35	Brun

$$\sigma_{Age}(A) = \left\{ \begin{array}{ll} leq_{Age} & [mean(Age(A)) + \alpha * dev] \\ geq_{Age} & [mean(Age(A)) + \alpha * dev] \end{array} \right\}$$

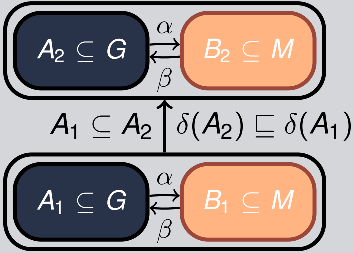


Types of strategies

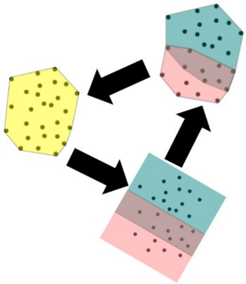
- A basic strategy depends on characteristics and generates predicates - called selectors - that are true only for a strict subset of $A \subset G$
- A meta strategy depends on other strategies
  - `Naive-Strategy()`: generates all possible subconcepts by considering every possible subset of objects
  - `Recursive-Strategy(strategy-name)`: recursively applies the strategy provided as a parameter to further explore the concept hierarchy
- A valued strategy depends on a measure
  - `Selective-Strategy(strategy-name, measure, ratio, threshold)`: applies a strategy and retains only the top `ratio` fraction of the generated subconcepts that satisfy the `threshold`, according to a filtering measure:
    - **Support**: the number of objects contained in a concept.
    - **Entropy**: the class-based entropy of a concept with respect to a class attribute.

Pattern Structure [8]

If the predicate space 2^P is equipped with a subsumption relation such that $\delta(A') \sqsubseteq \delta(A)$ whenever $A \subseteq A'$, then the Galois connection between 2^G and 2^P is preserved.

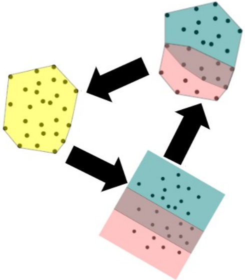


Rather than being precomputed, the description $\delta(A)$ is generated on the fly by GALACTIC during the analysis.



Description $\delta(A)$

The description $\delta(A)$ is defined as predicates that characterize the boundary of the generalized convex hull of objects A

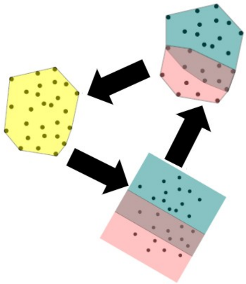


Description $\delta(A)$

The description $\delta(A)$ is defined as predicates that characterize the boundary of the generalized convex hull of objects A

Strategy $\sigma(A)$

A strategy $\sigma(A)$ defines a cut of the generalized convex hull, used to refine A into candidate subgroups $A' \subseteq A$



Description $\delta(A)$

The description $\delta(A)$ is defined as predicates that characterize the boundary of the generalized convex hull of objects A

Strategy $\sigma(A)$

A strategy $\sigma(A)$ defines a cut of the generalized convex hull, used to refine A into candidate subgroups $A' \subseteq A$

Subconcept $(A', \delta(A'))$

For each selected candidate subgroup $A' \subseteq A$, a new description $\delta(A'))$ is computed. The resulting pair $(A', \delta(A')$ forms a subconcept of the concept $(A, \delta(A))$

GALACTIC maintains a concept lattice structure:

- The Galois connection between the data space and the predicate-based description spaces is preserved
- The three bijective mappings between data, concepts, and rules are maintained
- Concept lattice and basis of rules generation algorithms are deterministic and require no parameter tuning.

GALACTIC generates predicates without preprocessing:

- Predicates are progressively discovered during analysis, initially with the description predicates of the dataset (G, S)
- They are incrementally enriched by predicates of newly added concepts
- The set P of discovered predicates corresponds to the data, and the resulting lattice is the concept lattice of (G, P, δ) [10].

Description spaces [15]

The description spaces are characterized through a double Galois connection between the data space $(2^G, \subseteq)$ and the space of descriptive predicates 2^P depending on whether they are considered:

- as classical binary attributes $(2^P, \subseteq)$
- or as predicates endowed with semantic relationships $(2^P, \sqsubseteq)$

$$(2^G, \subseteq) \underset{\alpha \supseteq \delta}{\overset{\beta}{\leftrightarrows}} (2^P, \subseteq) \underset{\epsilon = max_{\sqsubseteq}}{\overset{\alpha \circ \beta}{\leftrightarrows}} (2^P, \sqsubseteq)$$

The predicate description $\delta(A)$ is characterized by $\mathrm{max}_{\sqsubseteq}(\alpha(A))$, the semantically maximal predicates of the potentially infinite set $\alpha(A)$
They can be computed dynamically as the extreme points $\epsilon(\alpha(A))$ of the generalized convex hull

GALACTIC implements two algorithms, starting from the full concept $(G, \delta(G))$:

NextPriorityConcept (NPC)

The **NPC algorithm** [10]:

- generates the lattice in a top-down approach, with a same recursive strategy
- uses a constraint propagation mechanism to ensure that a lattice is produced

SublatticeExtension (SLE)

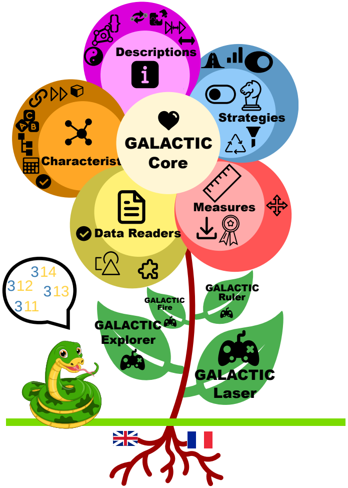
The **SLE algorithm** [12]:

- generates the lattice in a random approach, with the same principle as the RCC algorithm [11] for classical FCA binary data
- enables the insertion of user-selected concepts, with the possibility of switching strategies

The platform

Written in python and Fully extensible

- A core library:
  - Implementation of NPC, RCC and SLE algorithms
  - A nested collection of python packages for the algebraic foundations
- An ecosystem of extensions:
  - Data readers
  - Characteristics
  - Descriptions
  - Strategies
  - Measures
- Several applications



Tutorials

Tutorial Prog1

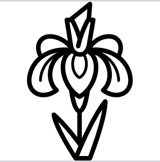
Learn how to build and manipulate a hierarchy of concepts:

- For core simple data type: `Boolean`, `Category` and `Number`
- For more complex data handled in the extensions: `String`, `Sequences`, `Temporal Sequences`, `Interval Sequences` (forthcoming)

with:

- Specialized descriptions and strategies
- Core strategies and measures

Notebook

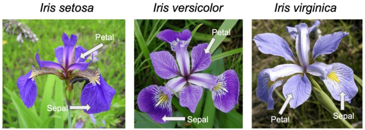


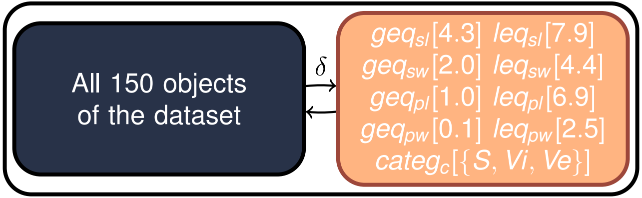
Iris sample

Iris DataSet

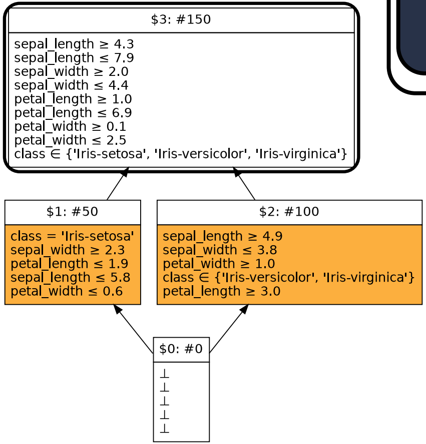
Attributes	Characteristic	Description	Basic Strategy	Valued Strategy
sepal_length (sl)	Number	δ_{sl}^{N}	σ_{sl}^{N}	$\sigma_{sl}^{E}(c)\ \sigma_{sl}^{S}$
sepal_width (sw)	Number	δ_{sw}^{N}	σ_{sw}^{N}	$\sigma_{sw}^{E}(c)\ \sigma_{sw}^{S}$
petal_length (pl)	Number	δ_{pl}^{N}	σ_{pl}^{N}	$\sigma_{pl}^{E}(c)\ \sigma_{pl}^{S}$
petal_width (pw)	Number	δ_{pw}^{N}	σ_{pw}^{N}	$\sigma_{pw}^{E}(c)\ \sigma_{pw}^{S}$
class (c)	Category	δ_{c}^{C}		

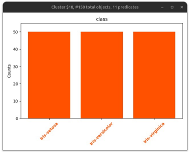
- `Numerical-Description` δ^N (Number)
- `Category-Description` δ^C (Category)
- `Numerical-Quantile-Strategy` σ^N (Number)
- `Entropy-Strategy` σ^E (Entropy measure)
- `Support-Strategy` σ^S (Support measure)

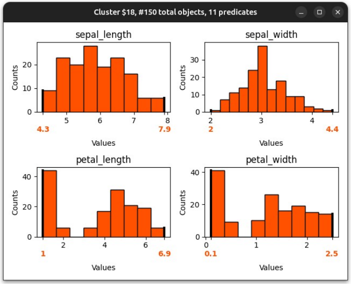


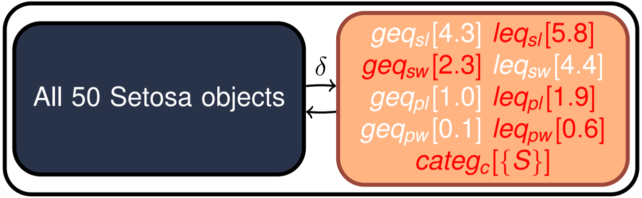

All 150 objects of the dataset
δ
$geq_{sl}[4.3]$ $leq_{sl}[7.9]$
$geq_{sw}[2.0]$ $leq_{sw}[4.4]$
$geq_{pl}[1.0]$ $leq_{pl}[6.9]$
$geq_{pw}[0.1]$ $leq_{pw}[2.5]$
$categ_c[\{S, Vi, Ve\}]$

Concept $3

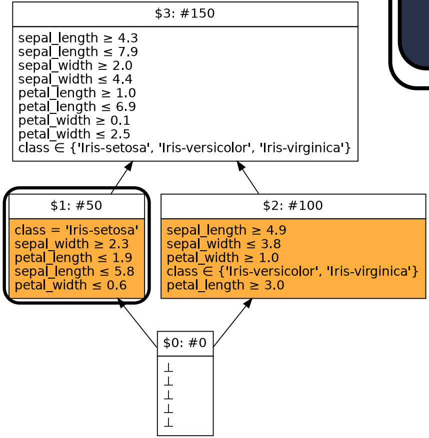

$3: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$1: #50
class = 'Iris-setosa'
sepal_width ≥ 2.3
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$2: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$0: #0
⊥
⊥
⊥
⊥
⊥

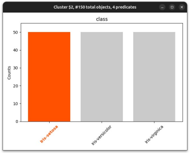

class
Counts
Iris-setosa
Iris-versicolor
Iris-virginica

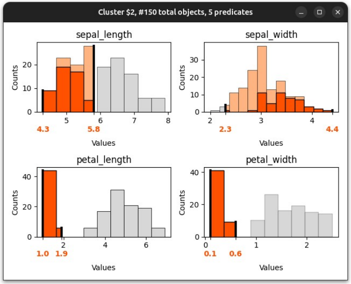

Cluster $18, #150 total objects, 11 predicates
sepal_length
sepal_width
petal_length
petal_width
Counts
Values
4.3
7.9
2
4.4
1
6.9
0.1
2.5

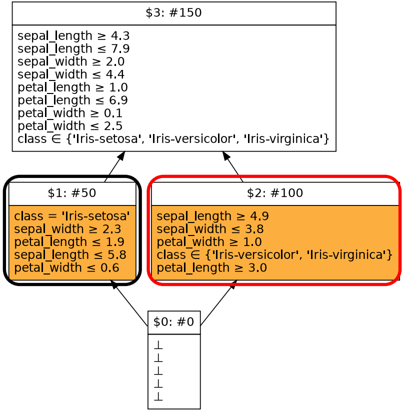

All 50 Setosa objects
δ
$geq_{sl}[4.3]$ $leq_{sl}[5.8]$
$geq_{sw}[2.3]$ $leq_{sw}[4.4]$
$geq_{pl}[1.0]$ $leq_{pl}[1.9]$
$geq_{pw}[0.1]$ $leq_{pw}[0.6]$
$categ_c[\{S\}]$

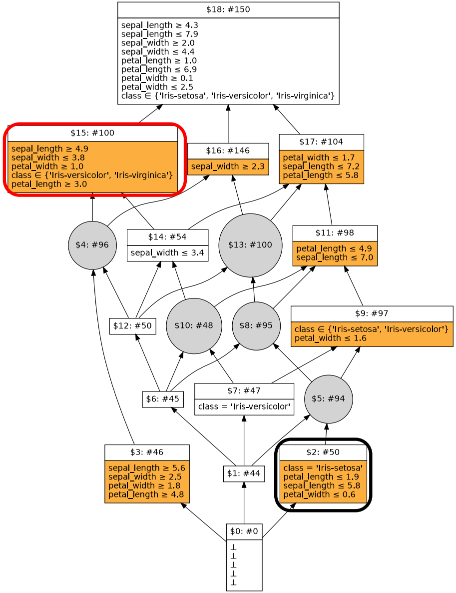
Concept $1

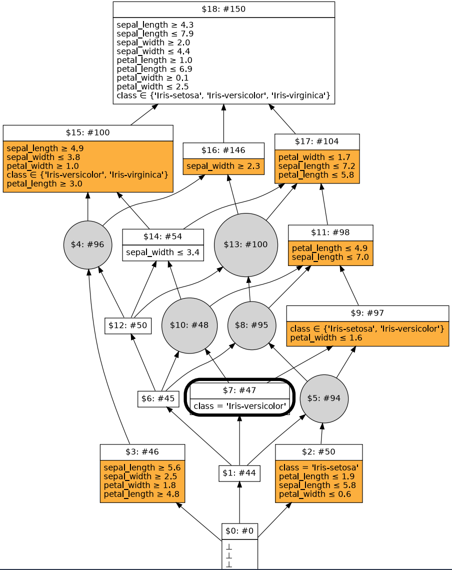

$3: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$1: #50
class = 'Iris-setosa'
sepal_width ≥ 2.3
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$2: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$0: #0
⊥
⊥
⊥
⊥
⊥

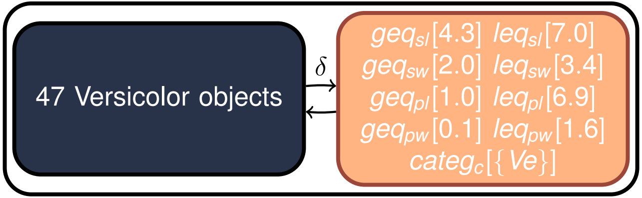

class
Iris-setosa
Iris-versicolor
Iris-virginica


Cluster $2, #150 total objects, 5 predicates
sepal_length
sepal_width
petal_length
petal_width
Counts
Values
4.3
5.8
2.3
4.4
1.0
1.9
0.1
0.6

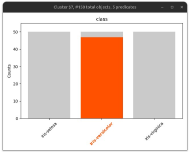

$3: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$1: #50
class = 'Iris-setosa'
sepal_width ≥ 2.3
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$2: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$0: #0
⊥
⊥
⊥
⊥
⊥

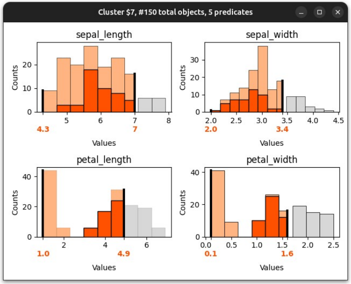

$18: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$15: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$16: #146
sepal_width ≥ 2.3
$17: #104
petal_width ≤ 1.7
sepal_length ≤ 7.2
petal_length ≤ 5.8
$4: #96
$14: #54
sepal_width ≤ 3.4
$13: #100
$11: #98
petal_length ≤ 4.9
sepal_length ≤ 7.0
$12: #50
$10: #48
$8: #95
$9: #97
class ∈ {'Iris-setosa', 'Iris-versicolor'}
petal_width ≤ 1.6
$6: #45
$7: #47
class = 'Iris-versicolor'
$5: #94
$3: #46
sepal_length ≥ 5.6
sepal_width ≥ 2.5
petal_width ≥ 1.8
petal_length ≥ 4.8
$1: #44
$2: #50
class = 'Iris-setosa'
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$0: #0
⊥
⊥
⊥
⊥
⊥

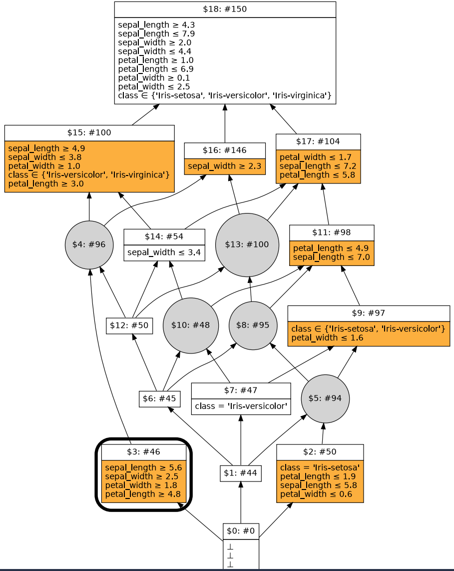

$18: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$15: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$16: #146
sepal_width ≥ 2.3
$17: #104
petal_width ≤ 1.7
sepal_length ≤ 7.2
petal_length ≤ 5.8
$4: #96
$14: #54
sepal_width ≤ 3.4
$13: #100
$11: #98
petal_length ≤ 4.9
sepal_length ≤ 7.0
$12: #50
$10: #48
$8: #95
$9: #97
class ∈ {'Iris-setosa', 'Iris-versicolor'}
petal_width ≤ 1.6
$6: #45
$7: #47
class = 'Iris-versicolor'
$5: #94
$3: #46
sepal_length ≥ 5.6
sepal_width ≥ 2.5
petal_width ≥ 1.8
petal_length ≥ 4.8
$1: #44
$2: #50
class = 'Iris-setosa'
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$0: #0
⊥
⊥
⊥

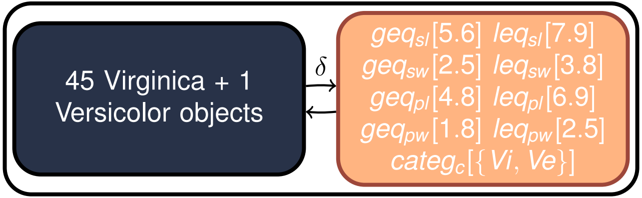

47 Versicolor objects
δ
$geq_{sl}[4.3]$ $leq_{sl}[7.0]$
$geq_{sw}[2.0]$ $leq_{sw}[3.4]$
$geq_{pl}[1.0]$ $leq_{pl}[6.9]$
$geq_{pw}[0.1]$ $leq_{pw}[1.6]$
$categ_c[\{Ve\}]$

Concept $7

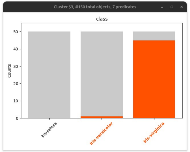

class
Counts
Iris-setosa
Iris-versicolor
Iris-virginica

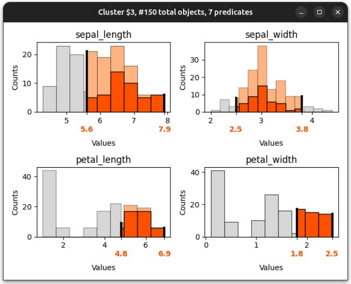

Cluster $7, #150 total objects, 5 predicates
sepal_length
sepal_width
petal_length
petal_width
Counts
Values
4.3
7
2.0
3.4
1.0
4.9
0.1
1.6

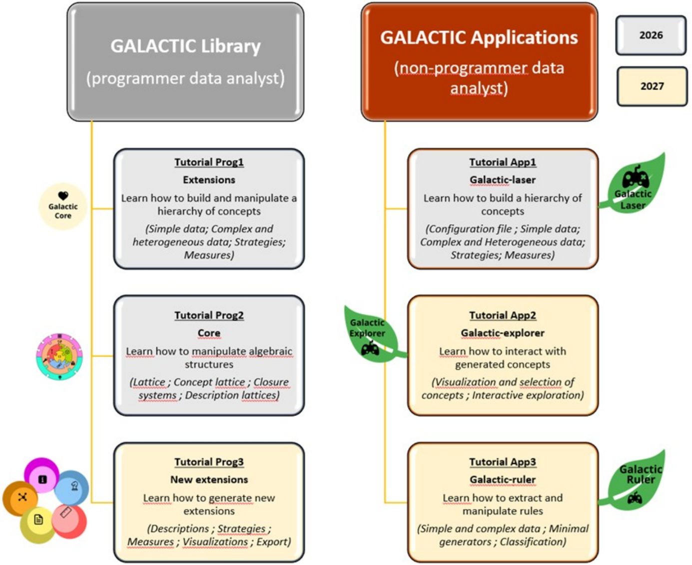

$18: #150
sepal_length ≥ 4.3
sepal_length ≤ 7.9
sepal_width ≥ 2.0
sepal_width ≤ 4.4
petal_length ≥ 1.0
petal_length ≤ 6.9
petal_width ≥ 0.1
petal_width ≤ 2.5
class ∈ {'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
$15: #100
sepal_length ≥ 4.9
sepal_width ≤ 3.8
petal_width ≥ 1.0
class ∈ {'Iris-versicolor', 'Iris-virginica'}
petal_length ≥ 3.0
$16: #146
sepal_width ≥ 2.3
$17: #104
petal_width ≤ 1.7
sepal_length ≤ 7.2
petal_length ≤ 5.8
$4: #96
$14: #54
sepal_width ≤ 3.4
$13: #100
$11: #98
petal_length ≤ 4.9
sepal_length ≤ 7.0
$12: #50
$10: #48
$8: #95
$9: #97
class ∈ {'Iris-setosa', 'Iris-versicolor'}
petal_width ≤ 1.6
$6: #45
$7: #47
class = 'Iris-versicolor'
$5: #94
$3: #46
sepal_length ≥ 5.6
sepal_width ≥ 2.5
petal_width ≥ 1.8
petal_length ≥ 4.8
$1: #44
$2: #50
class = 'Iris-setosa'
petal_length ≤ 1.9
sepal_length ≤ 5.8
petal_width ≤ 0.6
$0: #0
⊥
⊥
⊥


45 Virginica + 1 Versicolor objects
δ
$geq_{sl}[5.6]$ $leq_{sl}[7.9]$
$geq_{sw}[2.5]$ $leq_{sw}[3.8]$
$geq_{pl}[4.8]$ $leq_{pl}[6.9]$
$geq_{pw}[1.8]$ $leq_{pw}[2.5]$
$categ_c[\{Vi, Ve\}]$

Concept $3


class


Cluster $3, #150 total objects, 7 predicates
sepal_length
sepal_width
petal_length
petal_width
Counts
Values
5.6
7.9
2.5
3.8
4.8
6.9
1.8
2.5


GALACTIC Library
(programmer data analyst)
GALACTIC Applications
(non-programmer data analyst)
2026
2027
Galactic Core
Tutorial Prog1
Extensions
Learn how to build and manipulate a hierarchy of concepts
(Simple data; Complex and heterogeneous data; Strategies; Measures)
Tutorial Prog2
Core
Learn how to manipulate algebraic structures
(Lattice ; Concept lattice ; Closure systems ; Description lattices)
Tutorial Prog3
New extensions
Learn how to generate new extensions
(Descriptions ; Strategies ; Measures ; Visualizations ; Export)
Tutorial App1
Galactic-laser
Learn how to build a hierarchy of concepts
(Configuration file ; Simple data; Complex and Heterogeneous data; Strategies; Measures)
Galactic Laser
Tutorial App2
Galactic-explorer
Learn how to interact with generated concepts
(Visualization and selection of concepts ; Interactive exploration)
Galactic Explorer
Tutorial App3
Galactic-ruler
Learn how to extract and manipulate rules
(Simple and complex data ; Minimal generators ; Classification)
Galactic Ruler

Tutorials for programmer data-analysts

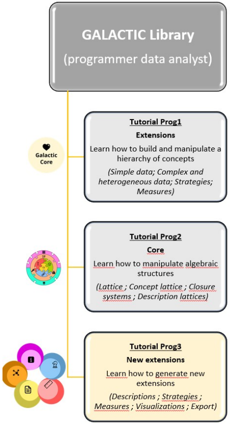
Tutorial Prog2 (forthcoming)

Learn how to work with algebraic structures:

- Concept lattices, lattice theory, and closure operators
- Triplet of bijections between data, concept lattices, and rule bases
- Description spaces and main algorithms
- Extensions of rule bases

Tutorial Prog3 (forthcoming)

Learn how to generate new characteristic, descriptions, and strategies



Tutorials for non-programmer data-analysts

Tutorial App1 (forthcoming)

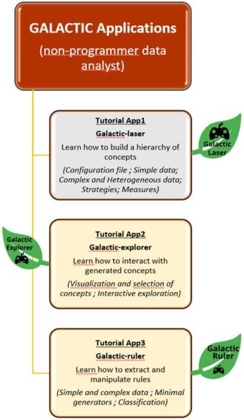
Learn how to build a hierarchy of concepts using the `Galactic-laser` application (YAML configuration files)

Tutorial App2 (forthcoming)

Learn how to interact with concepts using the `Galactic-explorer` application

Tutorial App3 (forthcoming)

Learn how to extract rules using the `Galactic-ruler` application

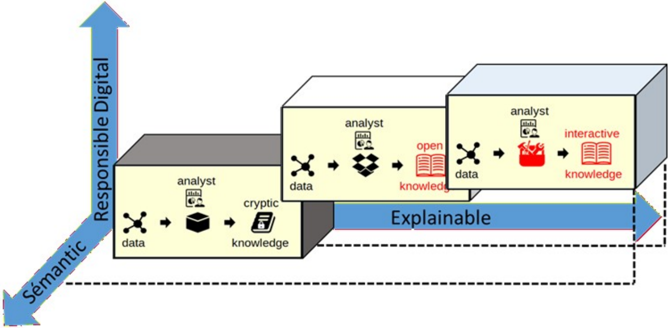


Conclusion

- An initial version 0.4 in 2019 as a proof of concept
  - Descriptions and strategies for (temporal) sequences [16] (PhDs work)
  - `Galactic-laser` and `Galactic-explorer` applications
  - Advances in the algebraic understanding
- A core redesign version 0.5 in 2026, aiming at a higher TRL
  - Release of version 0.5 at PFIA 2026
  - Website: `https://www.thegalactic.org`
  - Ongoing work and remaining improvements
- A future version 1.0 in 2027
  - With completed extensions, visualizations[2] and accompanying tutorials

[2] Thanks to Louna Denis for the concepts visualizations of the Iris dataset

GALACTIC proposes predicate embeddings as an alternative to classical vector embeddings for semantic enrichment and interactivity



Some questions ?

ml.univ-lr.fr/sympa/info/galactic

www.thegalactic.org

References I

R. R. Agrawal, "Fast algorithms for mining association rules in large databases," in *In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, p. 487–499, 1994.

P. Fournier-Viger, J.-W. Lin, R. Kiran, Y. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern recognition*, vol. 1, no. 1, pp. 54–77, 2017.

B. Ganter and R. Wille, *Formal Concept Analysis, Mathematical foundations*.
Springer Verlag, Berlin, 1999.

O. Ore, "Galois connexions," *Transactions of the American Mathematical Society*, vol. 55, p. 493–513, 1944.

K. Bertet, C. Demko, J.-F. Viaud, and C. Guérin, "Lattices, closure systems and implication bases: a survey of structural aspects and algorithms," *Theoretical Computer Science*, vol. 743, pp. 93–109, 2018.

B. M. M. Barbut, *Ordres et classifications: Algèbre et combinatoire*.
1970.

R. Wille, "Restructuring lattice theory : an approach based on hierarchies of concepts," *Ordered sets*, pp. 445–470, 1982.
I. Rival (ed.), Dordrecht-Boston, Reidel.

B. Ganter and S. Kuznetsov, "Pattern Structures and Their Projections," in *Proceedings of 9th ICCS* (H. Delugach and G. Stumme, eds.), Lecture Notes in Computer Science 2120, pp. 129–142, Springer, 2001.

S. Ferré and O.Ridoux, "A Logical Generalization of Formal Concept Analysis," in *Proceeedings of ICCS* (B. Ganter and G. Mineau, eds.), Lecture Notes in Computer Science 1867, pp. 371–384, Springer, 2000.

References II

C. Demko, K. Bertet, C. Faucher, J.-F. Viaud, and S. Kuznetsov, "NextPriorityConcept: A New and Generic Algorithm Computing Concepts from Complex and Heterogeneous Data," *Theoretical Computer Science*, vol. 845, pp. 1–20, 2020.

C. Demko, J. Richard, M. W. Kemgne, C. Faucher, and K. Bertet, "Reduced Subcontext Completion in Lattices and Concept Lattices," in *Proceedings of CONCEPTS*, Lecture Notes in Computer Science 15941, pp. 51–66, Springer, 2025.

C. Demko and K. Bertet, "Sublattice extensions of moore families," in *Proceedings of CONCEPTS*, Lecture Notes in Computer Science, Springer, 2026.

R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *In Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology (EDBT)*, p. 3–17, 1996.

X. Yan and J. Han, "Gspan: Graph-based substructure pattern mining," in *Proc. 2002 of Int. Conf. on Data Mining (ICDM'02)*, 2002.

C. Demko, K. Bertet, J.-F. Viaud, C. Faucher, and D. Mondou, "Description lattices of generalised convex hulls," *International Journal of Approximate Reasoning*, p. 109269, 2024.

S.-E. Boukhetta, C. Demko, K. Bertet, J. Richard, and C. Cayèré, "Temporal sequence mining using FCA and GALACTIC," in *Proceedings of ICCS* (T. Braun, M. Gehrke, T. Hanika, and N. Hernandez, eds.), vol. 12879, pp. 185–199, Springer, 2021.