GALACTIC architecture

The **GALACTIC** Organization <contact@thegalactic.org>

2018-2023 [1]

[1]

2023-01-17

GALACTIC architecture

GALACTIC architecture

The **GALACTIC** Organization <contact@thegalactic.org>

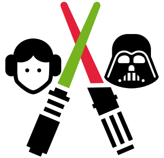
2018-2023 [1]

[1] © 2018-2023 the **GALACTIC** Organization. This document is licensed under CC-by-nc-nd (https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en)

Acronym

GALACTIC stands for

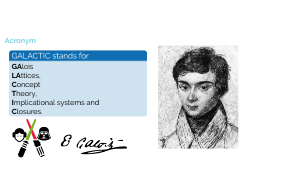
GAlois
LAttices,
Concept
Theory,
Implicational systems and
Closures.



E Galois

2023-01-17

GALACTIC architecture
└Introduction
└Acronym
└Acronym



Acronym
GALACTIC stands for
GAlois
LAttices,
Concept
Theory,
Implicational systems and
Closures.

- the **GALACTIC** logo is a tribute to the french mathematician Évariste Galois who died following a gallant duel at the age of twenty;
- it is also a graphical summary of lattice drawings.

Purpose

GALACTIC framework

Develop a framework on:

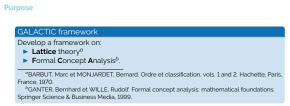
- **Lattice** theory[a]
- **F**ormal **C**oncept **A**nalysis[b].

[a] BARBUT, Marc et MONJARDET, Bernard. Ordre et classification, vols. 1 and 2. Hachette, Paris, France, 1970.

[b] GANTER, Bernhard et WILLE, Rudolf. Formal concept analysis: mathematical foundations. Springer Science & Business Media, 1999.

2023-01-17

GALACTIC architecture
└Introduction
  └Purpose
    └Purpose



Purpose

GALACTIC framework

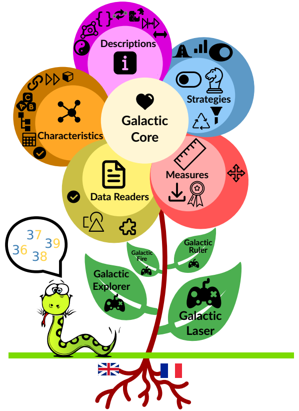
Develop a framework on:

- **Lattice** theory[a]
- **F**ormal **C**oncept **A**nalysis[b].

[a] BARBUT, Marc et MONJARDET, Bernard. Ordre et classification, vols. 1 and 2. Hachette, Paris, France, 1970.

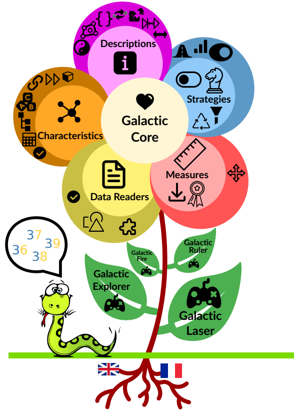
[b] GANTER, Bernhard et WILLE, Rudolf. Formal concept analysis: mathematical foundations. Springer Science & Business Media, 1999.

Architecture



- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

Architecture



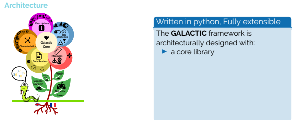
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library

2023-01-17

GALACTIC architecture
└Architecture
 └Architecture
  └Architecture


Architecture

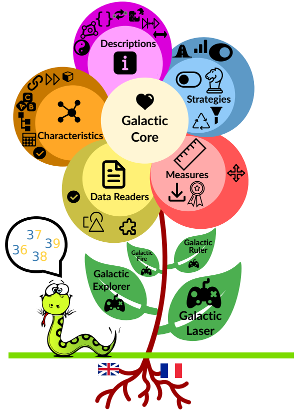
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

Architecture



Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications

Architecture

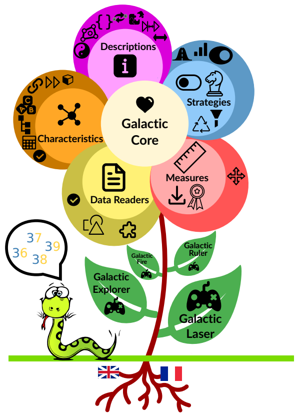
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

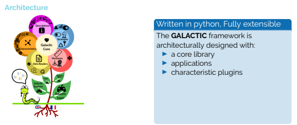
Architecture



Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins


Architecture

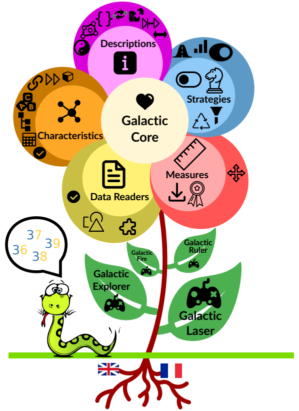
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

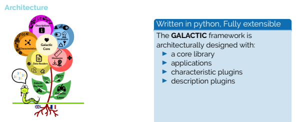
Architecture



Written in python, Fully extensible

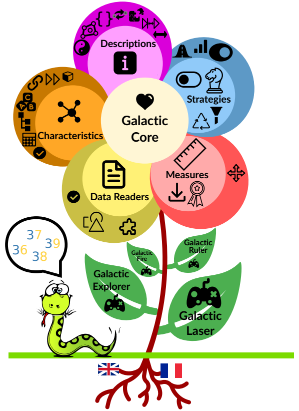
The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins
- description plugins


Architecture
Written in python, Fully extensible
The **GALACTIC** framework is architecturally designed with:
- a core library
- applications
- characteristic plugins
- description plugins

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

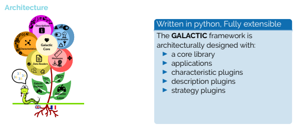
Architecture



Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins


Architecture

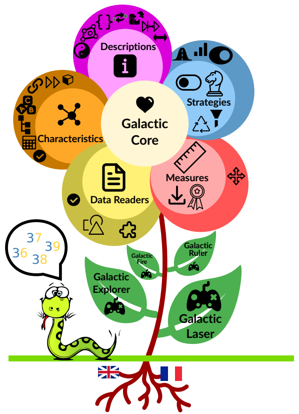
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

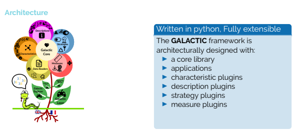
Architecture



Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins
- measure plugins


Architecture

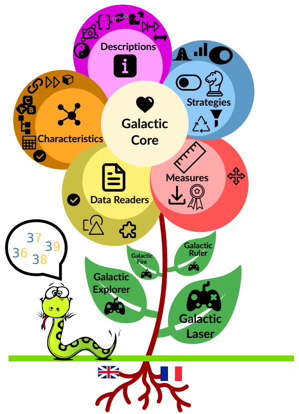
Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins
- measure plugins

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

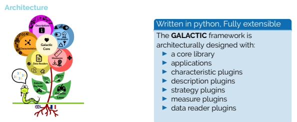
Architecture



Written in python, Fully extensible

The **GALACTIC** framework is architecturally designed with:

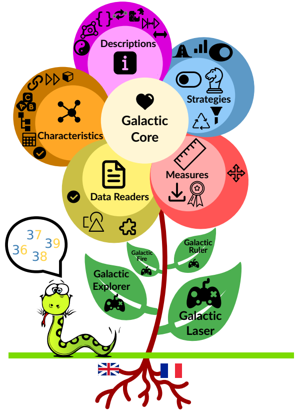
- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins
- measure plugins
- data reader plugins


Architecture

Written in python, Fully extensible
The **GALACTIC** framework is architecturally designed with:
- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins
- measure plugins
- data reader plugins

- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

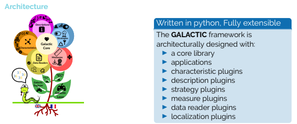
Architecture



Written in python, Fully extensible

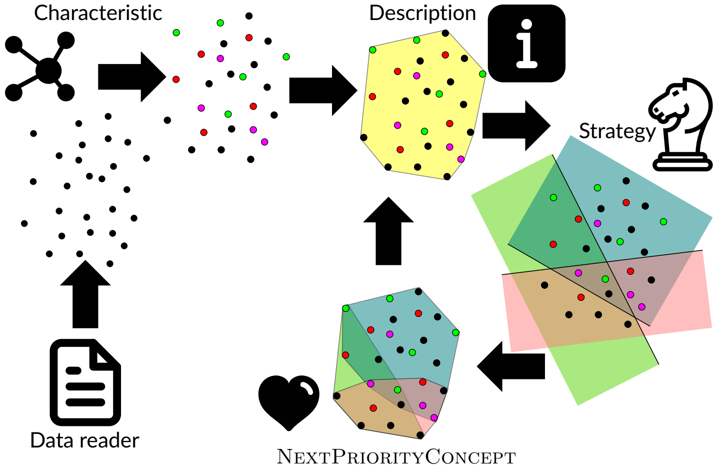
The **GALACTIC** framework is architecturally designed with:

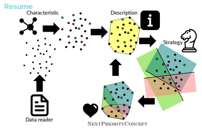
- a core library
- applications
- characteristic plugins
- description plugins
- strategy plugins
- measure plugins
- data reader plugins
- localization plugins



- the core library is the center of the flower;
- the applications are in the leaves;
- characteristic plugins are in the orange petal;
- strategy plugins are in the blue petal;
- measure plugins are in the red petal;
- data reader plugins are in the yellow petal;
- localization plugins are in the roots.

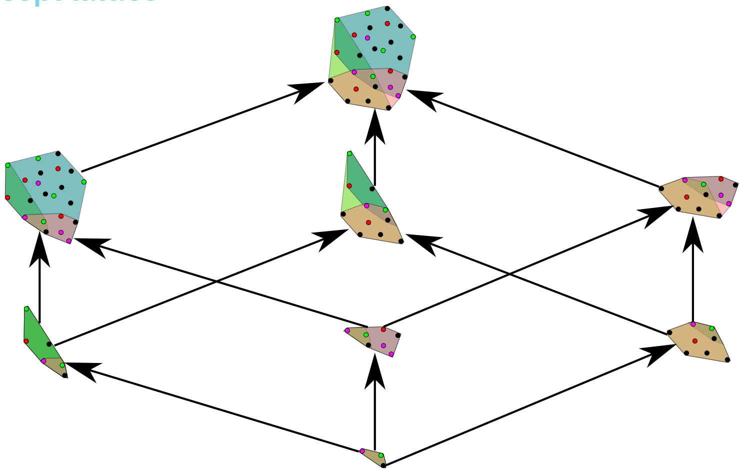
Resume





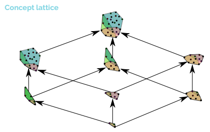
- data readers convert data files to context;
- subsets of individuals are described by sets of predicates called descriptions;
- strategies divide individuals into subsets;
- the NextPriorityConcept algorithm maintains the notion of **Lattice** during the process;
- the lattice can be converted into a minimum and consistent **Basis of rules**.

Concept lattice

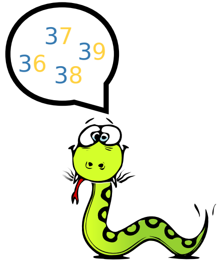


2023-01-17

GALACTIC architecture
└Architecture
└Architecture
└Concept lattice

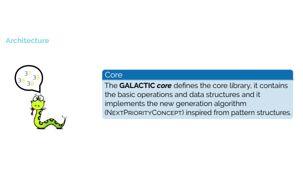


Architecture



Core

The **GALACTIC *core*** defines the core library, it contains the basic operations and data structures and it implements the new generation algorithm (NextPriorityConcept) inspired from pattern structures.


Architecture

Core
The **GALACTIC *core*** defines the core library, it contains the basic operations and data structures and it implements the new generation algorithm (NextPriorityConcept) inspired from pattern structures.

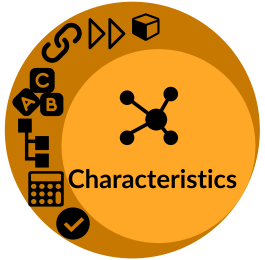
- written for python >= 3.6
- the NextPriorityConcept algorithm will be published in an international journal this autumn.

Sergei Kuznetsov was invited to the L3i in 2016, and after our discussions, we managed to propose this new algorithm.

Characteristic Plugins

Definition

Characteristic plugins define characteristics such as numerical characteristics, boolean characteristics.

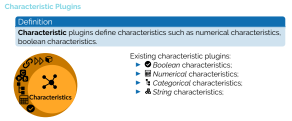


Existing characteristic plugins:

- ✓ *Boolean* characteristics;
- *Numerical* characteristics;
- *Categorical* characteristics;
- *String* characteristics;

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Characteristic Plugins


Characteristic Plugins

Definition

Characteristic plugins define characteristics such as numerical characteristics, boolean characteristics.

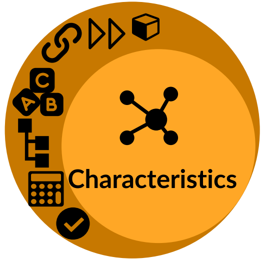
Existing characteristic plugins:

- ✓ *Boolean* characteristics;
- *Numerical* characteristics;
- *Categorical* characteristics;
- *String* characteristics;

Characteristic Plugins

Definition

Characteristic plugins define characteristics such as numerical characteristics, boolean characteristics.



Existing characteristic plugins:

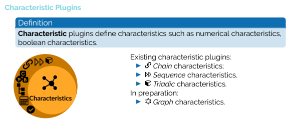
- 🔗 *Chain* characteristics;
- ▷▷ *Sequence* characteristics.
- *Triadic* characteristics.

In preparation:

- *Graph* characteristics.

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Characteristic Plugins


Characteristic Plugins
Definition
Characteristic plugins define characteristics such as numerical characteristics, boolean characteristics.
Existing characteristic plugins:
- *Chain* characteristics;
- *Sequence* characteristics.
- *Triadic* characteristics.
In preparation:
- *Graph* characteristics.

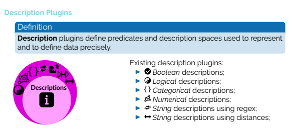
Description Plugins

Definition

Description plugins define predicates and description spaces used to represent and to define data precisely.

Existing description plugins:

- ✓ *Boolean* descriptions;
- ☯ *Logical* descriptions;
- { } *Categorical* descriptions;
- *Numerical* descriptions;
- ⇄ *String* descriptions using regex;
- ↔ *String* descriptions using distances;


Description Plugins

Definition

Description plugins define predicates and description spaces used to represent and to define data precisely.

Existing description plugins:

- ✓ *Boolean* descriptions;
- ☯ *Logical* descriptions;
- { } *Categorical* descriptions;
- *Numerical* descriptions;
- ⇄ *String* descriptions using regex;
- ↔ *String* descriptions using distances;

Description Plugins

Definition

Description plugins define predicates and description spaces used to represent and to define data precisely.

Existing description plugins:

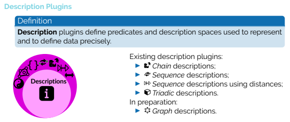
- *Chain* descriptions;
- *Sequence* descriptions;
- *Sequence* descriptions using distances;
- *Triadic* descriptions.

In preparation:

- *Graph* descriptions.

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Description Plugins


Description Plugins

Definition

Description plugins define predicates and description spaces used to represent and to define data precisely.

Existing description plugins:
- *Chain* descriptions;
- *Sequence* descriptions;
- *Sequence* descriptions using distances;
- *Triadic* descriptions.

In preparation:
- *Graph* descriptions.

Strategy Plugins

Definition

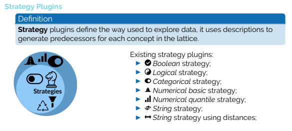
Strategy plugins define the way used to explore data, it uses descriptions to generate predecessors for each concept in the lattice.

Existing strategy plugins:

- ✓ *Boolean* strategy;
- ☯ *Logical* strategy;
- ◐ *Categorical* strategy;
- ▲ *Numerical basic* strategy;
- ▮ *Numerical quantile* strategy;
- ⇄ *String* strategy;
- ↔ *String* strategy using distances;

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Strategy Plugins


Strategy Plugins
Definition
Strategy plugins define the way used to explore data, it uses descriptions to generate predecessors for each concept in the lattice.
Existing strategy plugins:
Boolean strategy;
Logical strategy;
Categorical strategy;
Numerical basic strategy;
Numerical quantile strategy;
String strategy;
String strategy using distances;

Strategy Plugins

Definition

Strategy plugins define the way used to explore data, it uses descriptions to generate predecessors for each concept in the lattice.

Existing strategy plugins:

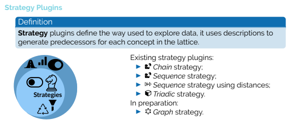
- *Chain* strategy;
- *Sequence* strategy;
- *Sequence* strategy using distances;
- *Triadic* strategy.

In preparation:

- *Graph* strategy.

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Strategy Plugins



Strategy Plugins

Definition

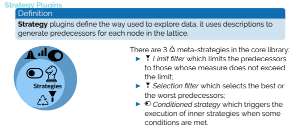
Strategy plugins define the way used to explore data, it uses descriptions to generate predecessors for each node in the lattice.

There are 3 ♲ meta-strategies in the core library:

- ⏷ *Limit filter* which limits the predecessors to those whose measure does not exceed the limit;
- ⏷ *Selection filter* which selects the best or the worst predecessors;
- ◐ *Conditioned strategy* which triggers the execution of inner strategies when some conditions are met.

2023-01-17

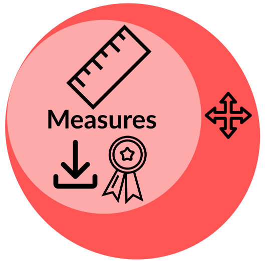
GALACTIC architecture
└Architecture
 └Plugins
  └Strategy Plugins


Strategy Plugins
Definition
Strategy plugins define the way used to explore data, it uses descriptions to generate predecessors for each node in the lattice.
Strategies
There are 3 ♲ meta-strategies in the core library:
- ⏷ *Limit filter* which limits the predecessors to those whose measure does not exceed the limit;
- ⏷ *Selection filter* which selects the best or the worst predecessors;
- ◐ *Conditioned strategy* which triggers the execution of inner strategies when some conditions are met.

Measure Plugins

Definition

Measure plugins are parameters of the *filter strategies* predefined in the core library.



There are 3 measures in the core library:

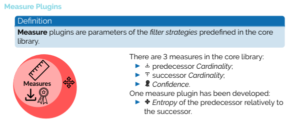
- ⤓ predecessor *Cardinality*;
- ⤒ successor *Cardinality*;
- 🎖 *Confidence*.

One measure plugin has been developed:

- ✥ *Entropy* of the predecessor relatively to the successor.

2023-01-17

GALACTIC architecture
└Architecture
  └Plugins
    └Measure Plugins


Measure Plugins

Definition

Measure plugins are parameters of the *filter strategies* predefined in the core library.

Measures

There are 3 measures in the core library:
- ⤓ predecessor *Cardinality*;
- ⤒ successor *Cardinality*;
- 🎖 *Confidence*.

One measure plugin has been developed:
- ✥ *Entropy* of the predecessor relatively to the successor.

Data Reader Plugins

Definition

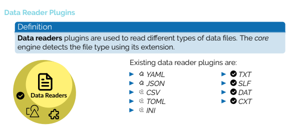
Data readers plugins are used to read different types of data files. The *core* engine detects the file type using its extension.

Existing data reader plugins are:

- 🧩 *YAML*
- 🧩 *JSON*
- △ *CSV*
- △ *TOML*
- △ *INI*
- ✓ *TXT*
- ✓ *SLF*
- ✓ *DAT*
- ✓ *CXT*

2023-01-17

GALACTIC architecture
└Architecture
 └Plugins
  └Data Reader Plugins


Data Reader Plugins

Definition

Data readers plugins are used to read different types of data files. The core engine detects the file type using its extension.

Data Readers

Existing data reader plugins are:

- *YAML*
- *JSON*
- *CSV*
- *TOML*
- *INI*
- *TXT*
- *SLF*
- *DAT*
- *CXT*

Localization Plugins

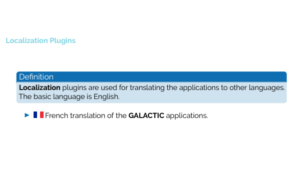
Definition

Localization plugins are used for translating the applications to other languages. The basic language is English.

- 🇫🇷 French translation of the **GALACTIC** applications.

2023-01-17

GALACTIC architecture
└Architecture
└Plugins
└Localization Plugins



Localization Plugins

Definition

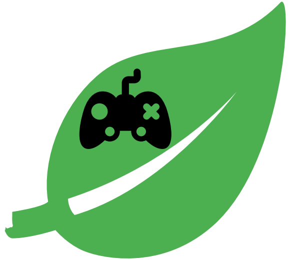
Localization plugins are used for translating the applications to other languages. The basic language is English.

- 🇫🇷 French translation of the **GALACTIC** applications.

Applications

Definition

Applications are developed for using the library; they are the interface of the user.

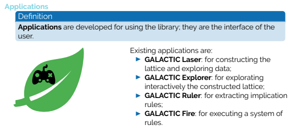


Existing applications are:

- **GALACTIC Laser**: for constructing the lattice and exploring data;
- **GALACTIC Explorer**: for explorating interactively the constructed lattice;
- **GALACTIC Ruler**: for extracting implication rules;
- **GALACTIC Fire**: for executing a system of rules.

2023-01-17

GALACTIC architecture
└Architecture
└Applications
└Applications



Applications

Definition

Applications are developed for using the library; they are the interface of the user.

Existing applications are:

- **GALACTIC Laser**: for constructing the lattice and exploring data;
- **GALACTIC Explorer**: for explorating interactively the constructed lattice;
- **GALACTIC Ruler**: for extracting implication rules;
- **GALACTIC Fire**: for executing a system of rules.

Jupyter notebooks

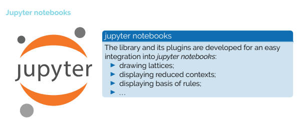
jupyter notebooks

The library and its plugins are developed for an easy integration into *jupyter notebooks*:

- drawing lattices;
- displaying reduced contexts;
- displaying basis of rules;
- ...

2023-01-17

GALACTIC architecture
└Architecture
  └Jupyter notebooks
    └Jupyter notebooks



Jupyter notebooks

jupyter notebooks

The library and its plugins are developed for an easy integration into *jupyter notebooks*:

- drawing lattices;
- displaying reduced contexts;
- displaying basis of rules;
- ...

Collaborative version control

git

The library is developed using the collaboration tool `git`, in the `gitlab` of the university. We are using

- ***pylint*** and ***flake8*** (with plugins) for testing code quality;
- ***tox*** for generating tests.

2023-01-17

GALACTIC architecture
└Architecture
 └Collaborative version control
  └Collaborative version control

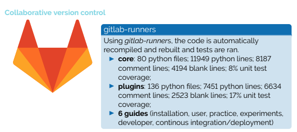
Collaborative version control
git
The library is developed using the collaboration tool git, in the gitlab of the university. We are using
- pylint and flake8 (with plugins) for testing code quality;
- tox for generating tests.

Collaborative version control

gitlab-runners

Using *gitlab-runners*, the code is automatically recompiled and rebuilt and tests are ran.

- **core**: 80 python files; 11949 python lines; 8187 comment lines; 4194 blank lines; 8% unit test coverage;
- **plugins**: 136 python files; 7451 python lines; 6634 comment lines; 2523 blank lines; 17% unit test coverage;
- **6 guides** (installation, user, practice, experiments, developer, continous integration/deployment)


Collaborative version control

gitlab-runners

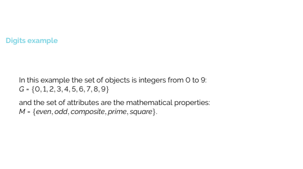
Using *gitlab-runners*, the code is automatically recompiled and rebuilt and tests are ran.

- **core**: 80 python files; 11949 python lines; 8187 comment lines; 4194 blank lines; 8% unit test coverage;
- **plugins**: 136 python files; 7451 python lines; 6634 comment lines; 2523 blank lines; 17% unit test coverage;
- **6 guides** (installation, user, practice, experiments, developer, continous integration/deployment)

Digits example

In this example the set of objects is integers from 0 to 9: $G = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

and the set of attributes are the mathematical properties: $M = \{even, odd, composite, prime, square\}$.



Digits example

In this example the set of objects is integers from 0 to 9: $G = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

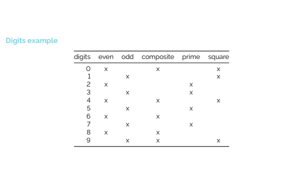
and the set of attributes are the mathematical properties: $M = \{even, odd, composite, prime, square\}$.

Digits example

digits	even	odd	composite	prime	square
0	x		x		x
1		x			x
2	x			x	
3		x		x	
4	x		x		x
5		x		x	
6	x		x		
7		x		x	
8	x		x		
9		x	x		x

2023-01-17

GALACTIC architecture
└Examples
  └Digits example
    └Digits example

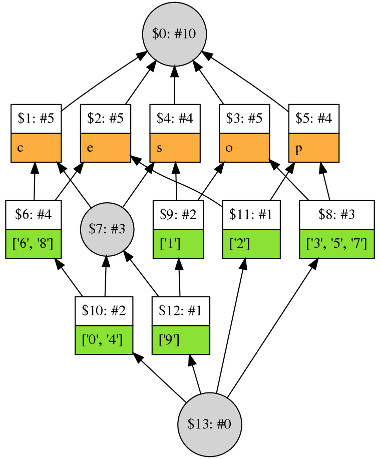


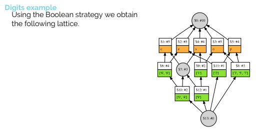
Digits example

digits	even	odd	composite	prime	square
0	x		x		x
1		x			x
2	x			x	
3		x		x	
4	x		x		x
5		x		x	
6	x		x		
7		x		x	
8	x		x		
9		x	x		x

Digits example

Using the Boolean strategy we obtain the following lattice.




Digits example
Using the Boolean strategy we obtain the following lattice.

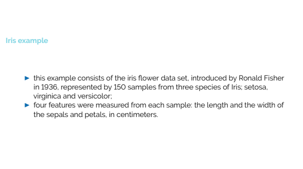
What's the Boolean strategy? Precise that it corresponds to the "classical" concept lattice.

Iris example

- this example consists of the iris flower data set, introduced by Ronald Fisher in 1936, represented by 150 samples from three species of Iris; setosa, virginica and versicolor;
- four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

2023-01-17

GALACTIC architecture
└Examples
  └Iris example
    └Iris example



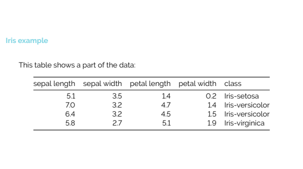
Iris example

- this example consists of the iris flower data set, introduced by Ronald Fisher in 1936, represented by 150 samples from three species of Iris; setosa, virginica and versicolor;
- four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

Iris example

This table shows a part of the data:

sepal length	sepal width	petal length	petal width	class
5.1	3.5	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.8	2.7	5.1	1.9	Iris-virginica



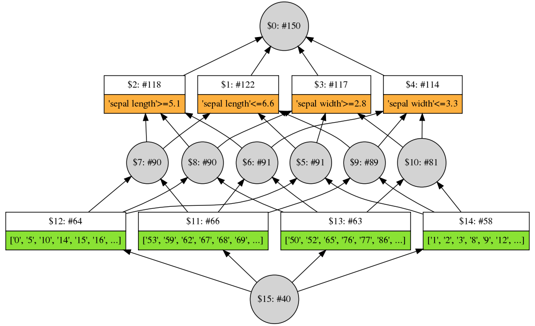
Iris example

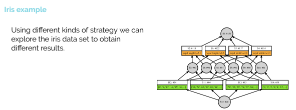
This table shows a part of the data:

sepal length	sepal width	petal length	petal width	class
5.1	3.5	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.8	2.7	5.1	1.9	Iris-virginica

Iris example

Using different kinds of strategy we can explore the iris data set to obtain different results.

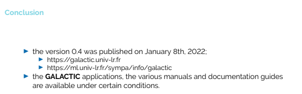



Iris example

Using different kinds of strategy we can explore the iris data set to obtain different results.

Conclusion

- the version 0.4 was published on January 8th, 2022;
  - https://galactic.univ-lr.fr
  - https://ml.univ-lr.fr/sympa/info/galactic
- the **GALACTIC** applications, the various manuals and documentation guides are available under certain conditions.



Conclusion

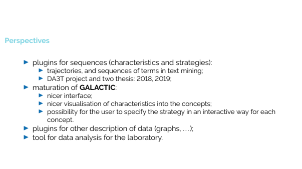
- the version 0.4 was published on January 8th, 2022;
  - https://galactic.univ-lr.fr
  - https://ml.univ-lr.fr/sympa/info/galactic
- the **GALACTIC** applications, the various manuals and documentation guides are available under certain conditions.

Perspectives

- plugins for sequences (characteristics and strategies):
  - trajectories, and sequences of terms in text mining;
  - DA3T project and two thesis: 2018, 2019;
- maturation of **GALACTIC**:
  - nicer interface;
  - nicer visualisation of characteristics into the concepts;
  - possibility for the user to specify the strategy in an interactive way for each concept.
- plugins for other description of data (graphs, …);
- tool for data analysis for the laboratory.

2023-01-17

GALACTIC architecture
└Conclusion
└Conclusion
└Perspectives

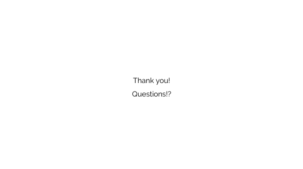


Perspectives

- plugins for sequences (characteristics and strategies):
  - trajectories, and sequences of terms in text mining;
  - DA3T project and two thesis: 2018, 2019;
- maturation of **GALACTIC**:
  - nicer interface;
  - nicer visualisation of characteristics into the concepts;
  - possibility for the user to specify the strategy in an interactive way for each concept.
- plugins for other description of data (graphs, …);
- tool for data analysis for the laboratory.

Thank you!

Questions!?



Thank you!

Questions!?