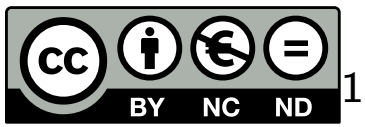
Sequential plugins

Strings, Chains, Sequences

The Galactic Organization <contact@thegalactic.org>

2019-2020



[1]

La Rochelle Université

Sequence types

Sequences

Sequential plugins are plugins developed for the *Galactic* library to analyse sequential data.
We distinguish three types.

- Sequences 
- Chains 
- Strings 

Sequences

Sequences are constructed from a set of elements of the same type called **Alphabet**

For example, DNA is composed of a succession of nucleotides. There are four different nucleotides that constitute the alphabet,

$\Sigma = \{adenosine\,(A),\ cytidine\,(C),\ guanosine\,(G),\ thymidine\,(T)\}.$

Sequences

A **sequence** S is an ordered list of couples, $S =< s_1, s_2, ..., s_n >$, where every s_i is a couple formed of:

- an identifier $S[i].id$
- an element $S[i].e$.

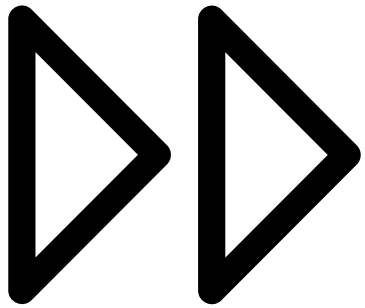


Figure 1: Sequence

Chains

Chains are sequences without id. For a set of alphabets, a chain C is an ordered list of elements.

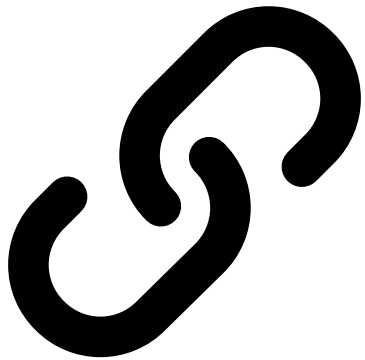


Figure 2: Chain

Strings

String are sequences of characters, it's important to consider this data type separately because regular expressions are very useful for such type of data.

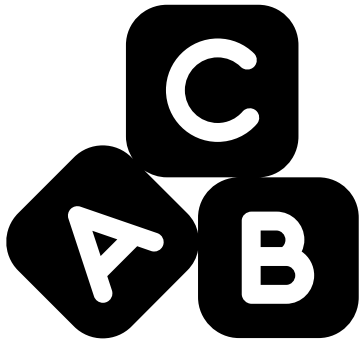


Figure 3: String

Sequential descriptions

We can resume sequential plugins in this table :

	String	Chain	Sequence
Simple	✓	✓	✓
Complete	✓	✓	
Prefix		✓	✓
Distance	✓		✓

Simple match description

For a given length **l**, the description is the set of all subsequences of length $= \mathbf{l}$

Wakeup,Breakfast,**Dinner**,Read
Wakeup,Sports,Read,**Dinner**
Wakeup,Work,Sports,**Dinner**

---

l=2

Also, we can specify :

- **window** : searching for subsequence in a sliding window
- **gaps** : for specify a maximal gap between elements.

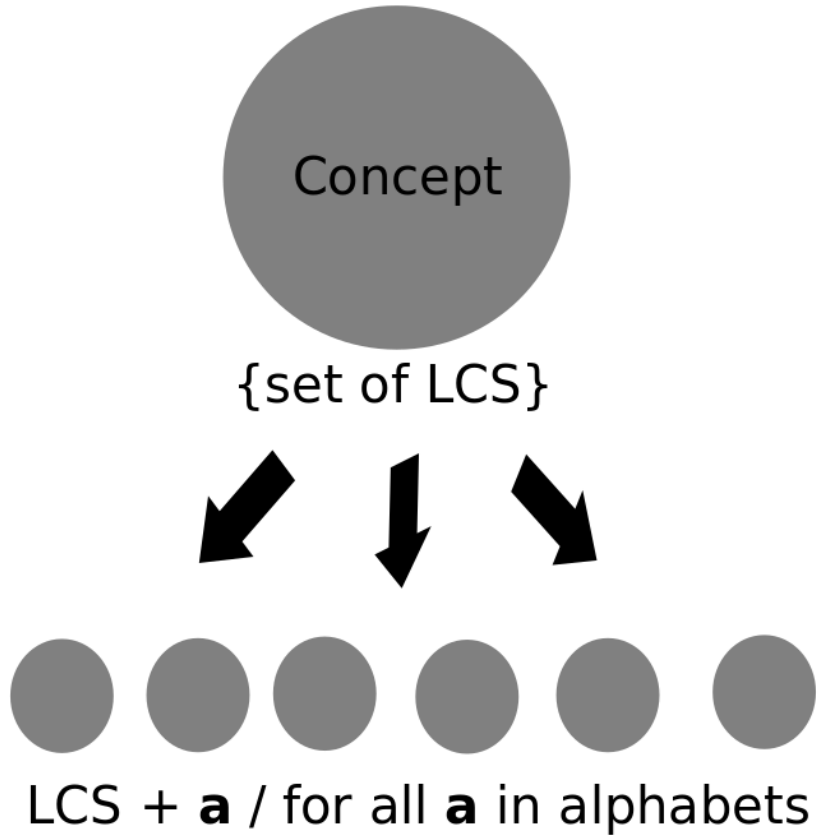
Complete match description

The description is the set of all longest common subsequences (LCS).

A**BCDE**
F**BD**CA**E**
DC**B**K**D**A**E**

Complete match strategy

At each concept, we take the set of longest common subsequences, and we generate predicates by adding one alphabet.



Distance match description

For a given length **l**, the description is the set of all subsequences of length = **l**, with **distances** (minimal and maximal) between elements, and **cardinality** (minimal and maximal)

For **l=2**

A,D,**B**,**A**,**B**
A,C,**B**
A,E,Z,**B**

Description :
sequences match A [0, 2] B from 1 to 2 times.

Prefix match description

The description is the common prefix of the set of sequences

M00,**M02**,M03,M04
M00,**M02**,M04,M05,M08
M00,**M02**,M03,M05,M06

Prefix match strategy

At each concept, we take the prefix and we generate predicates by adding one alphabet.

Daily actions

Consider a sequence database, that we call the *daily-actions database* constructed from daily actions of a person.

We asked the members of the L3i[2] to make a sequence of their daily actions when they are in days of work, or in holidays.

[2] https://l3i.univ-larochelle.fr/

Daily actions

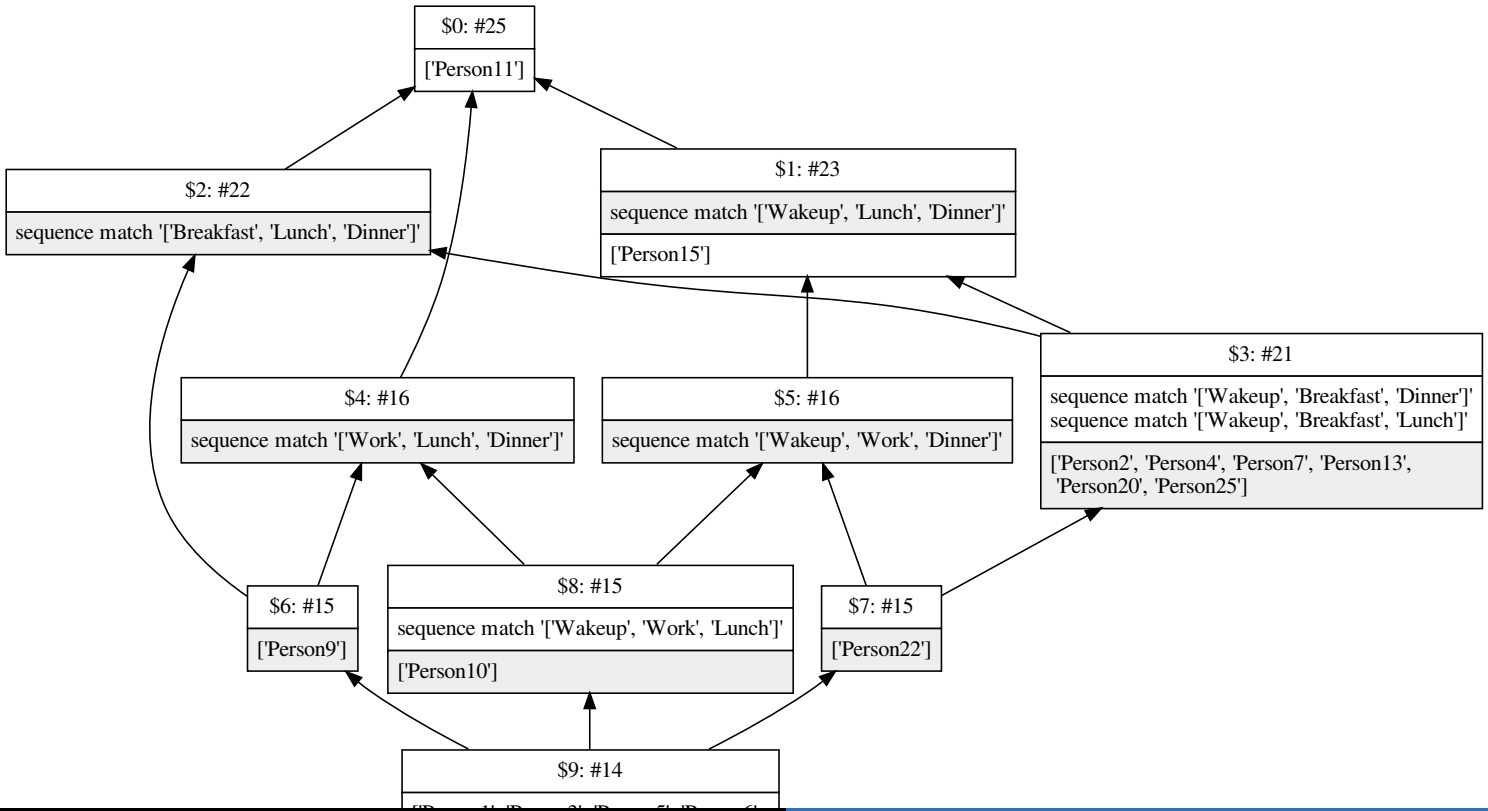
We limited our alphabet to this set of actions :

$\Sigma = \{Wakeup, Breakfast, Work, Coffee, Lunch, Sports, Dinner, Read, Rest, Sleep, Other\}$

person	Sequence
Person1	{8: Wakeup, 9: Breakfast, 10: Work, 12: Coffee, 13: Lunch, 20: Sports, 22: Dinner, 23: Read}
Person2	{10: Wakeup, 11: Breakfast, 14: Lunch, 20: Sports, 21: Read, 22: Dinner}
Person4	{11: Wakeup, 12: Breakfast, 14: Lunch, 20: Dinner}

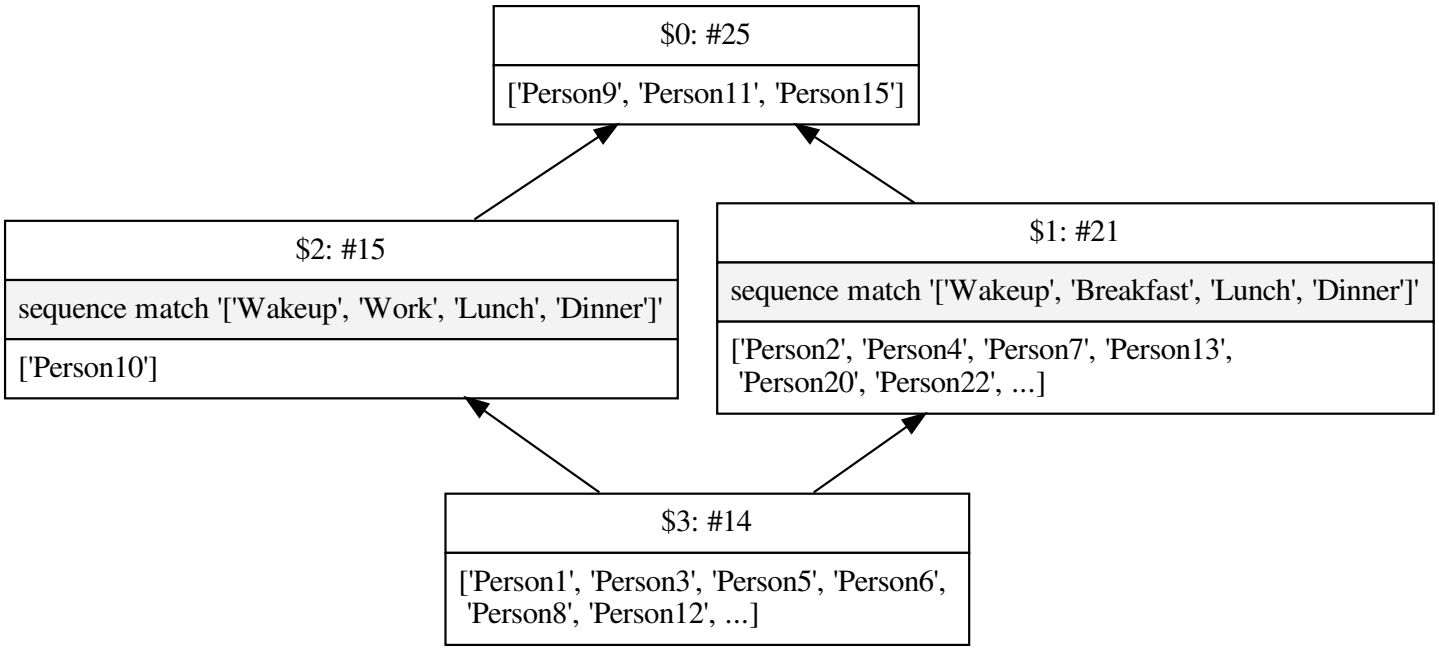
Daily actions with simple match

Simple strategy with length = 3, and LimitFilter strategy with support = 15



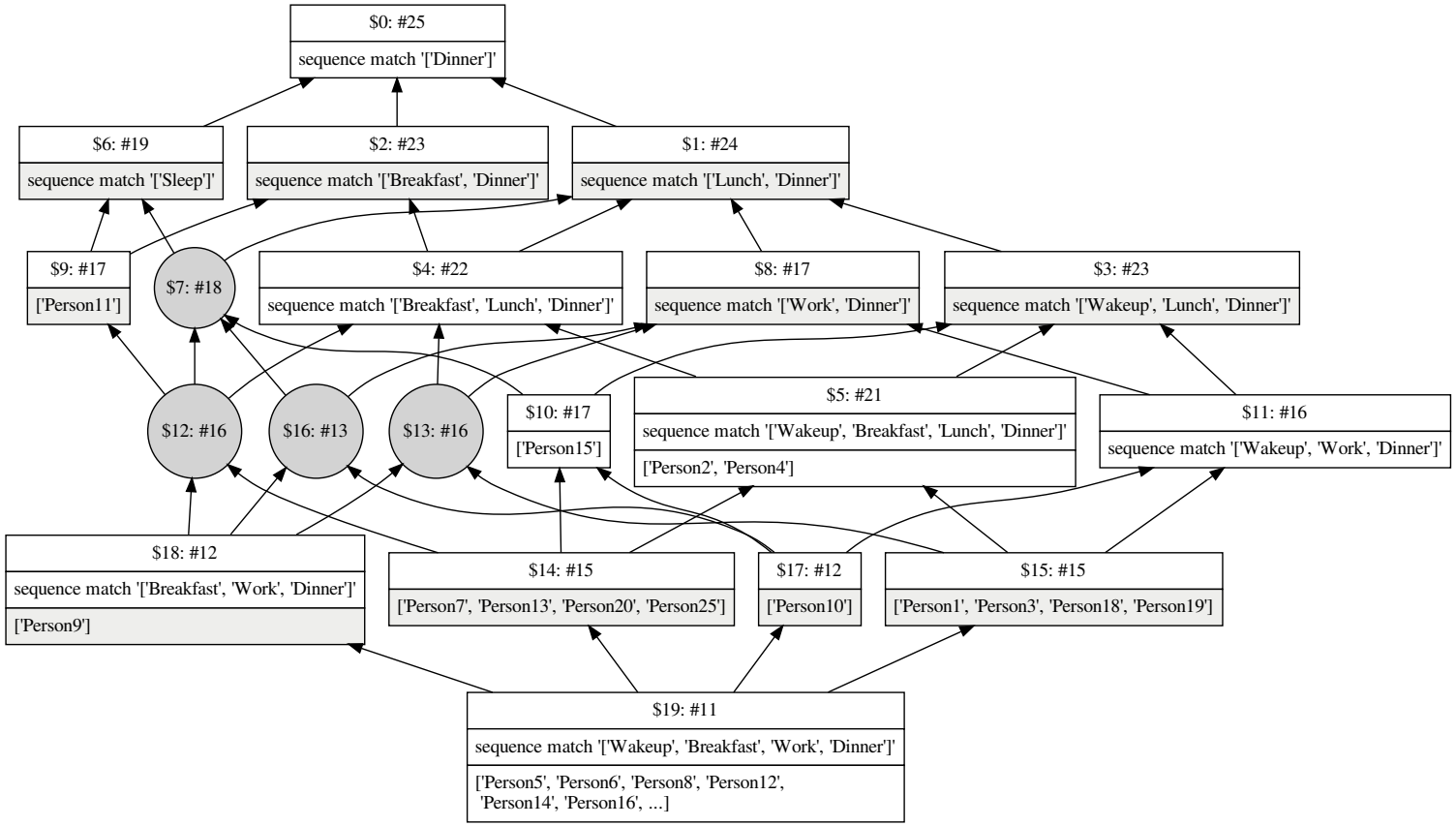
Daily actions with simple match

Simple strategy with length = 4, and LimitFilter strategy with support = 15



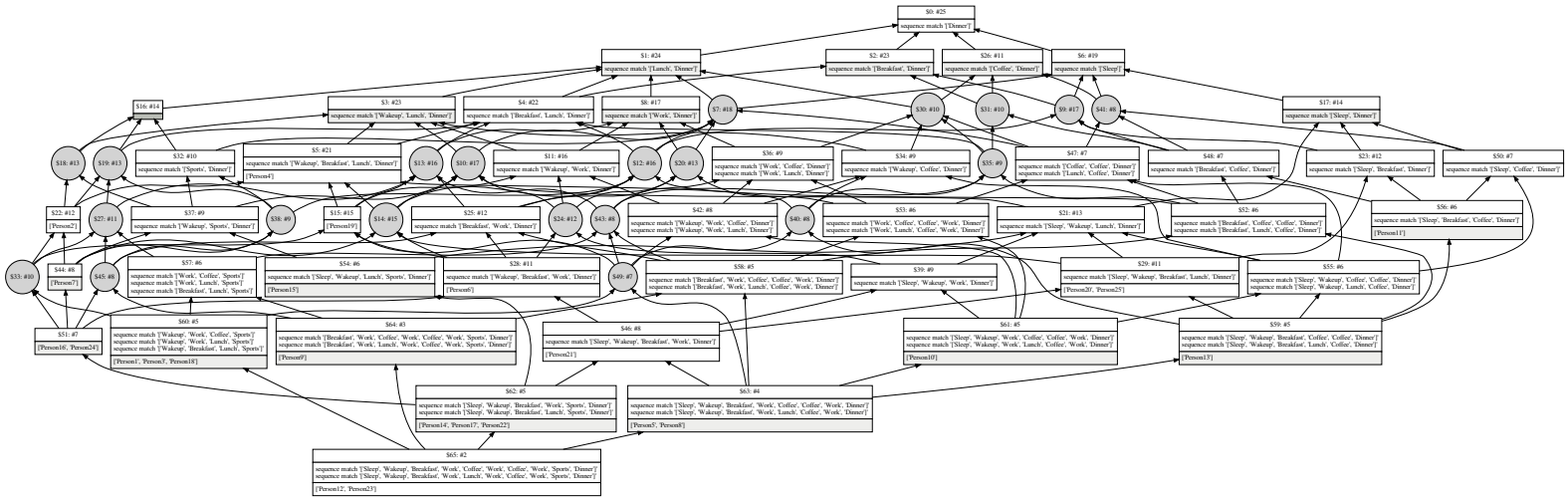
Daily actions with complete match

Using a LimitFilter strategy with support = 15

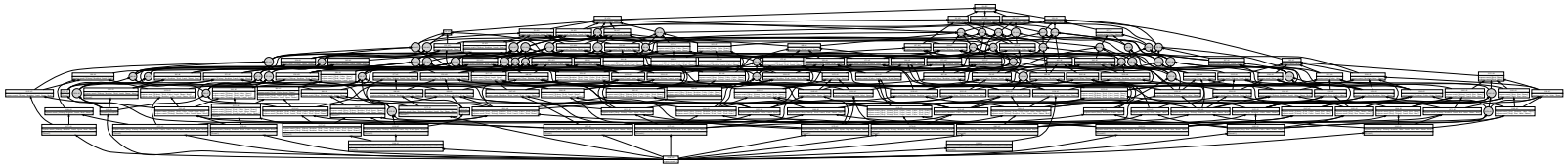


Daily actions with complete match

LimitFilter strategy with support = 10

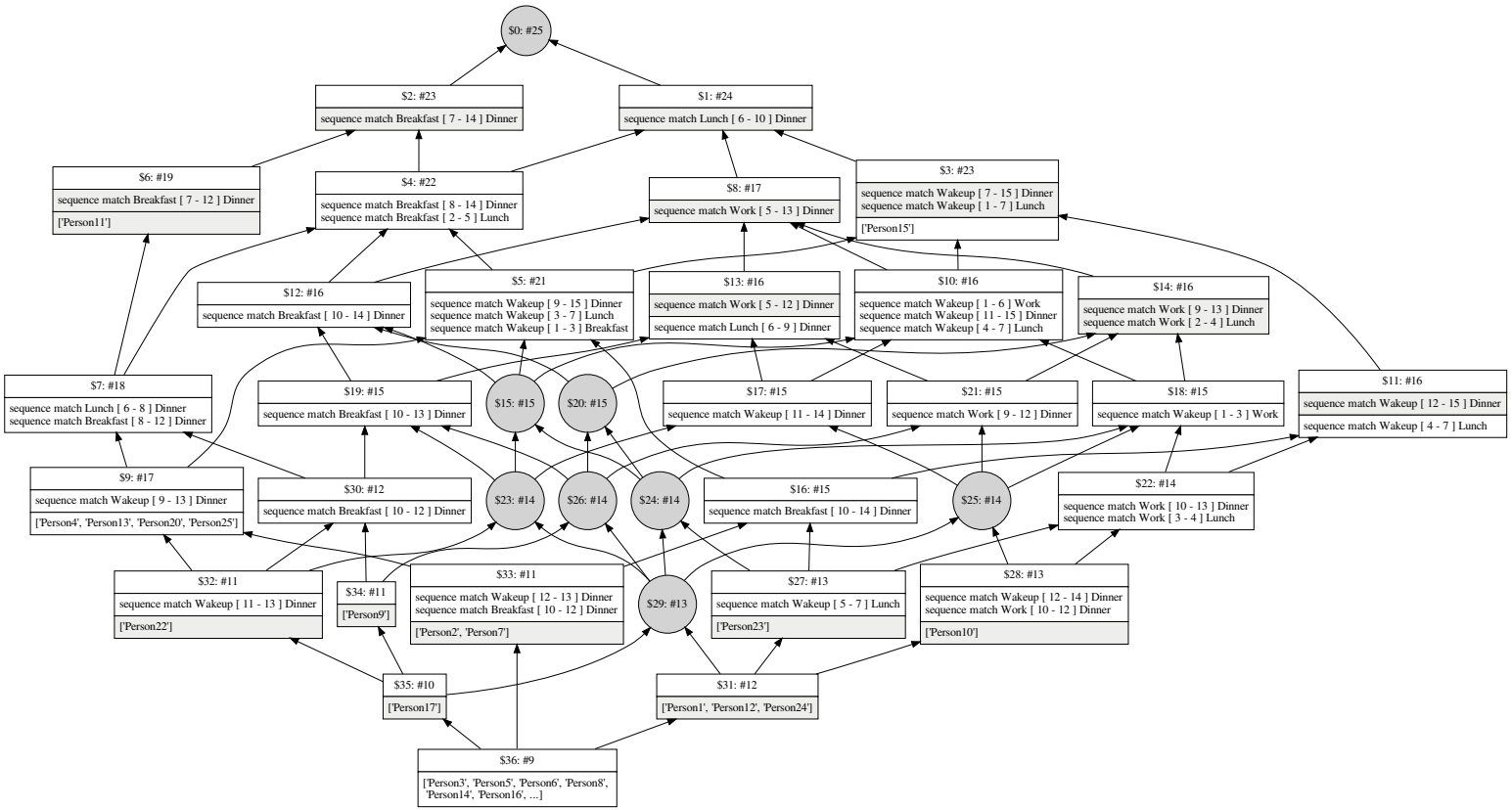


Daily actions with complete match



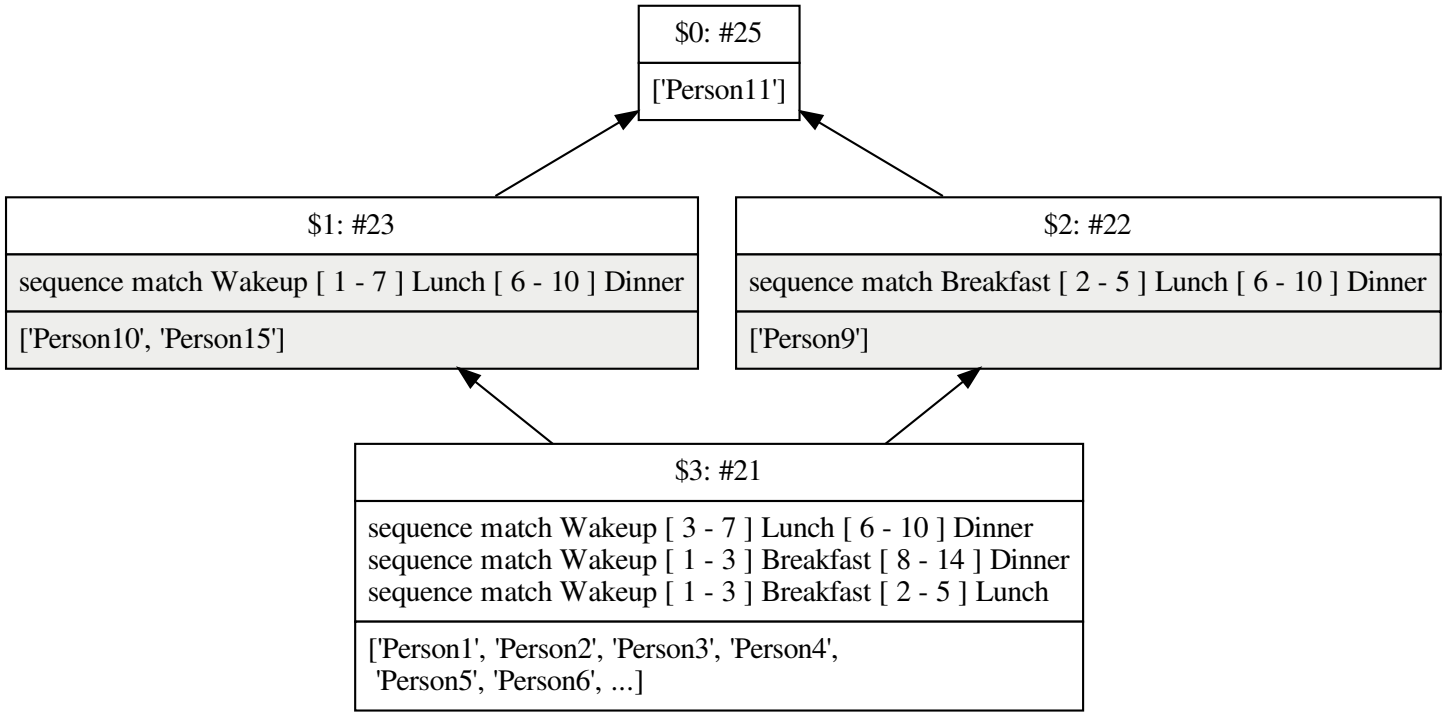
Daily actions with distance match

Simple strategy with length = 2, and LimitFilter strategy with support = 15



Daily actions with distance match

Simple strategy with length = 3, and LimitFilter strategy with support = 18



Wine City

- The data is form the Wine City.
- Gathered from the visits on a period of one year (May 2016 to May 2017).
- The data has been cleaned and processed before.
- Visitors navigate from modules to modules exploring the museum.
- The museum is open, and they are not “Guided”.

Wine City

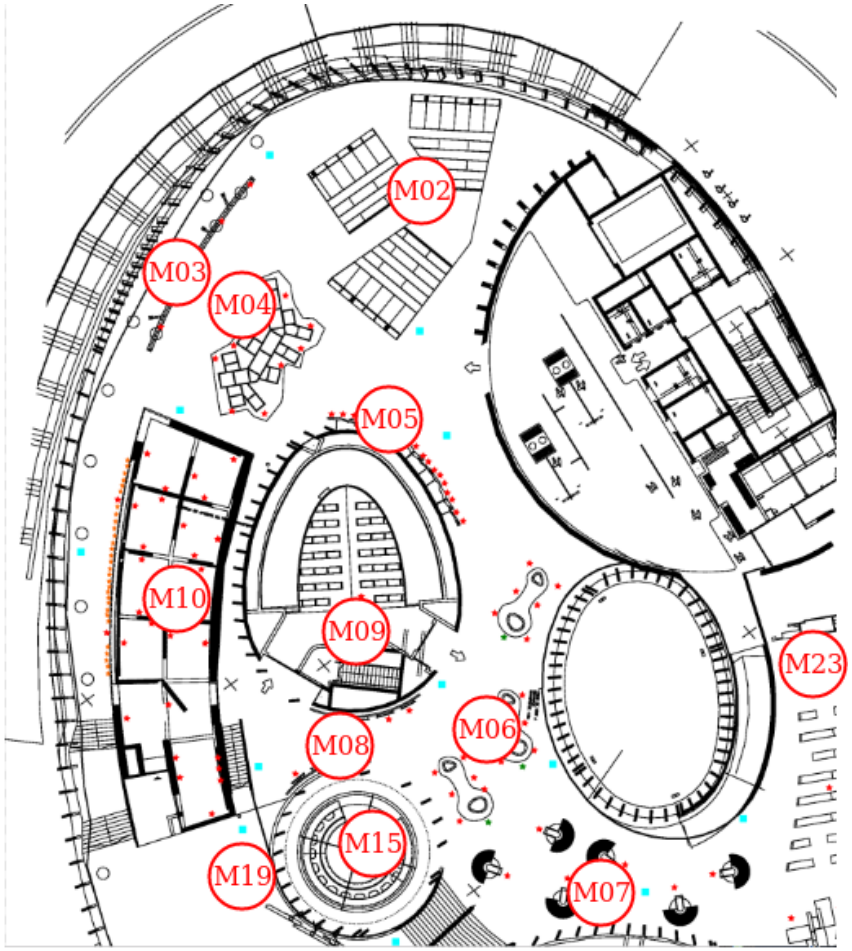
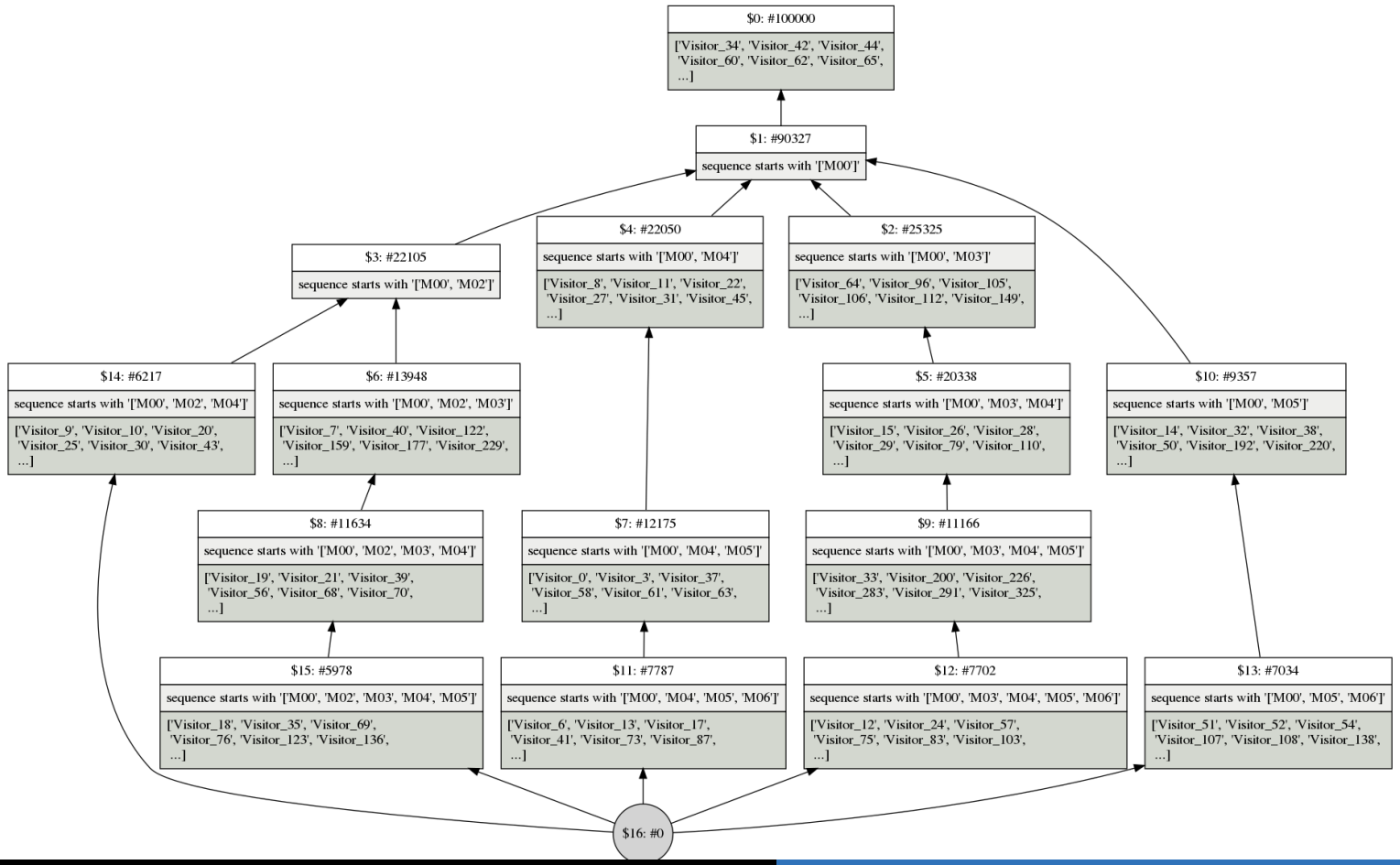


Figure 4: Modules location in the museum

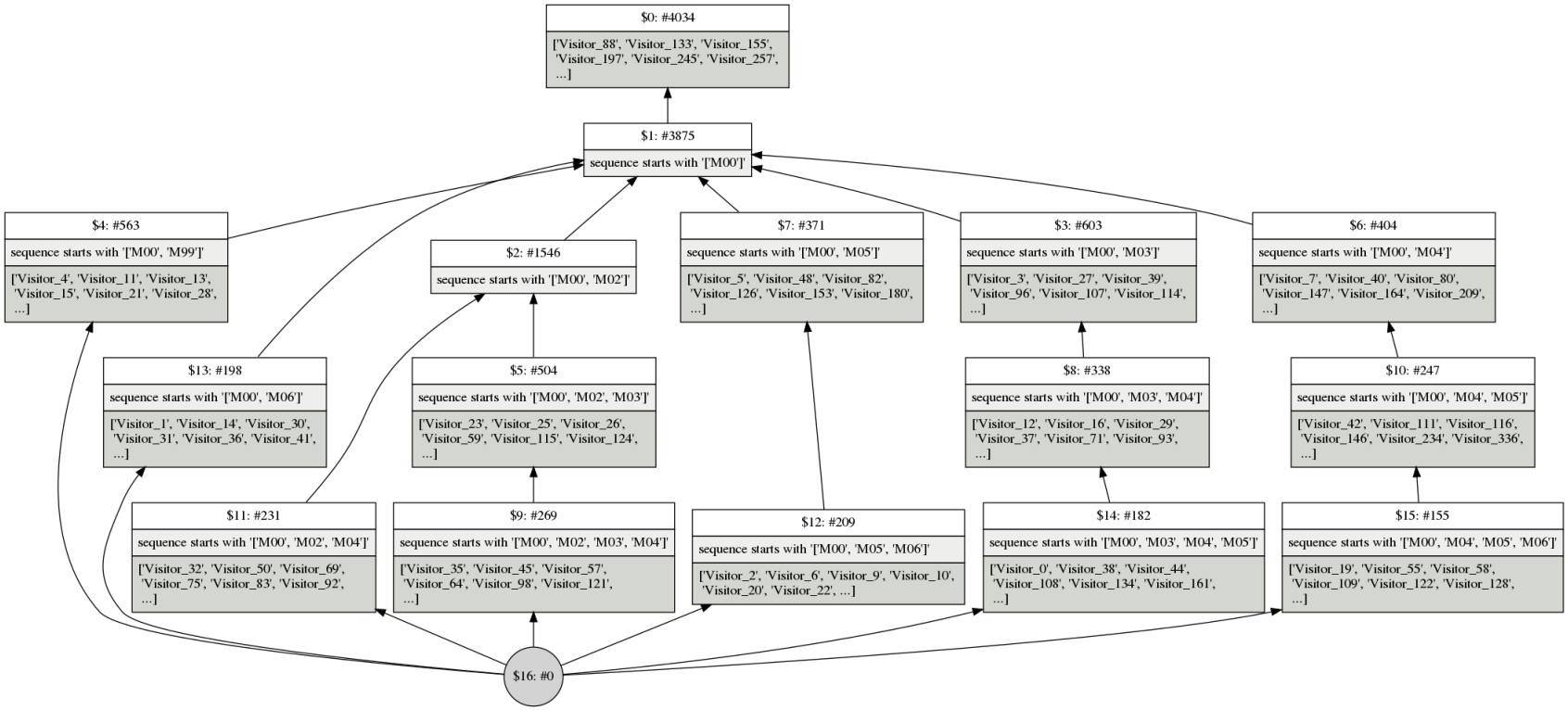
Wine City with prefix match

Using LimitFilter strategy with support = 5000 (100000 visitor)



Wine City with prefix match

LimitFilter strategy with support = 150 (~9000 visitor)



Installation

Install Graphviz

```
$ conda install -c anaconda graphviz
```

Install Galactic

```
$ pip install --upgrade --find-links
https://galactic.univ-lr.fr/packages py-galactic
```

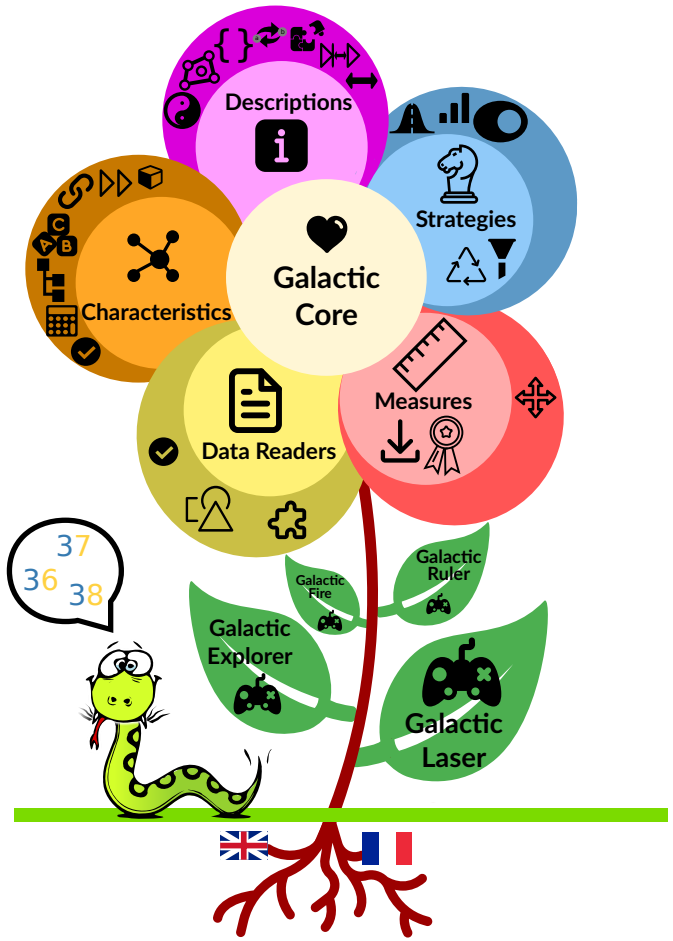


Figure 5: Galactic architecture