Sequence clustering with Galactic

K. Bertet C. Demko

Objective: The objective of this practical work is to generate and visualize concepts/clusters of sequences using GALACTIC.

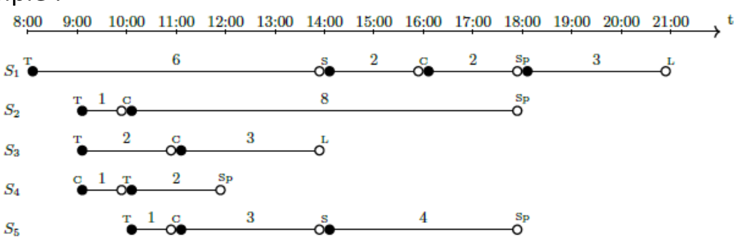
1. Visualization of temporal sequencies

Let us consider the following dataset involving temporal sequences of daily actions:

{Work/Travail(T) ; Siesta/Sieste(S) ; Coffee/Café(C) ; Sport(Sp); Reading/Lecture(L)}

	Name	Daily actions
s_1	Minh	s_1 = <(T,8), (S,14), (C,16), (Sp,18), (L,21) >
s_2	Do	s_2 = <(T,9), (C,10), (Sp,18) >
s_3	Julien	s_3 = <(T,9), (C,11), (L,14) >
s_4	Vu	s_4 = <(C,9),(T,10), (Sp,12) >
s_5	Thanh	s_5 = <(T,10), (C,11), (S,14), (Sp,18) >

Implement a *function draw-sequencies (sequencies)* to visualize a list of temporal sequencies using matplotlib. For example :



Consider the following dataset involving sequences without temporal information:

	Name	Daily actions
s_1	Minh	s_1 = <T, S, C, Sp, L >
s_2	Do	s_2 = <T, C, Sp >
s_3	Julien	s_3 = <T, C, L >
s_4	Vu	s_4 = <C, T, Sp >
s_5	Thanh	s_5 = < T, C, S, Sp >

We will denote by Σ the dictionary of daily actions, and by M the set of sequences. Calculate the following concepts using the description of a set of sequences by their maximal common subsequences. More formally, for a set of sequences $A \subseteq M$, the description by **maximal common subsequences** (SCM) is defined by:

$$\delta_{SCM}(A) = \{\text{s match X} \mid \text{X} \in \Sigma^* \text{ maximal common subsequence of A}\}$$

- $\delta_{SCM}(s_2, s_3)$
- $\delta_{SCM}(s_3)$
- $\delta_{SCM}(s_1, s_2, s_5)$
- $\delta_{SCM}(s_1, s_2, s_3, s_4, s_5)$
- $\delta_{SCM}(s_1, s_4)$

The concept lattice (or hierarchy of patterns) is given by Fig 1 in a reduced representation.

- Retrieve each complete concept $(A, \delta_{SCM}(A))$ from its reduced form.

- Represent the binary table that is representative of this lattice, with the sequences in rows and the generated predicates in columns.

A naive algorithm to calculate this lattice would be to compute the description $\delta_{SCM}(A)$ for each subset $A \subseteq M$. What would be the theoretical complexity of this algorithm, knowing that the calculation of common subsequences is an NP-complete problem?

The algorithm NextPriorityConcept is a hierarchy generation algorithm by division. The **root concept** is first calculated:

$$(M,\ \delta_{SCM}(M) = \{s\ match\ < T >, s\ match\ < C >\})$$

Then **candidate subgroups** $A \subseteq M$ are computed, where each candidate subgroup A satisfies a selector predicate obtained by adding a new element $a \in M$ to one of the two predicates of $\delta_{SCM}(M)$:

s match <T,a>, s match <a,T>, s match <C,a>, s match <a,C>

The predecessors of the root concept are selected as the maximal subgroups of these candidate subgroups $A \subseteq M$, then each corresponding concept $(A, \delta_{SCM}(A))$ is calculated.

1. Calculate all the candidate subsets $A \subseteq M$
2. Select those that maximize A
3. Deduce the concepts $(A, \delta_{SCM}(A))$ corresponding to the childs/predecessors of the root

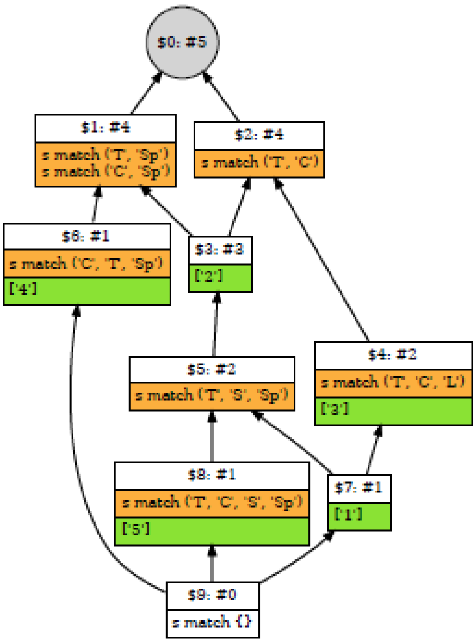


Figure 1. Concept lattice for the description δ_{SCM} and the stategy σ_{SN}

[To go further] In general, for a concept $(A, \delta_{SCM}(A))$, its predecessors are the concepts $(A', \delta_{SCM}(A'))$ where the subgroups $A' \subset A$ are the maximal subgroups that satisfy the predicates defined by the strategy σ_{SN} :

$$\sigma_{SN}(A) = \{s\ match < x_1..\boldsymbol{a}, x_j..x_k > | < x_1..x_k > \in \delta_{SCM}(A), a \in \Sigma \text{ and } 1 \leq \text{j} \leq k+1\}$$

The strategy σ_{SN} is called the **naive strategy** (SN), in the sense that it allows generating all the concepts. It is possible to define other strategies that generate fewer concepts, for example, the **augmented strategy** (SA) σ_{SA} :

$$\sigma_{SA}(A) = \{s\ match < x, a > | \, x \in \delta(A)\ and\ a \in \Sigma\}$$

- Calculate the predecessor concepts of the concept $(\{s_1, s_3\}, \{s\ match < T, C, L >\})$ using each of these two strategies.
- Calculate the concept lattice using the augmented strategy σ_{SA}

[To go further] It is also possible to consider less costly descriptions than the description by maximal common subsequences:
The description by **prefix common subsequence** (SCP):

$$\delta_{SCP}(A) = \{\text{s match X} \mid \text{X} \in \Sigma^* \text{prefix common subsequence of sequences of A A}\}$$

The description by **common subsequences of size k** (KSC), with k provided as a parameter:

$$\delta_{SCP}(A, k) = \{\text{s match X} \mid \text{X} \in \Sigma^* \text{sous-séquence de taille k des séquences de A}\}$$

Calculate the lattice obtained with the prefix description δ_{SCP} and the augmented strategy σ_{SA}

[To go further] We now wish to analyze sequences with temporal information.

	Name	Daily actions
1	Minh	s_1 =<(T,8), (S,14), (C,16), (Sp,18), (L,21) >
2	Do	s_2 =<(T,9), (C,10), (Sp,18) >
3	Julien	s_3 =<(T,9), (C,11), (L,14) >
4	Vu	s_4 =<(C,9),(T,10), (Sp,12) >
5	Thanh	s_5 =<(T,10), (C,11), (S,14), (Sp,18) >

For such temporal sequences, adapted descriptions and strategies are defined. The lattice in Figure 2 is generated with the description by **maximal common distance subsequences** (SDCM) and the **naive distance strategy** (SDN).
Provide the description predicates δ_{SDCM} of the following subgroups. What is their interpretation?

- $\{s_1, s_3\}$
- $\{s_1, s_2, s_3, s_5\}$
- $\{s_1, s_4\}$
- $\{s_1, s_4, s_5\}$

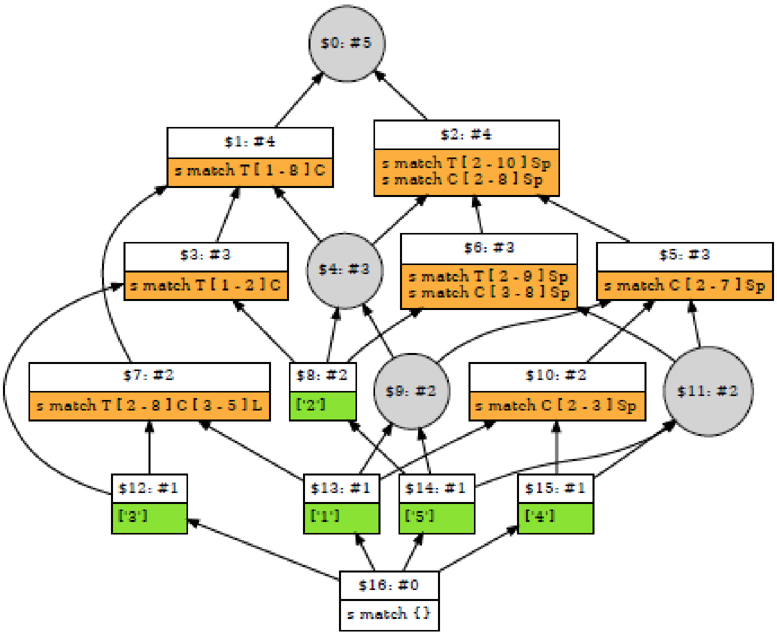


Figure 2 Concept lattice for the description δ_{SDCM} and the strategy σ_{SDN}

2. Analysis of the dataset *Daily-Action* of sequences with Galactic

The data-set *Daily-Action* is composed of sequencies describing daily actions of 25 persons, where daily actions are :

{Wakeup, Breakfast, Work, Coffee, Lunch, Sports, Dinner, Read, Rest, Sleep, Other}

And available here : share/galactic/sequence/data/Daily-Actions/

Analyze this dataset using the following exploration file which specifies descriptions by simple maximal subsequences, and the simple strategy which generates all possible subsequences of lenth equal to 2:
~/.local/share/galactic/sequence/data/explorers/chain/simple-match-basic.yaml

We want to take into account the temporal information of each action in order to refine the analysis. To do so, use the following exploration which specifies:

- the *description.sequence.CompleteDistance* for the descriptions,
- and a naive strategy *!strategy.sequence.distance.basic.NaiveDistance* which generates all possible concepts:

```
characteristics:
- &id001 !characteristic.sequence.Sequence
  characteristic: !characteristic.core.Key
    name: "sequence"
descriptions:
- !description.sequence.CompleteDistance
  - *id001
strategies:
- !strategy.sequence.distance.basic.NaiveDistance
  - *id001
```

Analyse the dataset with the same description, but changing the strategy :

```
characteristics:
- &id001 !characteristic.sequence.Sequence
  characteristic: !characteristic.core.Key
    name: "sequence"
descriptions:
- !description.sequence.CompleteDistance
  - *id001
strategies:
- !strategy.sequence.distance.basic.CompleteDistance
  - *id001
```

[To go further] Compare the generated description predicates, the number of concepts, and the execution time for each of these three analyzes of the sequences of the dataset *daily-action*.

Select the concepts with a support greather than 40%, where the support is :

$$support\ ((A, \delta(A)) = \frac{size\ of\ A}{size\ of\ the\ dataset}$$

Give a visualization of these selected concepts using the function *draw-sequencies* $(\delta(A))$