Measuring and and Scaling in Concept Lattices

Novel Tools in conexp-clj

Tom Hanika

Information Systems and Machine Learning Lab, University of Hildesheim

2025-09-08 @ CONCEPTS 2025

What is conexp-clj

conexp-clj by Borchmann/Hanika [HH19]

Where can I get it? `https://github.com/tomhanika/conexp-clj`

Statistics

- Clojure software
- Implements a wide range of FCA and more.
- Last updated: Aug 2025, created May 2009

Basic Functionality

You can run either a clojure program, some IDE for clojure, or interact with the REPL directly. The latter you can achieve by[1]:

```
java -jar conexp-clj-2.8.0-standalone-openjdk-17.jar
```

Load a formal context (cxt):

```
(def cxt1 (read-context "sp-food.cxt"))
```

Compute all formal concepts:

```
(def cs1 (concepts cxt1))
```

Draw Concept Lattice:

```
(use 'conexp.gui.draw)
(draw-concept-lattice cxt1)
```

Compute Canonical Base:

```
(canonical-base cxt1)
```

[1] `https://github.com/tomhanika/conexp-clj/releases/download/v2.8.0/conexp-clj-2.8.0-standalone-openjdk-17.jar`

Extended Functionality

- default namespace of `conexp-clj` is called `main`
- all analytical tools are available using the namespace `analysis`
- this namespace can be loaded using
  1. **(require 'conexp.analysis)**
  2. **(ns conexp.analysis)**

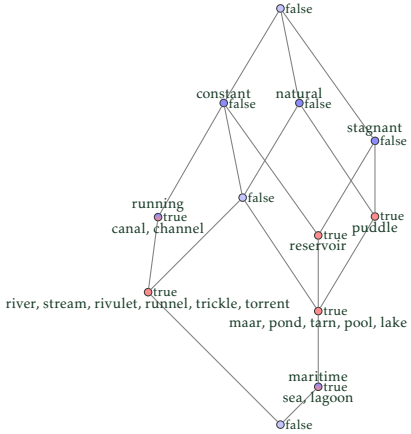
Certain functionalities are not loaded by default, e.g.:

- **(require 'fca.ordinal-motifs)** — ordinal motifs (nominal, ordinal, etc.)
- **(require 'fca.metric-contexts)** — contexts with metric on G or M
- **(require 'fca.non-monotonic-contexts)** – contexts with order on G or M
- **(require 'fca.smeasure)** — scale measures

Metrics

Distributivity

- **(distributive? (concept-lattice cxt))**
  `false`
- **(distributivity-degree (concept-lattice cxt))**
  110/117
- **(distributive-triples (concept-lattice cxt))**
  will tell you which triples are in distributivity
- **(modular? (concept-lattice cxt))**
  `false`
- **(modularity-degree (concept-lattice cxt))**
  1307/1320

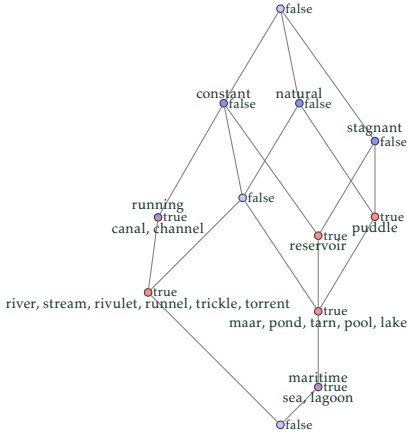


More to come during the next days, e.g., Birkhoff Completion

Entropy [HKS19]

Attribute information entropy is defined by $\sum_{m \in M} 1 - m''/|M|)/|M|$.

- **(attribute-information-entropy cxt)**
  16/25
- **(object-information-entropy cxt)**
  182/289
- **(shannon-attribute-information-entropy cxt)**
  2.18
- **(shannon-object-information-entropy cxt)**
  8.59
- **(modularity-degree (concept-lattice cxt))**
  1307/1320
- **(n-maximal-relevant cxt (concepts cxt) 3)**
  `#"constant" "natural" "stagnant"`



Substructures and Decomposition

Ordinal Motifs

```
(require 'conexp.fca.ordinal-motifs)
```

```
(def os
(conexp.fca.ordinal-motifs/generate-scale :ordinal 3))

(conexp.fca.ordinal-motifs/is-of-scale? :ordinal os)
true
```

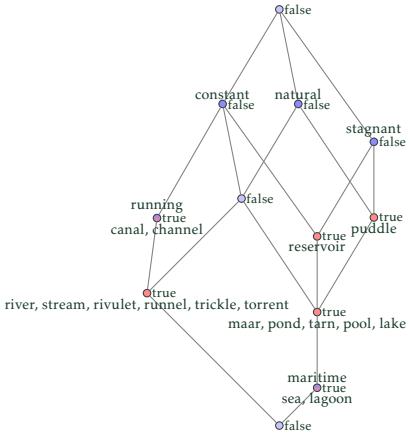
Ordinal Motifs: Example

```
(require 'conexp.fca.ordinal-motifs)
```

```
(def water-complex
  (conexp.fca.ordinal-motifs/make-scale-complex
    (dual-context cxt)))

(conexp.fca.ordinal-motifs/get-complex
  water-complex :contranominal :maximal true )
```

```
#{"running" "maritime"}
#{"running" "natural"}
#{"constant" "natural" "stagnant"}
#{"running" "stagnant"}
```



Scale-Measures [HH22]

(require 'conexp.fca.smeasure)

	running	*maritime*	*constant*	*natural*	*stagnant*
river	×	·	×	×	·
stream	×	·	×	×	·
reservoir	·	·	×	·	×
puddle	·	·	·	×	×
sea	·	×	×	×	×
lagoon	·	×	×	×	×
rivulet	×	·	×	×	·
maar	·	·	×	×	×
pond	·	·	×	×	×
runnel	×	·	×	×	·
tarn	·	·	×	×	×
canal	×	·	×	·	·
pool	·	·	×	×	×
trickle	×	·	×	×	·
channel	×	·	×	·	·
lake	·	·	×	×	×
torrent	×	·	×	×	·

	fun	*favorit*
river	×	×
stream	×	×
reservoir	·	·
puddle	·	×
sea	·	×
lagoon	·	×
rivulet	×	×
maar	·	×
pond	·	×
runnel	×	×
tarn	·	×
canal	×	·
pool	·	×
trickle	×	×
channel	×	·
lake	·	×
torrent	×	×

Scale-Measures [HH22]

(require 'conexp.fca.smeasure)

Scale-Measure for Bodies of water

- We want to measure the objects of *Bodies of water* using the attribute set {fun, favorite}.
- Since both contexts shall use the same objects, we can use an identity map.
- **(def sm1**
  (conexp.fca.smeasure/make-smeasure
  cxt cxt-sm identity)
- *If the result is not a scale-measure, no scale-measure will returned.*

	fun	*favorit*
river	×	×
stream	×	×
reservoir	·	·
puddle	·	×
sea	·	×
lagoon	·	×
rivulet	×	×
maar	·	×
pond	·	×
runnel	×	×
tarn	·	×
canal	×	·
pool	·	×
trickle	×	×
channel	×	·
lake	·	×
torrent	×	×

Scale-Measures II

(println sm1)

```
conexp.analysis=> (conexp.fca.smeasure/make-smeasure cxt cxt-sm identity)
          |running maritime constant natural stagnant                |fun favorit
----------+-------------------------------------------    ----------+------------
river     |x       .       x       x       .         →   river     |x   x
----------+-------------------------------------------    ----------+------------
stream    |x       .       x       x       .         →   stream    |x   x
----------+-------------------------------------------    ----------+------------
reservoir |.       .       x       .       x         →   reservoir |.   .
----------+-------------------------------------------    ----------+------------
puddle    |.       .       .       x       x         →   puddle    |.   x
----------+-------------------------------------------    ----------+------------
sea       |.       x       x       x       x         →   sea       |.   x
----------+-------------------------------------------    ----------+------------
lagoon    |.       x       x       x       x         →   lagoon    |.   x
----------+-------------------------------------------    ----------+------------
rivulet   |x       .       x       x       .         →   rivulet   |x   x
----------+-------------------------------------------    ----------+------------
maar      |.       .       x       x       x         →   maar      |.   x
----------+-------------------------------------------    ----------+------------
pond      |.       .       x       x       x         →   pond      |.   x
----------+-------------------------------------------    ----------+------------
runnel    |x       .       x       x       .         →   runnel    |x   x
----------+-------------------------------------------    ----------+------------
tarn      |.       .       x       x       x         →   tarn      |.   x
```

Scale-Measures III

A scale-measure can always be represented in a canonical form, i.e., using a identity map on objects and attributes that are created using conjunction/disjunction on the original attribute set.

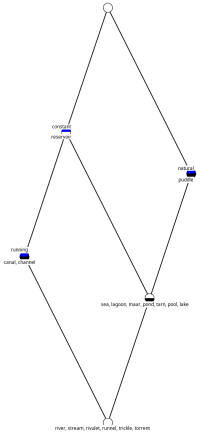
- **(conexp.fca.smeasure/canonical-smeasure-representation sm1)**
- **smeasure-valid-attr**
  Filters the scale-measure scale attributes to the set of valid attributes
- **smeasure-invalid-attr**
- **smeasure-valid-exts**
  Filters the scale-measure extents to the set of valid extents.

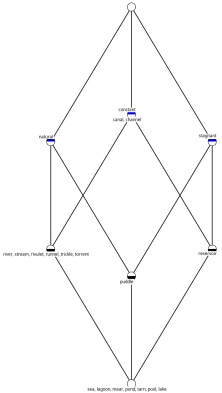
Maximal Decomposable Intervals

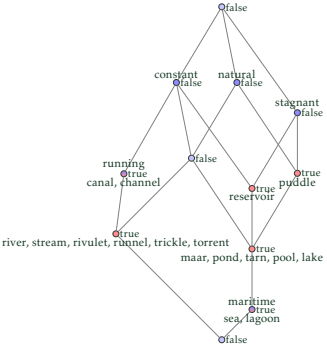
Part of Bachelor Thesis by Jannik Nordmeyer (https://doi.org/10.17170/kobra-2025021510916)

(def il (conexp.fca.decompositions/maximally-decomposable-intervals))

(draw-lattice (frist il)) (draw-lattice (second il))







Wishlist?

If you have one wish

- Which function would you like to see in `conexp-clj`
  (Pleas don't say better GUI)

Bibliography I

[HH19] Tom Hanika and Johannes Hirth. “Conexp-Clj - A Research Tool for FCA.” In: *Supplementary Proceedings of ICFCA 2019 Conference and Workshops, Frankfurt, Germany, June 25-28, 2019*. Ed. by Diana Cristea et al. Vol. 2378. CEUR Workshop Proceedings. CEUR-WS.org, 2019, pp. 70–75. URL: http://ceur-ws.org/Vol-2378/shortAT8.pdf.

[HH22] Tom Hanika and Johannes Hirth. “On the lattice of conceptual measurements.” In: *Information Sciences* 613 (2022), pp. 453–468. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.09.005. URL: https://www.sciencedirect.com/science/article/pii/S0020025522010489.

Bibliography II

[HKS19] Tom Hanika, Maren Koyda, and Gerd Stumme. “Relevant Attributes in Formal Contexts.” In: *Graph-Based Representation and Reasoning - 24th International Conference on Conceptual Structures, ICCS 2019, Marburg, Germany, July 1-4, 2019, Proceedings*. Ed. by Dominik Endres, Mehwish Alam, and Diana Sotropa. Vol. 11530. Lecture Notes in Computer Science. Springer, 2019, pp. 102–116. DOI: 10.1007/978-3-030-23182-8_8. URL: https://doi.org/10.1007/978-3-030-23182-8%5C_8.