

Logical Information Systems

From Taxonomies to Logics

Sébastien Ferré
Team LIS, Irista, University of Rennes 1

6th September 2007, DEXA-FIND, Regensburg

- original idea of Olivier Ridoux around 1996, first published in 2000
- motivations:
 - combining **querying** and **navigation**
 - like Dynamic Taxonomies
 - using **logics** for object descriptions and queries
- foundations:
 - **Formal Concept Analysis** [Godin et al., 1993]
 - extended to cope with logics [Ferré and Ridoux, 2000]

- 1 Formal Concept Analysis
- 2 Logics vs Taxonomies
- 3 Camelis: short demo
- 4 Conclusion

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

Formal Concept Analysis

A mathematical theory [Wille, 1982]

- a **formal context** is an information base
- object descriptions and queries are sets of attributes
- the **extent** of a query A is the set of objects sharing the attributes A
- the **intent** of a set of objects O is the set of attributes shared by O
- **concepts** are derived from the context as pairs (extent,intent),
- and organized in a **concept lattice**

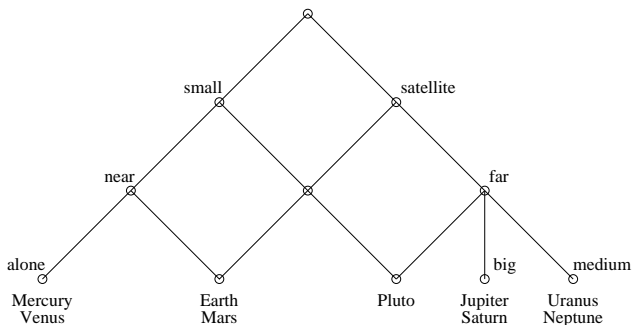
Formal Context

| | small | medium | big | near | far | satellite | alone |
|---------|-------|--------|-----|------|-----|-----------|-------|
| Mercury | • | | | • | | | • |
| Venus | • | | | • | | | • |
| Earth | • | | | • | | • | |
| Mars | • | | | • | | • | |
| Jupiter | | | • | | • | • | |
| Saturn | | | • | | • | • | |
| Uranus | | • | | | • | • | |
| Neptune | | • | | | • | • | |
| Pluto | • | | | | • | • | |

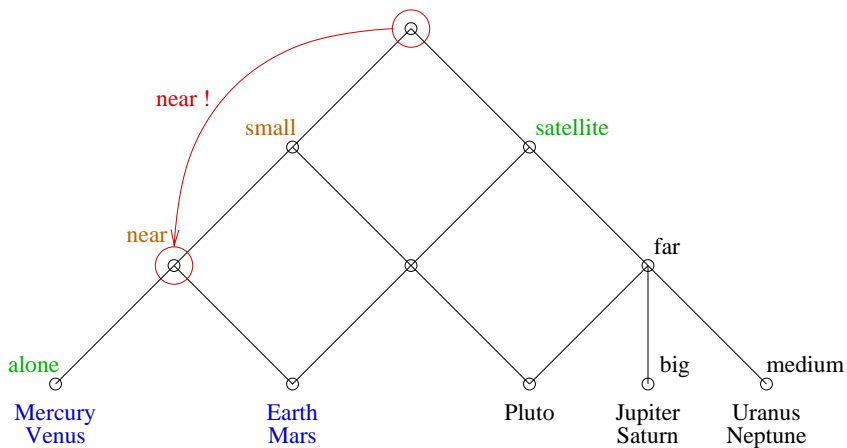
Concept Lattice

The set of **all concepts** derived from a context is ordered

- from the most general at the top (all objects) to the most specific
- this forms a **lattice** (a rich algebraic structure)



Navigation in the Concept Lattice



Formal Concept Analysis vs Dynamic Taxonomies

The two approaches result in a very similar interaction:

- only the attributes that are **relevant** to the current set of objects are presented to users
- each relevant attribute can be used a **navigation link** to refine the search

but...

- dynamic taxonomies make **large set of attributes** more manageable than a flat list
- the concept lattice is a very fruitful structure that defines the **whole navigation space**, and supports data-mining and machine learning tasks

Hence **Logical Concept Analysis**...

Formal Concept Analysis vs Dynamic Taxonomies

The two approaches result in a very similar interaction:

- only the attributes that are **relevant** to the current set of objects are presented to users
- each relevant attribute can be used a **navigation link** to refine the search

but...

- dynamic taxonomies make **large set of attributes** more manageable than a flat list
- the concept lattice is a very fruitful structure that defines the **whole navigation space**, and supports data-mining and machine learning tasks

Hence **Logical Concept Analysis**...

Logic vs Taxonomy

Both logic and taxonomy can be defined as a **partially ordered set** (X, \leq) . They differ in the way they are defined:

- a **taxonomy** is defined as the transitive closure of a directed acyclic graph (DAG)
- a **logic** defines X as a language (of formulas), and \leq as an entailment relation
 - the set X is often **infinite**
 - we suppose there is an **automatic procedure** to decide whether $x \leq y$, for every formulas x, y (decidable logic)

Examples of Logics

Location: taxonomy of locations, as a logic

- $X = \{\text{World, Europe, Spain, France, ...}\}$
- $x \leq y$ iff $x = y$ or it exists a $z \in X$ s.t $z \leq y$ and $(x, z) \in \{(\text{Europe,World}), (\text{Spain,Europe}), \dots\}$

Integer: logic of integers and intervals

- $X = \{[a, b] \mid a \leq b \in \mathbb{Z}\}$
- $[a_1, b_1] \leq [a_2, b_2]$ iff $a_2 \leq a_1$ and $b_1 \leq b_2$

String: logic of strings and substrings

- $X = \Sigma^*$, where Σ is an alphabet
- $x \leq y$ iff y is a substring of x

Examples of Logics

Location: taxonomy of locations, as a logic

- $X = \{\text{World, Europe, Spain, France, ...}\}$
- $x \leq y$ iff $x = y$ or it exists a $z \in X$ s.t $z \leq y$ and $(x, z) \in \{(\text{Europe, World}), (\text{Spain, Europe}), \dots\}$

Integer: logic of integers and intervals

- $X = \{[a, b] \mid a \leq b \in \mathbb{Z}\}$
- $[a_1, b_1] \leq [a_2, b_2]$ iff $a_2 \leq a_1$ and $b_1 \leq b_2$

String: logic of strings and substrings

- $X = \Sigma^*$, where Σ is an alphabet
- $x \leq y$ iff y is a substring of x

Examples of Logics

Location: taxonomy of locations, as a logic

- $X = \{\text{World, Europe, Spain, France, ...}\}$
- $x \leq y$ iff $x = y$ or it exists a $z \in X$ s.t $z \leq y$ and $(x, z) \in \{(\text{Europe, World}), (\text{Spain, Europe}), \dots\}$

Integer: logic of integers and intervals

- $X = \{[a, b] \mid a \leq b \in \mathbb{Z}\}$
- $[a_1, b_1] \leq [a_2, b_2]$ iff $a_2 \leq a_1$ and $b_1 \leq b_2$

String: logic of strings and substrings

- $X = \Sigma^*$, where Σ is an alphabet
- $x \leq y$ iff y is a substring of x

To summarize:

- a taxonomy is a logic whose entailment relation is (**extensionnaly**) defined as a set of edges,
- a logic is a taxonomy whose infinite set of edges is (**intentionnaly**) defined in a generic and computable way, so that
 - a taxonomy can be automatically produced given a finite subset of formulas,
 - in which new formulas can be automatically inserted,
 - hence every formula can be used in queries, even if not yet present in the finite taxonomy.

Logic vs Taxonomy (continued)

To summarize:

- a taxonomy is a logic whose entailment relation is (**extensionnaly**) defined as a set of edges,
- a logic is a taxonomy whose infinite set of edges is (**intentionnaly**) defined in a generic and computable way, so that
 - a taxonomy can be automatically produced given a finite subset of formulas,
 - in which new formulas can be automatically inserted,
 - hence every formula can be used in queries, even if not yet present in the finite taxonomy.

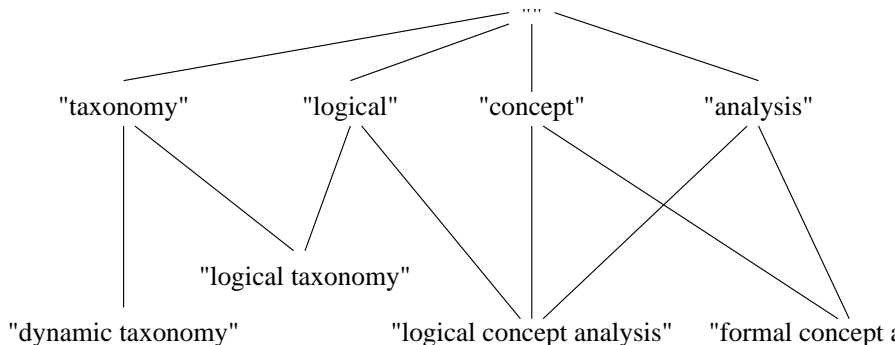
Logic vs Taxonomy (continued)

To summarize:

- a taxonomy is a logic whose entailment relation is (*extensionnaly*) defined as a set of edges,
- a logic is a taxonomy whose infinite set of edges is (*intentionnaly*) defined in a generic and computable way, so that
 - a taxonomy can be automatically produced given a finite subset of formulas,
 - in which new formulas can be automatically inserted,
 - hence every formula can be used in queries, even if not yet present in the finite taxonomy.

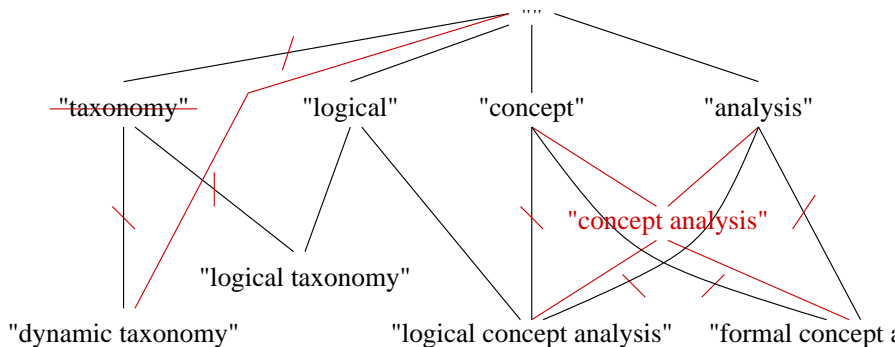
Taxonomy of Strings and Substrings

Taxonomy derived from a set of keywords.



Taxonomy of Strings and Substrings

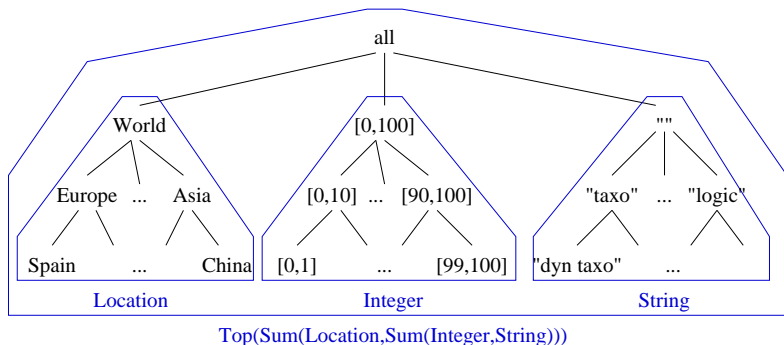
Taxonomy modified by the insertion and deletion of keywords.



Composing Logics

How to describe photos by a location, a size, and a comment ?

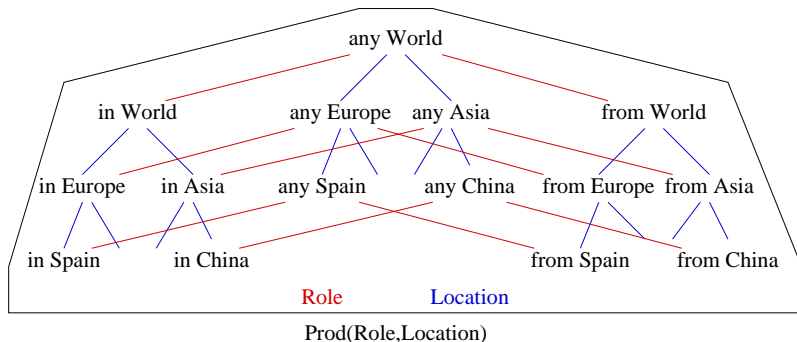
- $L = \text{Top}(\text{Sum}(\text{Location}, \text{Sum}(\text{Integer}, \text{String})))$



Composing Logics

How to distinguish between location (in) and origine (from) ?

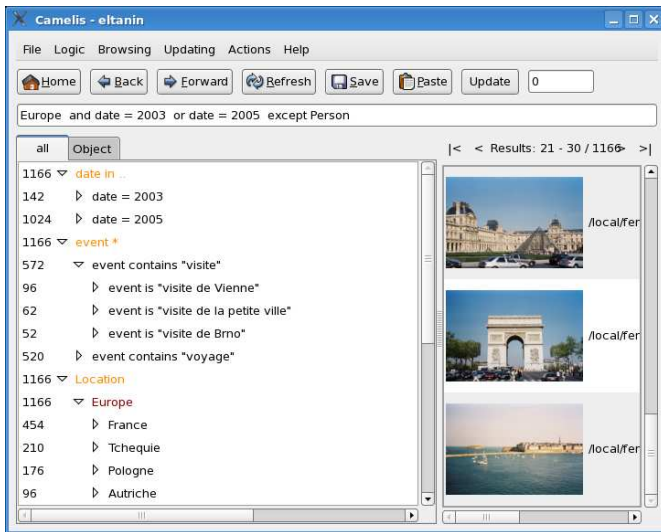
- $L = \text{Prod}(\text{Role}, \text{Location})$, where Role is a taxonomy of role names (in, from, etc.)



- The insertion of `in Italy` entails the immediate insertion of `any Italy` and `from Italy` at the right place.

Camelis: a LIS implementation with a GUI

Applied on my collection of more than 5000 photos, described by date, location, type, event, persons, and objects.



- navigation to build a query with and, or, not, and explain color on features
- add in query a new pattern and show automatic insertion
- manually insert a feature
- show updates in Location taxonomy
- upward navigation

Conclusion

Logics can be used in place of taxonomies for combining querying and navigation, and offer some advantages:

- objects can be described by **precise values** (e.g., strings)
- queries can use **expressive patterns** (e.g., substrings, regular expressions)
- values and patterns **need not be present** in the taxonomy before being used
 - but are **automatically inserted** after first use
- simple formulas can be combined in more **complex formulas**
- logics and taxonomies **can be reused** in different combinations (Prod, Sum, Top)

Useful links:

- the LIS team web site
 - <http://www.irisa.fr/LIS/>
- the Camelis web site
 - <http://www.irisa.fr/LIS/ferre/camelis/>
 - applies on photos, music files, BibTeX files, DBLP pages

