Wake the Concept Lattice in You!

Marianne Huchard
Montpellier University, LIRMM, France

Keynote Talk
19th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems
July 11-15, 2022 – Milan, Italy

Agenda

RESTRUCTURING LATTICE THEORY: AN APPROACH BASED ON HIERARCHIES OF CONCEPTS

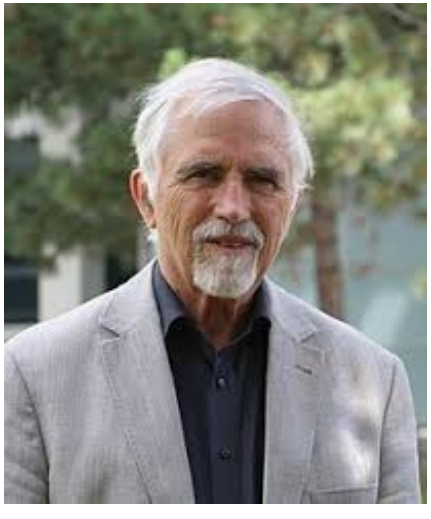
Rudolf Wille
Fachbereich Mathematik
Technische Hochschule Darmstadt
6100 Darmstadt
Federal Republic of Germany

ABSTRACT

Lattice theory today reflects the general status of current mathematics: there is a rich production of theoretical concepts, results, and developments, many of which are reached by elaborate mental gymnastics; on the other hand, the connections of the theory to its surroundings are getting weaker and weaker, with the result that the theory and even many of its parts become more isolated. Restructuring lattice theory is an attempt to reinvigorate connections with our general culture by interpreting the theory as concretely as possible, and in this way to promote better communication between lattice theorists and potential users of lattice theory.

The approach reported here goes back to the origin of the lattice concept in nineteenth-century attempts to formalize logic, where a fundamental step was the reduction of a concept to its "extent". We propose to make the reduction less abstract by retaining in some measure the "intent" of a concept. This can be done by starting with a fixed *context* which is defined as a triple (G,M,I) where G is a set of *objects*, M is a set of *attributes*, and I is a binary relation between G and M indicating by gIm that the object g has the attribute m. There is a natural Galois connection between G and M defined by $A' = \{m \in M \mid gIm \text{ for all } g \in A\}$ for $A \subseteq G$ and $B' = \{g \in G \mid gIm \text{ for all } m \in B\}$ for $B \subseteq M$. Now, a *concept* of the context (G,M,I) is introduced as a pair (A,B) with $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$, where A is called the *extent* and B the *intent* of the concept (A,B). The *hierarchy of concepts* given by the relation subconcept-superconcept is captured by the definition $(A_1,B_1) \leq (A_2,B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$ for concepts (A_1,B_1) and (A_2,B_2) of (G,M,I). Let $\underline{\mathfrak{B}}(G,M,I)$ be the

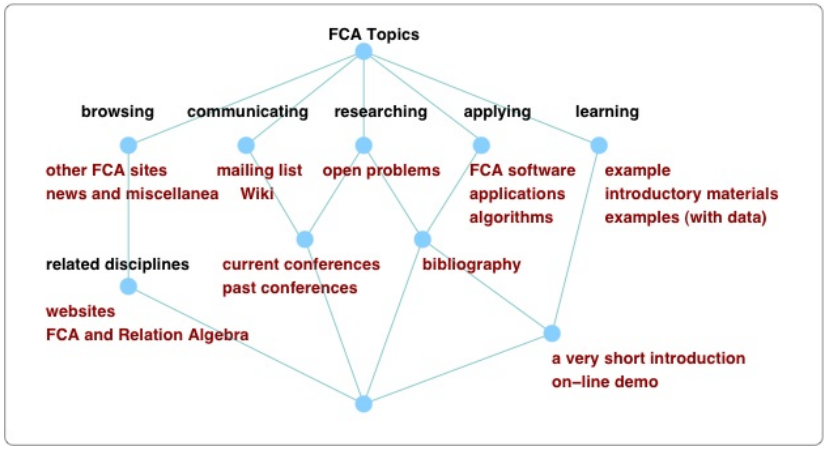
I. Rival (ed.), Ordered Sets, 445–470.



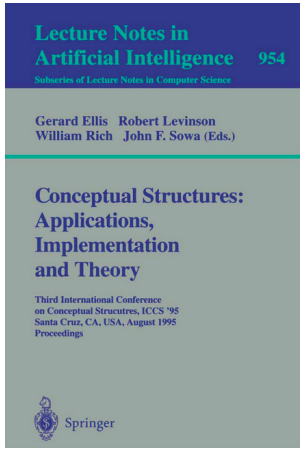
Dissemination 1982-2022

- Conceptualization, philosophical developments
- Data analysis, data mining, clustering
- Knowledge representation (ontology construction)
- Classification, indexation (information retrieval)
- Unsupervised learning
- Supervised learning (adding classes in description)
- Tools and applications

Dissemination 1982-2022



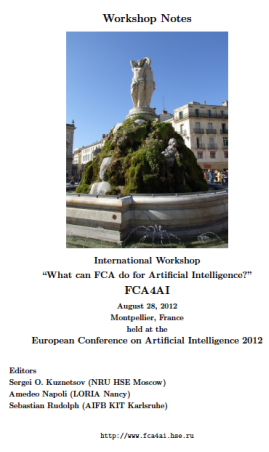
Credits Uta Priss https://upriss.github.io/fca/fca.html



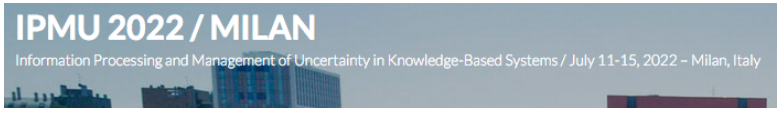
ICCS

ICFCA

CLA



FCA4AI

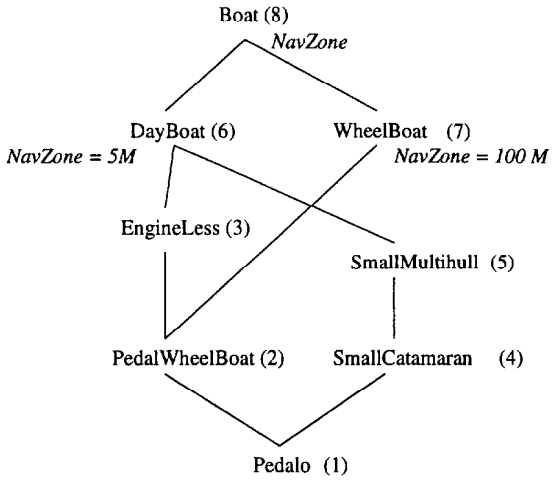


S21. FORMAL CONCEPT ANALYSIS AND UNCERTAINTY

Organizers: *Pablo Cordero*, *Domingo López-Rodríguez*, and *Ángel Mora*

Crossing FCA path

Partial orders

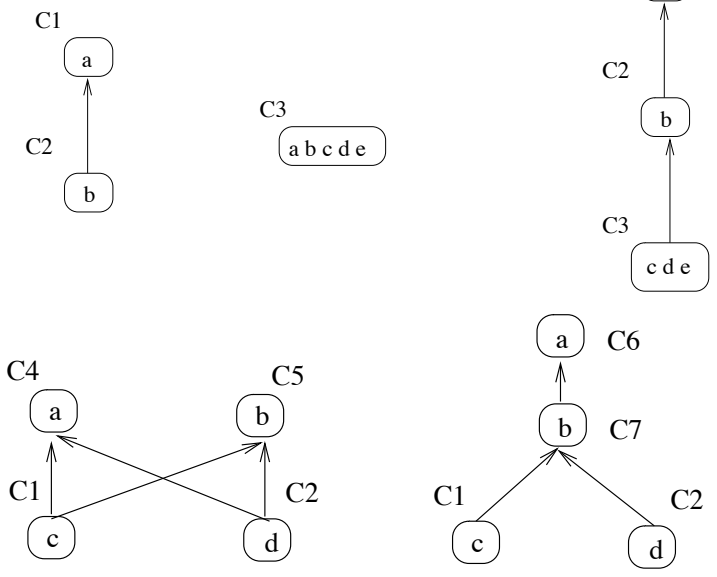


$CLOS(2) = LOOPS(2) = \mathcal{L}^*_{LOOPS}(2) = (2\ 3\ 6\ 7\ 8)$
$CLOS(4) = LOOPS(4) = \mathcal{L}^*_{LOOPS}(4) = (4\ 5\ 6\ 8)$
$CLOS(1) = LOOPS(1) = (1\ 2\ 3\ 7\ 4\ 5\ 6\ 8)$
$\mathcal{L}^*_{LOOPS}(1) = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8)$

Multiple inheritance conflict resolution

OOPLSA 1994

Concept lattices

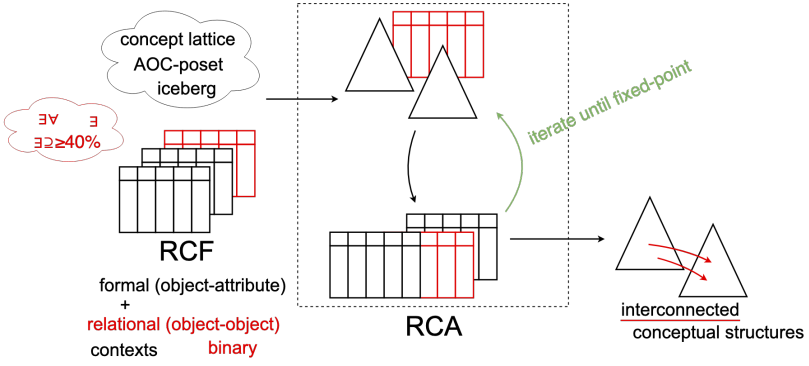


Class hierarchy compact factorization

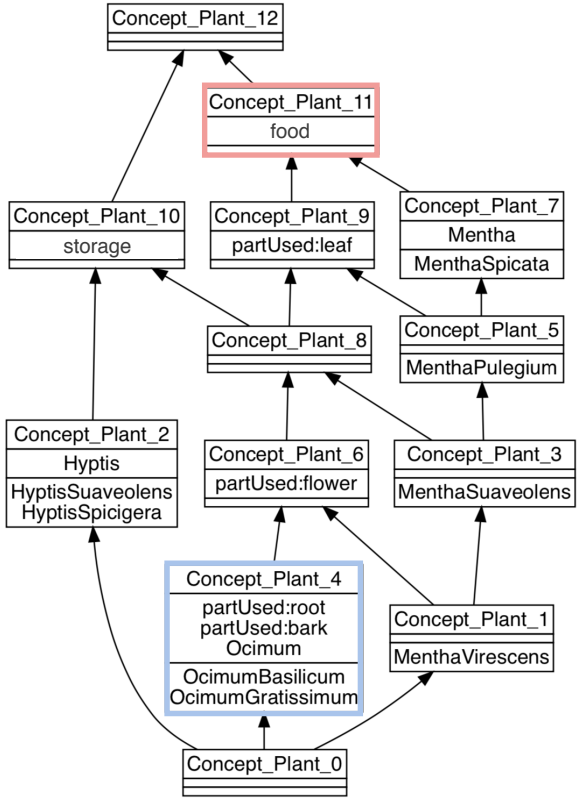
OOPSLA 1996

An 'engineering view' on FCA

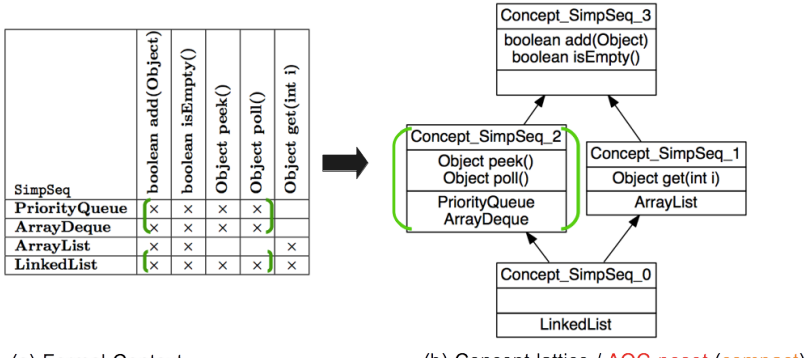
Algorithms, relational extension (RCA)



Environmental data applications



Software engineering applications



(a) Formal Context
classes X signatures of public methods

(b) Concept lattice / AOC-poset (compact)
maximal factorization, exact specialization

Agenda

Formal Concept Analysis

Formal concepts are "a natural feature of information representation which is as fundamental to hierarchies and object/attribute structures as set theory or relational algebra are for relational databases".

Uta Priss. 40th anniv. vol. of Annual Review of Inf. Sc. and Tech., 2006

Simple but powerful basics

- Formal Context
- Concept Lattice
- Galois connection

Formal Context

Triple (O, A, R), where O is a finite set of objects, A is a finite set of attributes and $R \subseteq O \times A$ is a binary relation.
$(o, a) \in R$ means that object o owns attribute a.

	flies	*nocturnal*	*feathered*	*migratory*	*red-bill*	*elytra*	*sea-habitat*	*wood-habitat*	*six-legged*	*eats-fish*	*water-habitat*
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×

Characteristic maps of $R \subseteq O \times A$

Map f associates an object set with the shared attributes.

$f : P(O) \to P(A) \quad X \longmapsto f(X) = \{y \in A \mid \forall x \in X, (x, y) \in R\}$

	flies	*nocturnal*	*feathered*	*migratory*	*red-bill*	*elytra*	*sea-habitat*	*wood-habitat*	*six-legged*	*eats-fish*	*water-habitat*
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×

In [Ganter and Wille, 1999], f is denoted by the polymorphic symbol $'$. We will use both notations depending the situations

Characteristic maps of $R \subseteq O \times A$

Map g associates an attribute set with the objects sharing them

$g : P(A) \to P(O) \quad Y \longmapsto g(Y) = \{x \in O \mid \forall y \in Y, (x, y) \in R\}$

	flies	*nocturnal*	*feathered*	*migratory*	*red-bill*	*elytra*	*sea-habitat*	*wood-habitat*	*six-legged*	*eats-fish*	*water-habitat*
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×

In [Ganter and Wille, 1999], g is denoted by the polymorphic symbol $'$. We will use both notations depending the situations

Formal Concept

Pair C=(E, I) such that $f(E) = I$ (or equivalently $E = g(I)$)
E is the concept *extent*; I is the concept *intent*.

	flies	*nocturnal*	*feathered*	*migratory*	*red-bill*	*elytra*	*sea-habitat*	*wood-habitat*	*six-legged*	*eats-fish*	*water-habitat*
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×

Concept ordering

Concepts can be ordered through the following partial order $\leq_s$:
$(E_1, I_1) \leq_s (E_2, I_2) \Leftrightarrow E_1 \subseteq E_2$, or equivalently $I_2 \subseteq I_1$
(E_1, I_1) sub-concept of (E_2, I_2); (E_2, I_2) super-concept of (E_1, I_1)

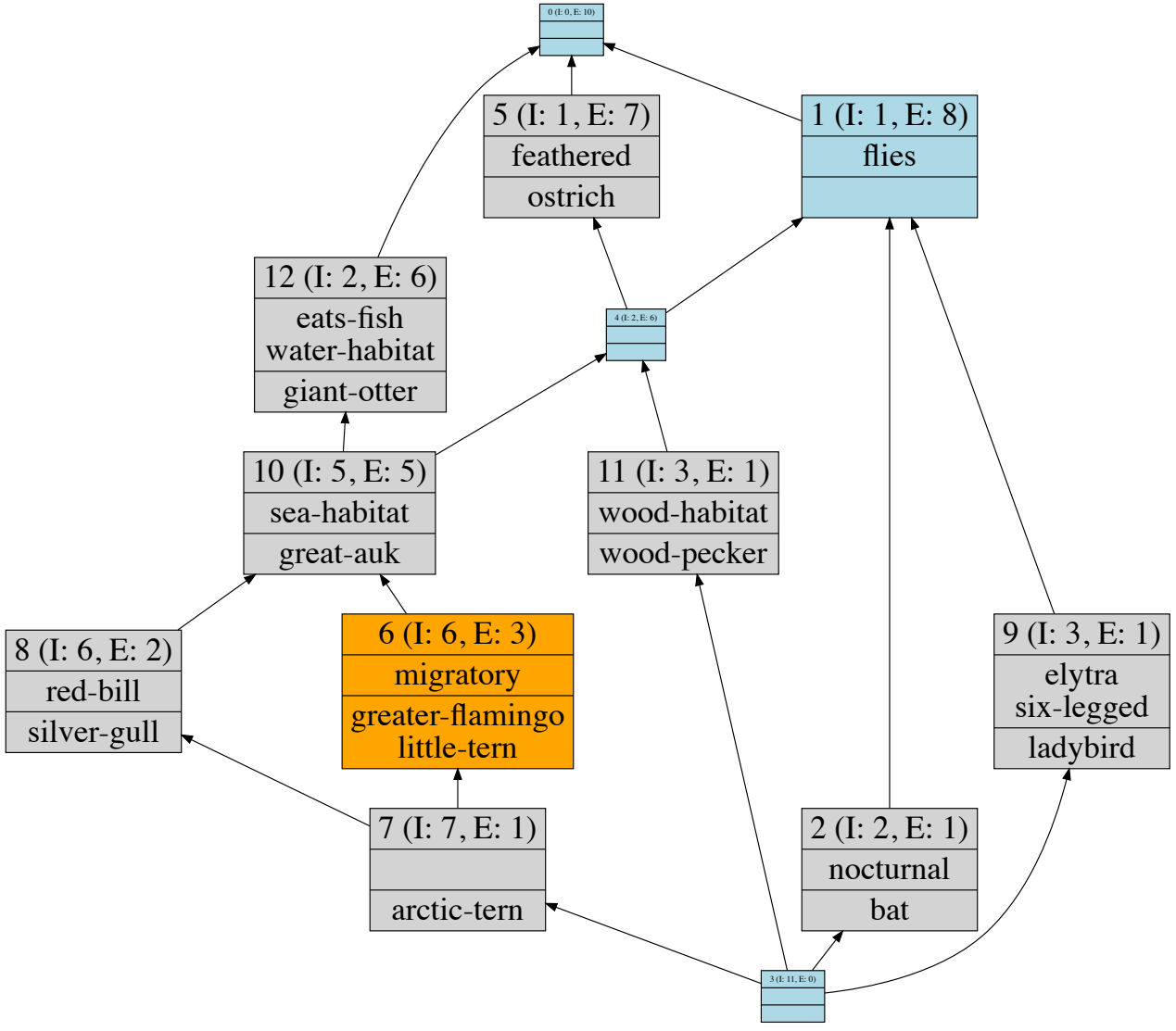
$C_{great\text{–}auk}$:
$X_{10} =$
$\{great\text{–}auk, silver\text{–}gull, greater\text{–}flamingo, little\text{–}tern, arctic\text{–}tern\}$
$Y_{10} = \{sea\text{–}habitat, eats\text{–}fish, water\text{–}habitat, feathered, flies\}$

$C_{silver\text{–}gull}$:
$X_8 = \{silver\text{–}gull, arctic\text{–}tern\}$
$Y_8 = \{sea\text{–}habitat, eats\text{–}fish, water\text{–}habitat, feathered, flies, red\text{–}bill\}$

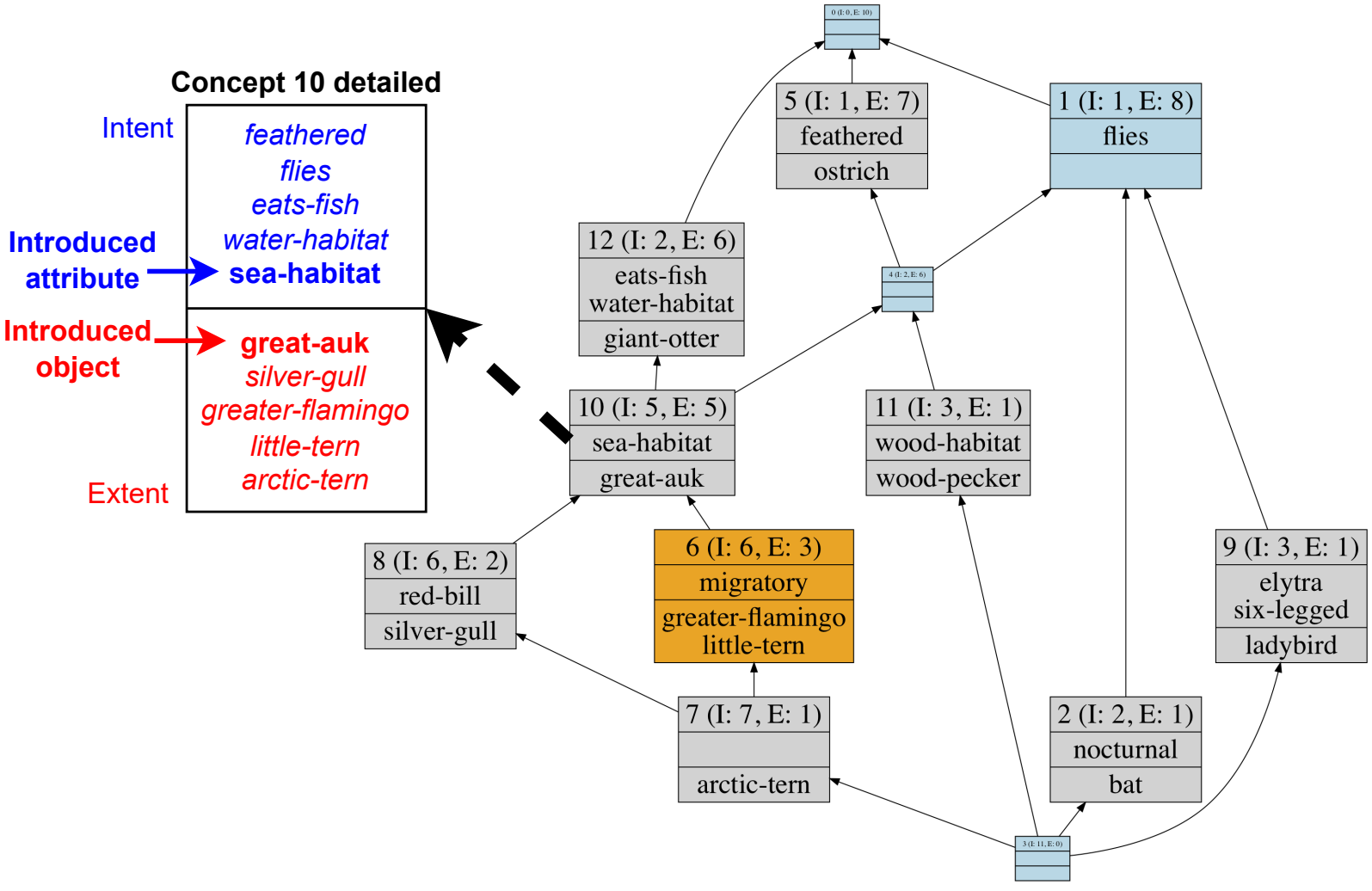
$C_{silver\text{–}gull} \leq_s C_{great\text{–}auk}$, as $X_8 \subseteq X_{10}$ (and $Y_{10} \subseteq Y_8$).

Concept lattice $\mathcal{L}_K = (\mathcal{C}_K, \leq_s)$

Set $\mathcal{C}_K$ of all concepts of $K = (O, A, R)$, provided with $\leq_s$

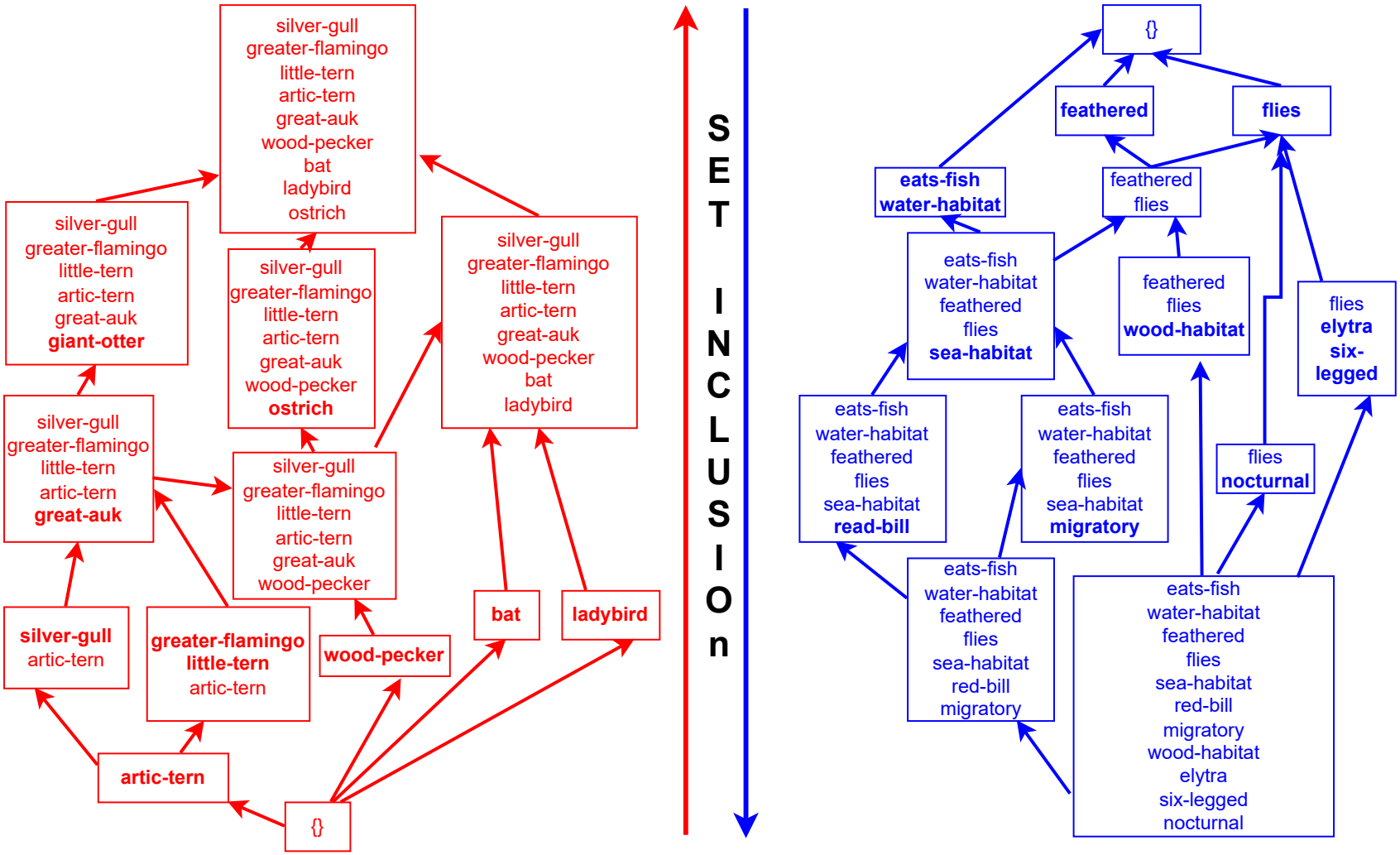


Concept lattice $\mathcal{L}_K = (\mathcal{C}_K, \leq_s)$

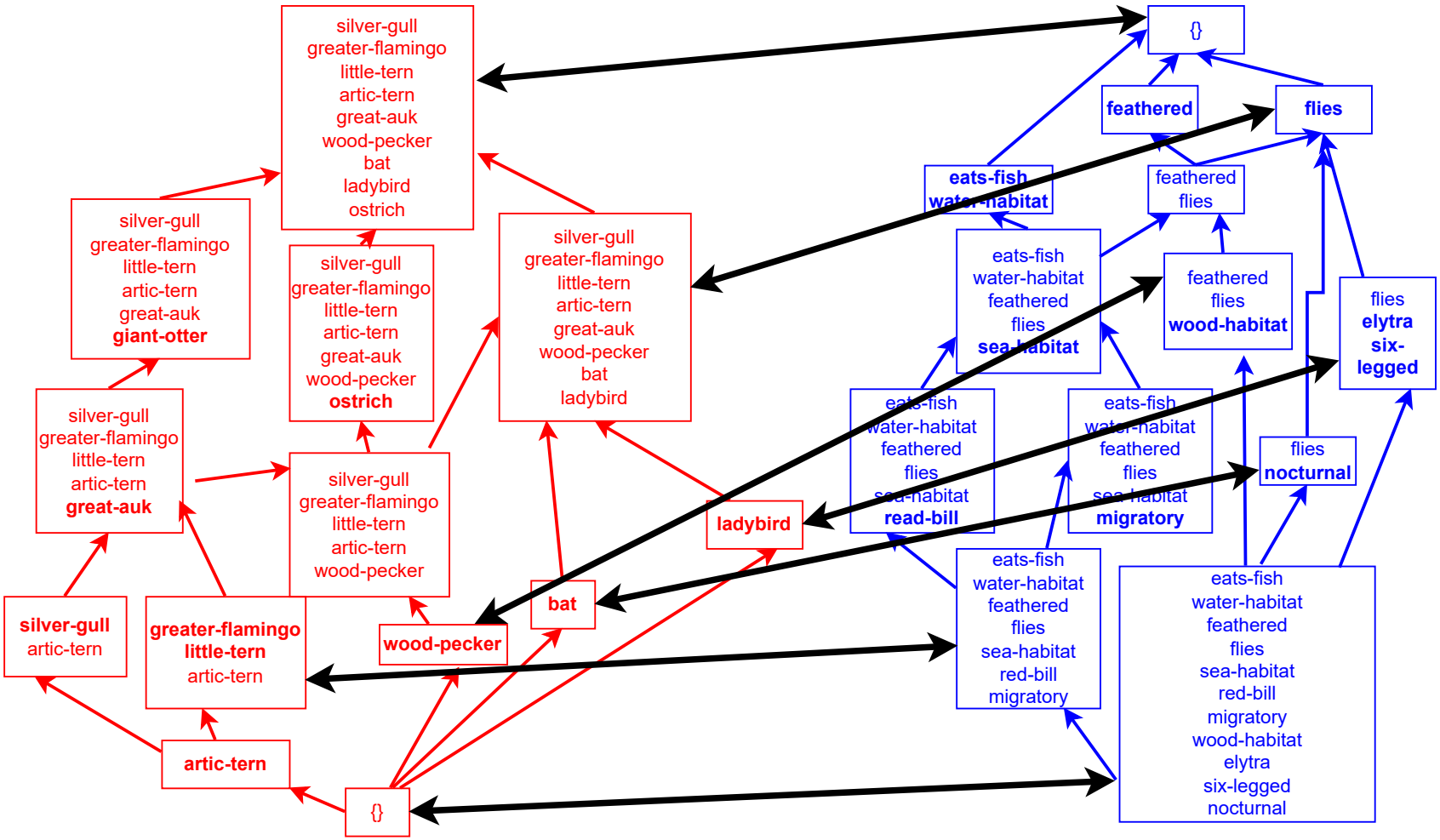


Top-down inherited attributes, bottom-up inherited objects

Concept lattices: Assembly of two isomorphic lattices



Assembly: through a Galois connection



Generalization: Galois connection

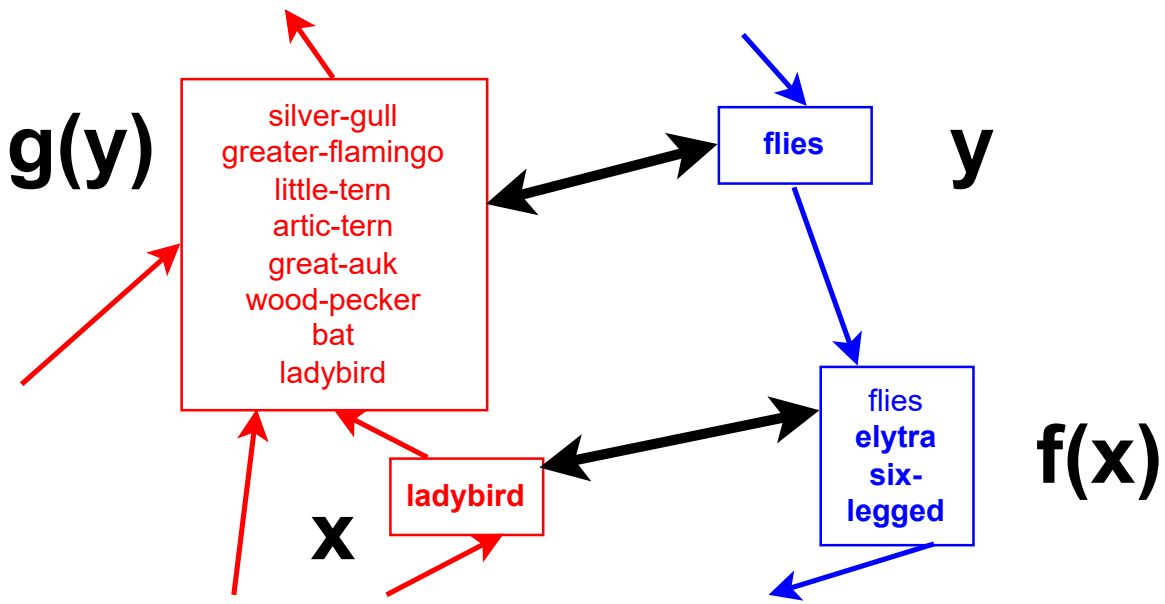
(f, g) associated with a $R \subseteq O \times A$ is a Galois connection between $(2^O, \subseteq)$ and $(2^A, \subseteq)$

Galois connection

For posets $(X, \leq_X)$ and $(Y, \leq_Y)$, and $f : X \to Y$ et $g : Y \to X$

The pair (f, g) is a Galois connection if:

$\forall x \in X$ and $\forall y \in Y$, $x \leq_X g(y) \Leftrightarrow y \leq_Y f(x)$.



poset: partially ordered set

Generalization: Galois connection

If (f, g) is a Galois connection between $(X, \leq_X)$ and $(Y, \leq_Y)$
$h_X = g \circ f$ et $h_Y = f \circ g$ are closure operators:

op closure operator on a poset $(P, \leq_P)$

- isotone: $X \leq_P Y \Rightarrow op(X) \leq_P op(Y)$
- extensive: $X \leq_P op(X)$
- idempotent: $op(op(X)) = op(X)$

X is a closed set iff $op(X) = X$

$X = \{great\text{–}auk, silver\text{–}gull\}$
$f(X) = \{sea\text{–}habitat, eats\text{–}fish, water\text{–}habitat, feathered, flies\}$
$g(f(X)) = \{great\text{–}auk, silver\text{–}gull, greater\text{–}flamingo, little\text{–}tern, arctic\text{–}tern\}$
$f(g(f(X))) = \{sea\text{–}habitat, eats\text{–}fish, water\text{–}habitat, feathered, flies\}$
$g(f(g(f(X)))) = \{great\text{–}auk, silver\text{–}gull, greater\text{–}flamingo, little\text{–}tern, arctic\text{–}tern\}$

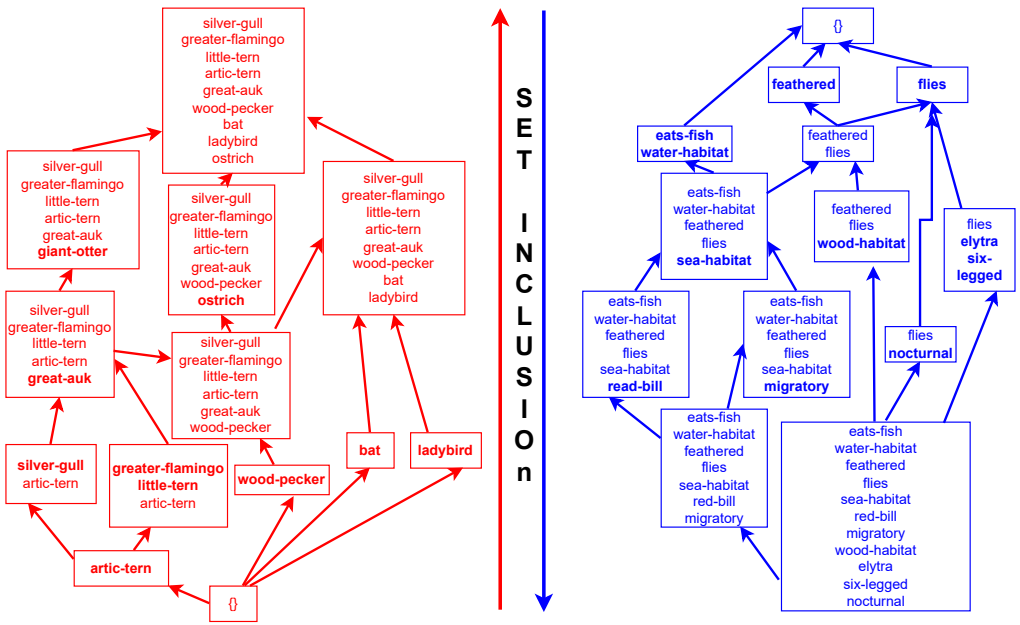
Generalization: Galois connection

Property (Closure lattice)

Let F the set of closed sets of a closure operator h on a finite lattice T. Let us consider the following operations on $x, y \in F$:

- $x \wedge_F y = x \wedge y$
- $x \vee_F y = h(x \vee y)$

$(F, \wedge_F, \vee_F)$ is a lattice.



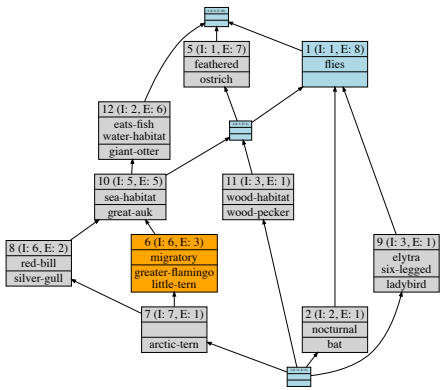
E.g. the lattices of the closed sets of $f \circ g$ and $g \circ f$ for (f, g) associated with $R \subseteq O \times A$:

$x \wedge y$ is $x \cap y$

$h(x \vee y)$ is $g \circ f(x \cup y)$ or $f \circ g(x \cup y)$

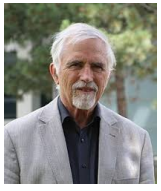
Generalization: Galois lattice

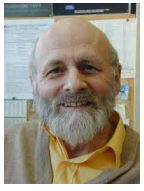
If (f, g) is a Galois connection between $(A, \leq_A)$ and $(B, \leq_B)$, and $h_A = g \circ f$ and $h_B = f \circ g$ are the associated closure operators:
If F_{h_A} are the closed sets of h_A and F_{h_B} the closed sets of h_B.
If A (resp. B) is a finite lattice, F_{h_A} (resp. F_{h_B}) can be provided with a lattice structure.
They are isomorphic (by f or g).
The Galois lattice is the set of pairs (x, y) such that $y = f(x)$ (or eq. $x = g(y)$), provided with the order $(x_1, y_1) \leq_G (x_2, y_2)$ iff $x_1 \leq_A x_2$



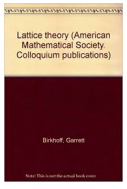
E.g. the Galois lattice of the Galois connection (f, g) associated with $R \subseteq O \times A$.

Inspiring references







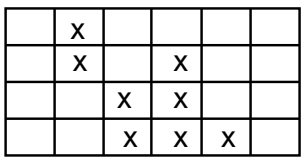

- Concept lattices
  (Rudolf Wille, 1982, Bernhard Ganter & Rudolf Wille, 1999)
- Lattice theory, Galois connections, Galois lattices
  (Marc Barbut & Bernard Monjardet, 1970; Georges David Birkhoff, 1940; Øystein Ore, 1944)
- Galois connection, subgroups/subfields
  (Evariste Galois, $\sim$ 1830)

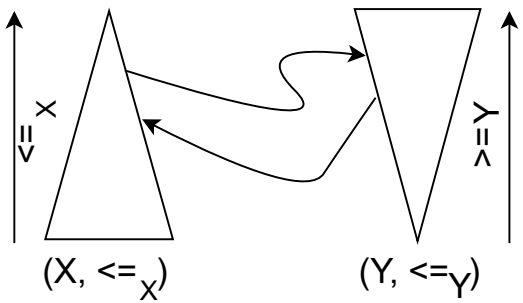
Sum up

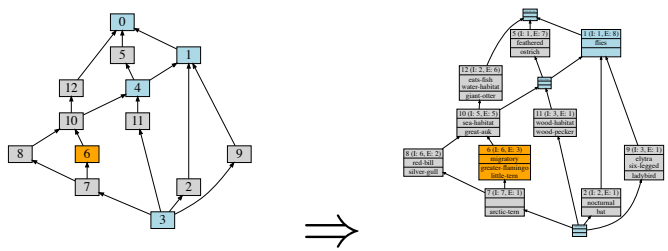
Any Galois connection between finite lattices $(X, \leq_X)$ and $(Y, \leq_Y)$ induces a Galois lattice (i.e. theory is not restricted to tabular data)

Duality between: Objects/attributes ; lattices $(X, \leq_X)$ and $(Y, \leq_Y)$

Genericity: Any lattice can be labelled to be a Galois lattice of some binary relation

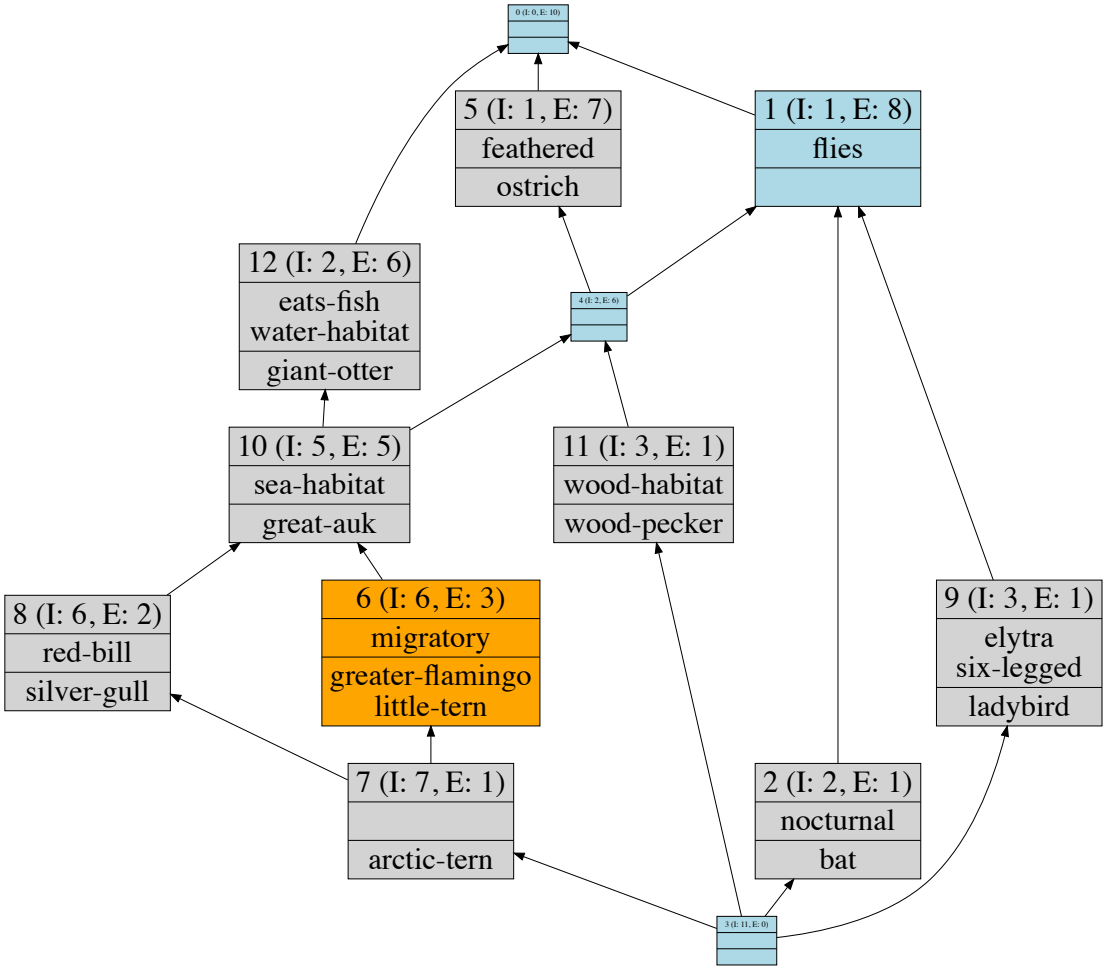






Agenda

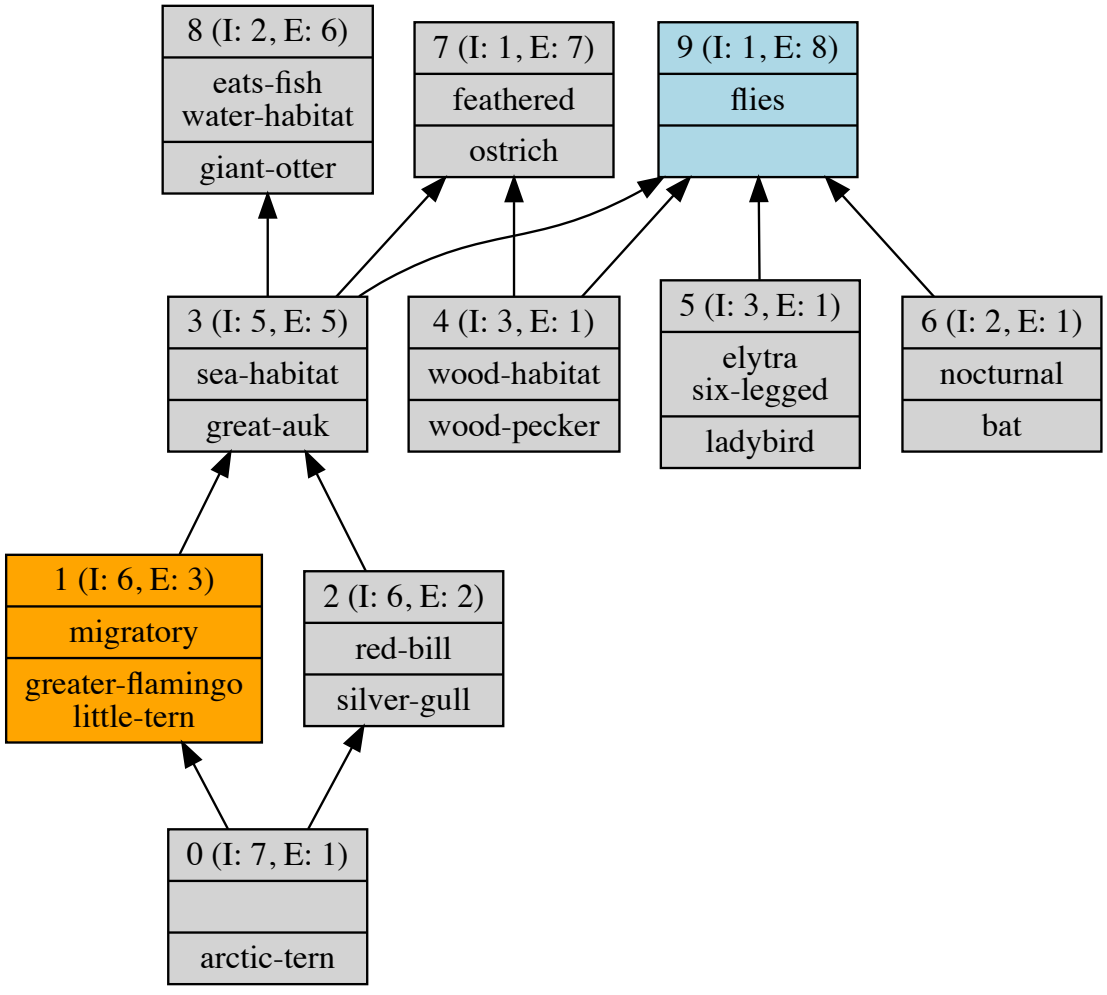
Conceptual structures: Concept lattice



Worst case:
#concepts $< 2^{min(|A|,|O|)}$

Reached with the lattice of all subsets of E where $E = O$ if $|O| = min(|A|, |O|)$ (otherwise, $E = A$)

Conceptual structures: AOC poset

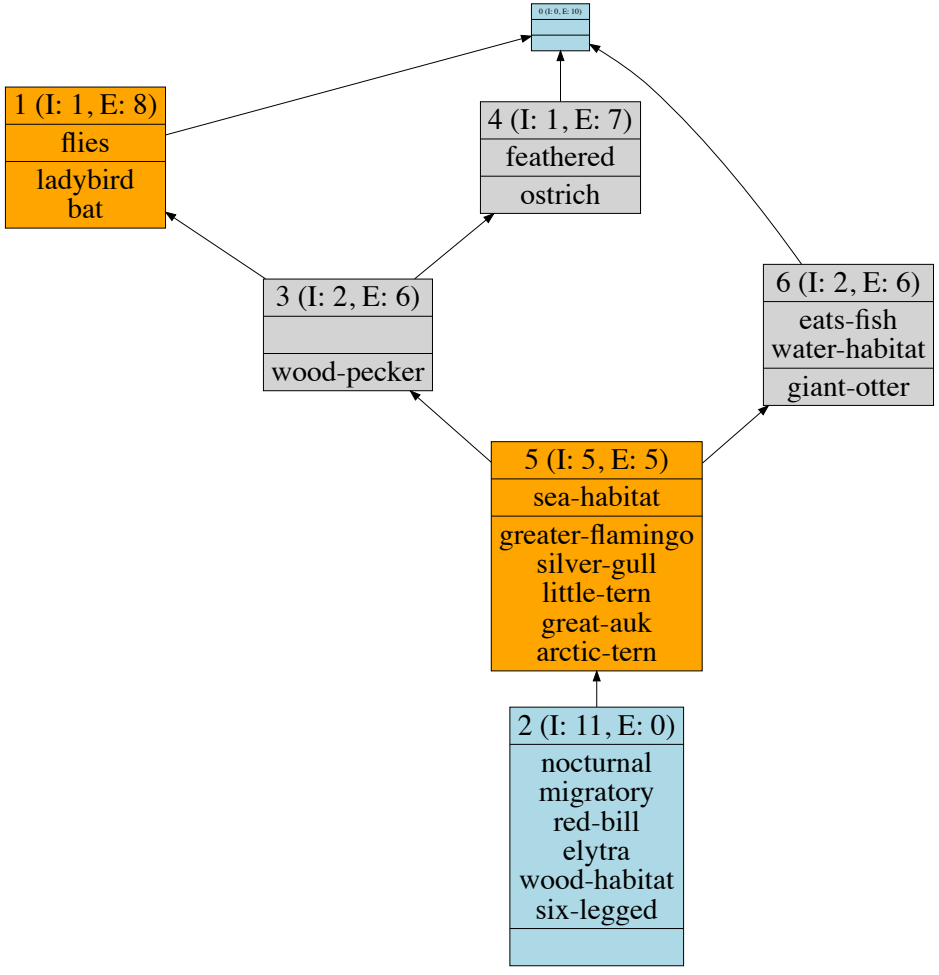


Sub-order of introducer concepts

Worst case complexity: #concepts $< |A| + |O|$

Reached when $|A| = |O|$ and every attribute is shared by several distinct objects (ex. bipartite crown graph)

Conceptual structures: Iceberg



Concepts with frequent intent or frequent extent

Here: concepts with frequent extent ($\geq 50\%$)

Implication rules $Prem \implies Conc$ of $K = (O, A, R)$

Pair $(Prem, Conc) \subseteq A \times A$ s.t. $g(Prem) \subseteq g(Conc)$

all the objects that own the attributes of Prem (premise) also own the attributes of Conc (conclusion)

$$elytra \implies six\text{-}legged$$

$$feathered \wedge eats - fish \wedge water - habitat \implies flies \wedge sea - habitat$$

Implication rules

Criteria for an implication set

- Sound: any implication of the set holds
- Complete: any other implication can be obtained from the set
- Basis: sound and complete
- Minimal cardinality
- Direct: any attribute closed set can be obtained in one iteration
- No redundancy between rules and inside rules

Implication rules

Diverse approaches

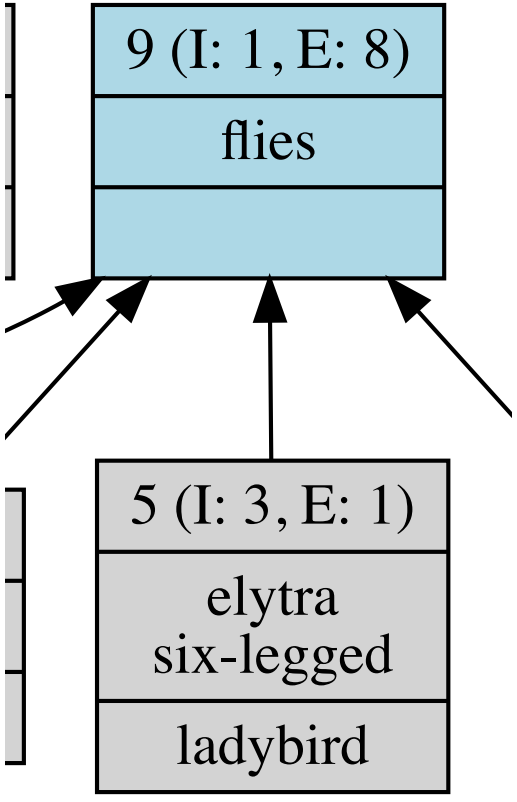
- Minimal non-redundant set of binary implications
- Basis of Duquennes-Guigues [Guigues, 1986] (For a systematic study, see [Bertet&Monjardet, 2010])
- Left-minimal direct basis of implications [Cordero 2013]
- Basis of proper premises [Reppe, 2008; Ryssel, 2014]
- and others ...

Duquenne-Guigues Basis of Implications (DGBI)

Cardinality minimal set of non redundant implications

<0> flies,feathered,sea-habitat,wood-habitat,eats-fish,water-habitat => nocturnal,migratory,red-bill,elytra,six-legged
<0> flies,feathered,elytra,six-legged => nocturnal,migratory,red-bill,sea-habitat,wood-habitat,eats-fish,water-habitat
<0> flies,nocturnal,elytra,six-legged => feathered,migratory,red-bill,sea-habitat,wood-habitat,eats-fish,water-habitat
<0> flies,nocturnal,feathered => migratory,red-bill,elytra,sea-habitat,wood-habitat,six-legged,eats-fish,water-habitat
<1> six-legged => flies,elytra
<1> wood-habitat => flies,feathered
<1> elytra => flies,six-legged
<1> nocturnal => flies
<2> red-bill => flies,feathered,sea-habitat,eats-fish,water-habitat
<3> migratory => flies,feathered,sea-habitat,eats-fish,water-habitat
<5> sea-habitat => flies,feathered,eats-fish,water-habitat
<5> feathered,eats-fish,water-habitat => flies,sea-habitat
<5> flies,eats-fish,water-habitat => feathered,sea-habitat
<6> water-habitat => eats-fish
<6> eats-fish => water-habitat

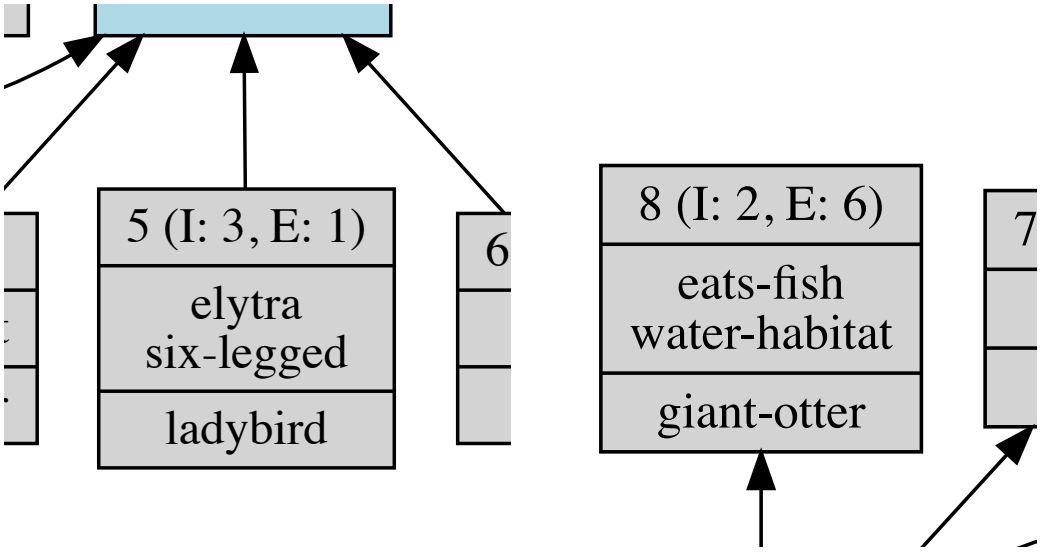
Logical Constraints: Binary implication



Concept 5 subconcept of Concept 9

elytra $\Longrightarrow$ flies

Logical Constraints: Co-occurrence (equivalence)



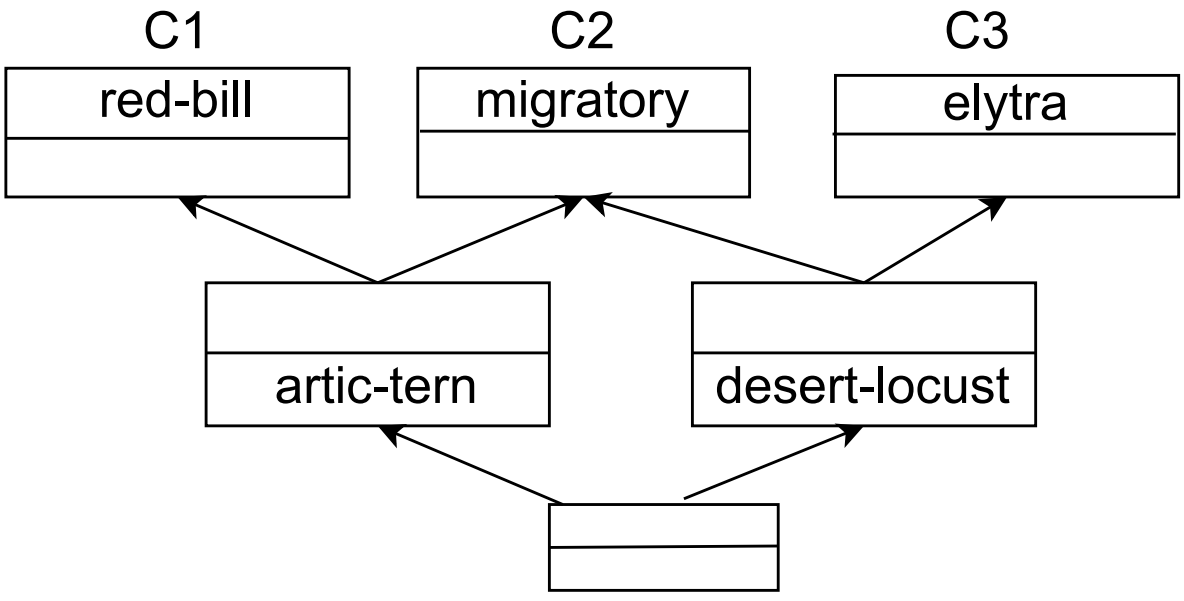
elytra and six-legged both introduced in Concept 5

$$\text{elytra} \Leftrightarrow \text{six-legged}$$

eats-fish and water-habitat both introduced in Concept 8

$$\text{eats-fish} \Leftrightarrow \text{water-habitat}$$

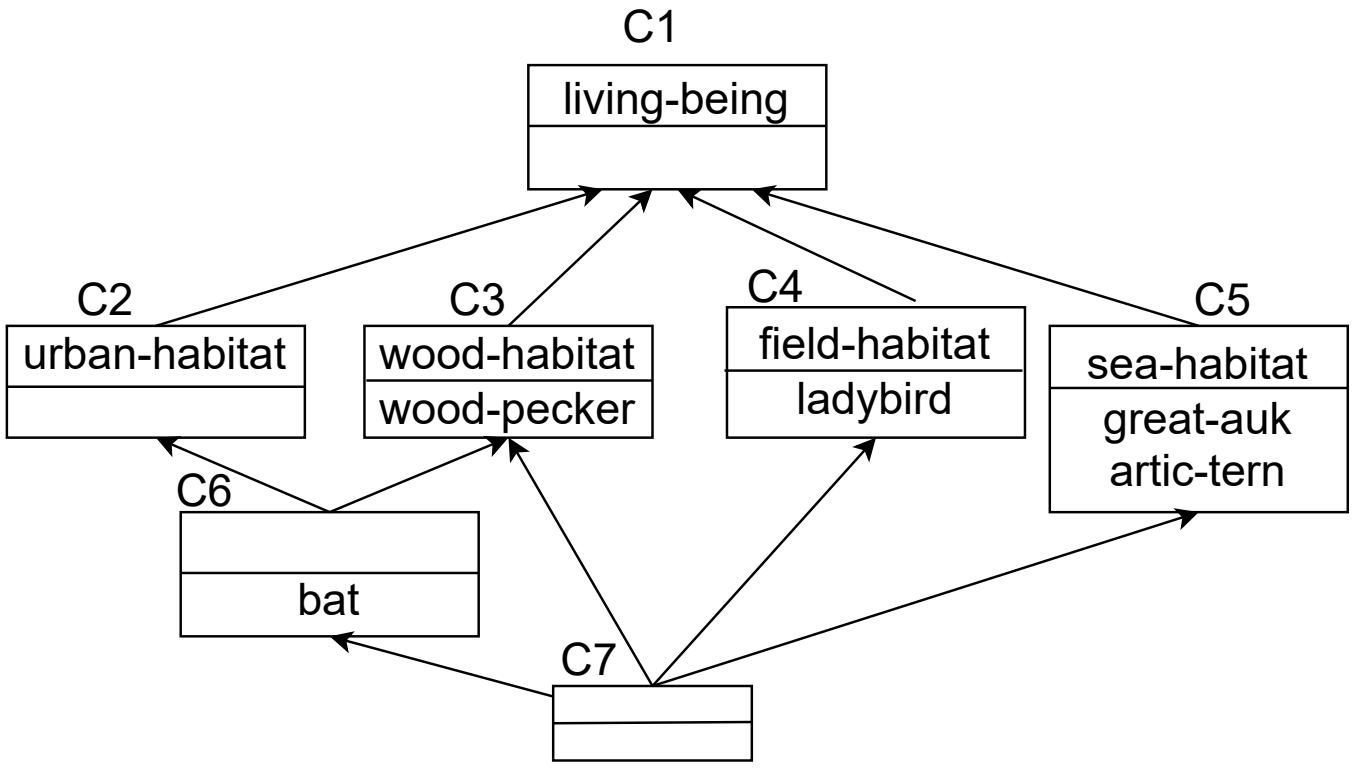
Logical Constraints: Mutual exclusion



Intersection of C1, C2, C3 extents is empty

$$\neg(\textit{red-bill} \wedge \textit{migratory} \wedge \textit{elytra})$$

Logical Constraints: Or

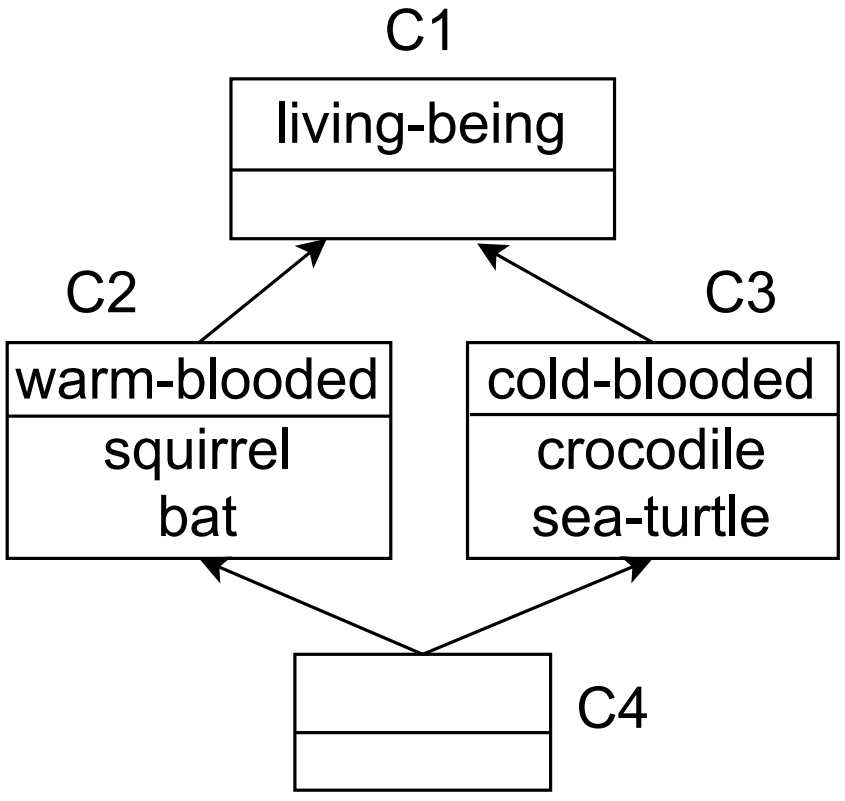


Extent of C1 is covered by extents of C2, C3, C4 and C5
(possibly non disjoint)

$$Extent(C1) = Extent(C2) \cup Extent(C3) \cup Extent(C4) \cup Extent(C5)$$

$$living\text{-}being \implies urban\text{-}habitat \vee wood\text{-}habitat \vee field\text{-}habitat \vee sea\text{-}habitat$$

Logical Constraints: Xor

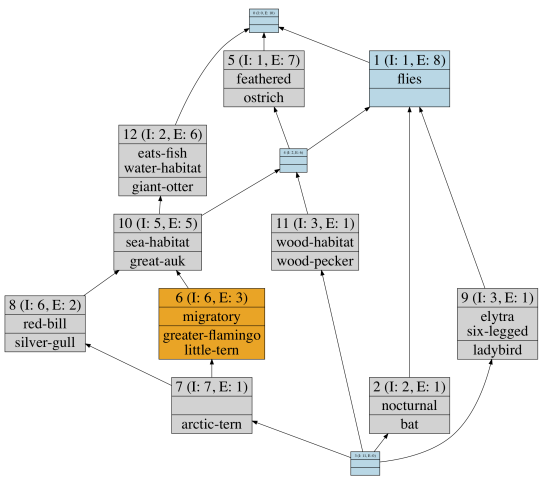


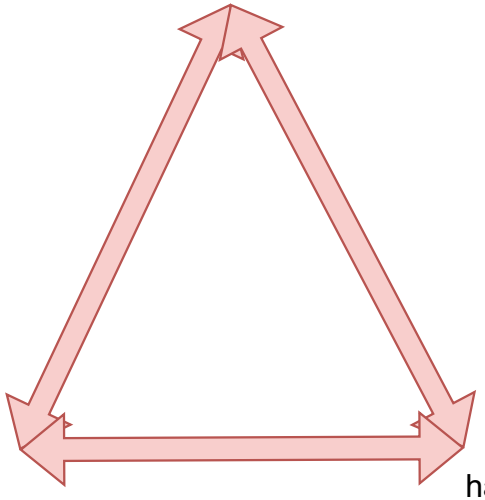
Extents of C2, C3 form a partition of C1 Extent
$Extent(C1) = Extent(C2) \cup Extent(C3)$ and
$Extent(C2) \cap Extent(C3) = \emptyset$

$$\boxed{\textit{living-being} \implies \textit{warm-blooded} \oplus \textit{cold-blooded}}$$

Sum up

	flies	nocturnal	feathered	migratory	red-bill	elytra	sea-habitat	wood-habitat	six-legged	eats-fish	water-habitat
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×





<0> flies,feathered,sea-habitat,wood-habitat,eats-fish,water-habitat => nocturnal,migratory,red-bill,elytra,six-legged
(...)
<0> flies,nocturnal,elytra,six-legged => feathered,migratory,red-bill,sea-habitat,wood-habitat,eats-fish,water-habitat
(...)
<1> nocturnal => flies\\
(...)
<5> flies,eats-fish,water-habitat => feathered,sea-habitat\\(...)

The magic triangle

Agenda

FCA for various data types

- Multi-valued attributes : integers, double, terms, structures, symbolic objects, etc. (Ganter et Wille, Diday, Polaillon, ...)
- Fuzzy (Belohlavek et al., Cabrera, Cordero, Enciso, Mora, Lòpez-Ròdriguez, Ojeda-Aciego et al., Cornejo, Medina et al., Yahia et al., Dubois, Prade ...)
- Value taxonomies (Godin et al., Carpineto et Romano, ...)
- Logical description (Chaudron et al., Ferré et al., ...)
- Graphs (Ganter and Kuznetsov, Liquière, Prediger et Wille, Kötters et al., Graph-FCA Ferré et al....)
- Multi-relational, RCA (Priss, Rouane et al., ...); RCA+Fuzzy (Boffa et al.)
- Polyadic (Sacarea, Tronca et al.)
- Sequences (Boukhetta, Demko, Bertet et al., Buzmakov et al.)
- Temporal data (Wolff et al., Nica, Braud, Dolques, Le Ber et al., Boukhetta, Demko, Bertet et al.)
- Pattern Structures (Ganter et al., Kuznetsov et al., Napoli et al., Buzmakov et al.)

Scaling: Multi-valued attributes

rent	begin	end
flat1	2000	2002
flat2	2000	2005
flat3	2010	2019
flat4	2020	

year	$\geq$2000	$\geq$2010	$\geq$2020
2000	×		
2010	×	×	
2020	×	×	×

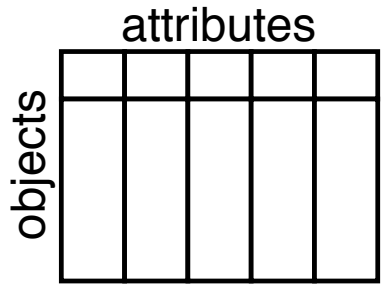
year	[2000,2010]	[2005,2010]	[2000,2005]	[2000,2020]
2002	×		×	×
2005	×	×	×	×
2019				×

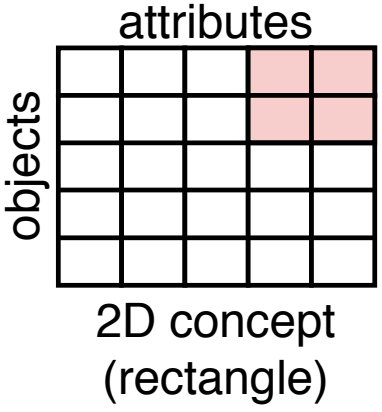
	begin			end			
rent	$\geq$2000	$\geq$2010	$\geq$2020	[2000,2010]	[2005,2010]	[2000,2005]	[2000,2020]
flat1	x			x		x	x
flat2	x			x	x	x	x
flat3	x	x					x
flat4	x		x				

[Ganter & Wille, 1999; Kaytoue et al, 2011] - Credits example P. Keip

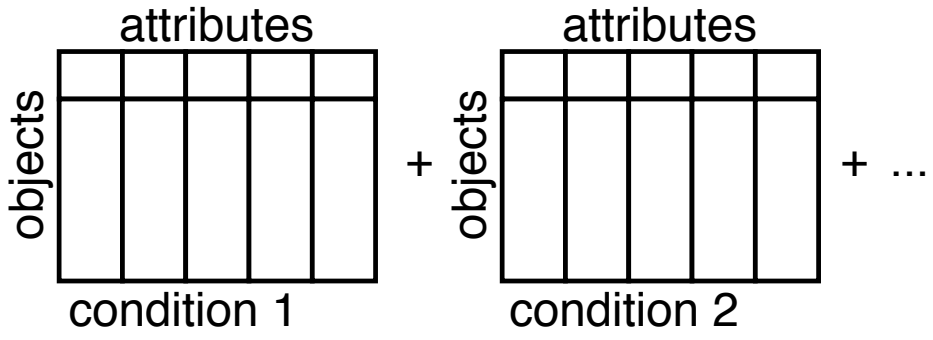
Triadic concept analysis

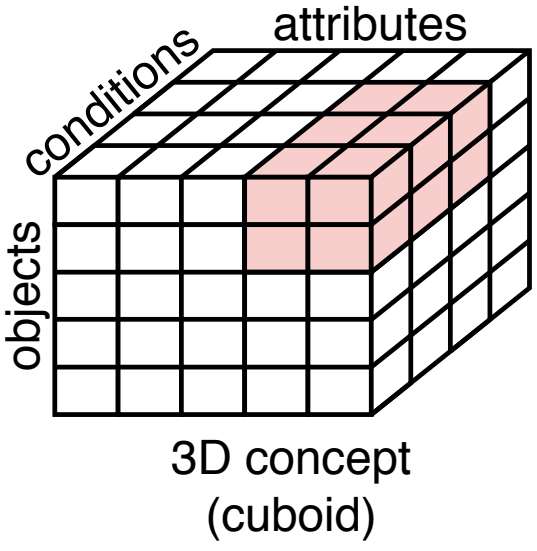
FCA





TCA





Pattern Structures

Pattern structures, Ganter and Kuznetsov 2001

In the following framework:

- a set of objects G
- a set of descriptions $(D, \sqcap)$ which is a semilattice
- a partial order on D: $a \sqsubseteq b$ iff $a \sqcap b = a$ (a is subsumed by b)
- a map $\delta : G \to D$

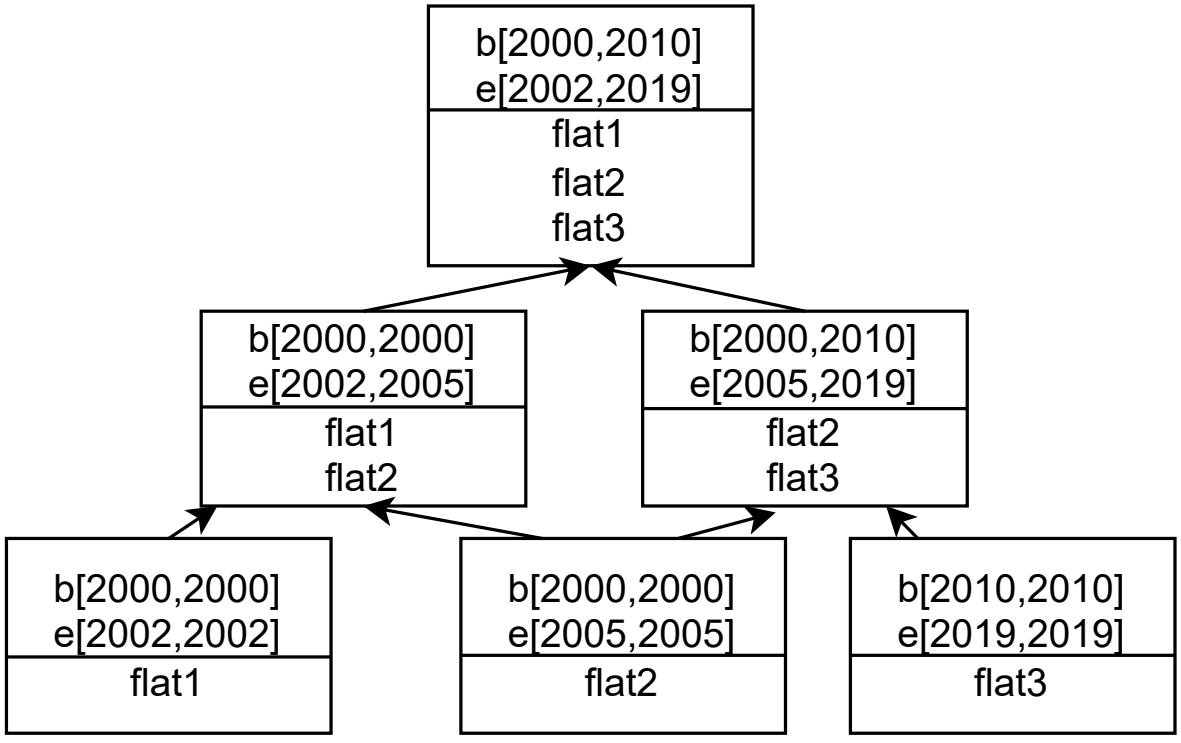
A Galois connection (f, g) can be defined:

- $\forall A \subseteq G, f(A) = \sqcap_{g \in A} \delta(g)$
- $\forall d \in D, g(d) = \{g \in G | d \sqsubseteq \delta(g)\}$

Ex.1 Interval Pattern Structures [Kaytoue et al., 2011]

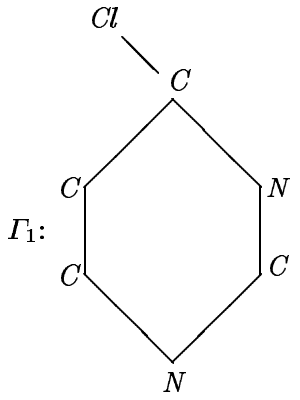
rent	begin	end
flat1	[2000,2000]	[2002,2002]
flat2	[2000,2000]	[2005,2005]
flat3	[2010,2010]	[2019,2019]

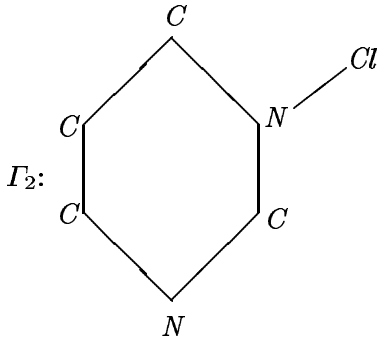
$$[a, b] \sqcap [c, d] = [min(a, c), max(b, d]$$



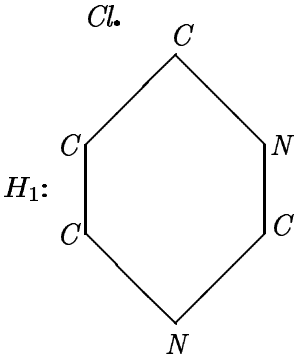
Ex.2 Graph Pattern Structures [Ganter&Kuznetsov, 2001]

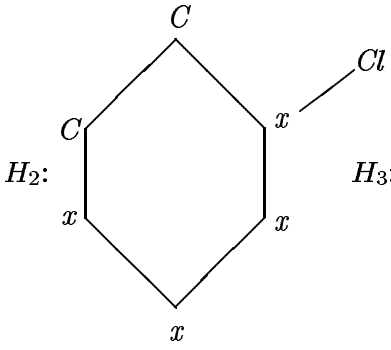
$G1$

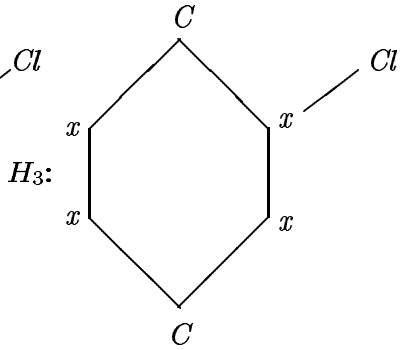




$G2$







$G1 \sqcap G2$

Relational Concept Analysis (RCA)

[Huchard et al., 2007, Rouane Hacène et al. 2013]





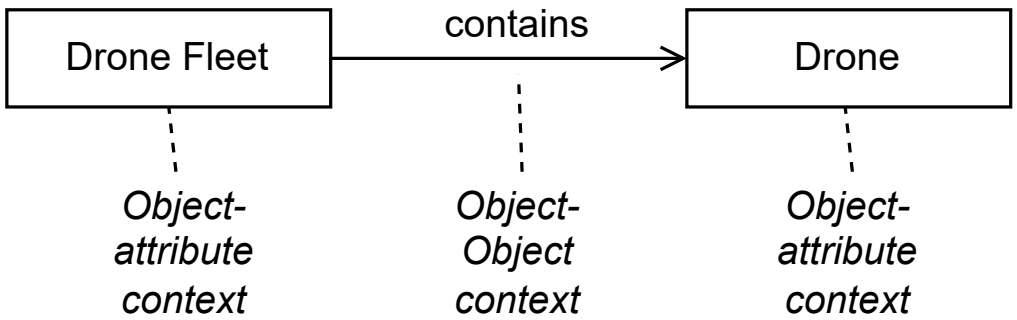
Principles

- Extends the purpose of FCA for taking into account object categories and links between objects
- Main principles:
  - a relational model based on the entity-relationship model (with binary relationships)
  - integrate relations in formal contexts between objects as *relational* attributes
  - various operators (quantifiers) inspired by description logics
  - iterative and tunable process
- RCA provides a set of interconnected lattices
- Produced structures can be written in DLs

Relational Context Family (RCF)

A RCF $\mathcal{F}$ is a pair (K, R) with:

- K is a set of object-attribute contexts $K_i = (O_i, A_i, I_i)$
- R is a set of object-object contexts $R_j = (O_k, O_l, I_j)$
  - (O_k, O_l) are the object sets of formal contexts $(K_k, K_l) \in K^2$
  - $I_j \subseteq O_k \times O_l$
  - K_k is the *source/domain context*
  - K_l is the *target/range context*
  - we may have $K_k = K_l$



Drone types (Object-Attribute context)

Drone	Gimbal	GPS	GLONASS	Avoidance	Headless	Altitude Hold	FT I 10	FT ge 10	FT ge 20
Syma X4S Assault 					×		×		
Syma X8G 					×		×		
Parrot Bebop 		×				×		×	
DJI Ryze Tello 						×		×	
Hubsan X4 H502S 		×			×	×		×	
Aosenma CG035 GPS FPV 	×	×			×	×		×	
DJI Mavic Air 	×	×	×	×	×	×			×
Yuneec Typoon H Pro 	×	×	×	×	×	×			×

https://www.thedronechart.com Hum, typo, here. Typoon in World of Warcraft? Teaspoon?

Drone fleets (Object-Attribute context)

DroneFleet	mission:agri	mission:rescue	mission:Audiovisual	archi:centralized	archi:distributed	guidance:human	guidance:ML	control:leaderFollower	control:behaviorBased
agriDF0	×			×		×		×	
agriDF1	×				×	×		×	
rescueDF2		×			×	×	×		×
rescueDF3		×			×	×	×		×
audiovisualDF4			×		×	×			×

Contains (Object-Object Context)

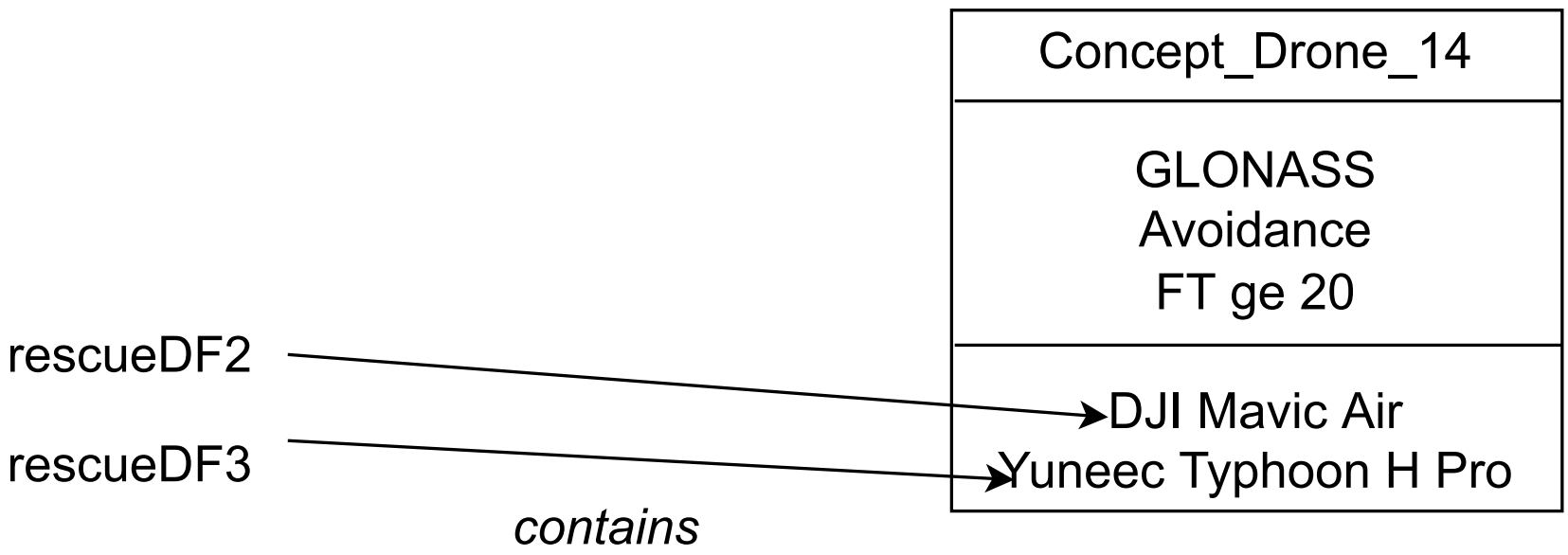
contains	Syma X4S Assault	Syma X8G	Parrot Bebop	DJI Ryze Tello	Hubsan X4 H502S	Aosenma CG035 GPS FPV	DJI Mavic Air	Yuneec Typoon H Pro
agriDF0	x							
agriDF1	x	x					x	x
rescueDF2							x	
rescueDF3								x
audiovisualDF4			x	x	x		x	x

$\Rightarrow$ `rescueDF2` and `rescueDF3` do not share drone types

Relational attributes

`rescueDF2` and `rescueDF3` do not share concrete drone types but they share the fact that all their drones with `GLONASS`, `GPS`, `FT` $\geq$ 20, etc.

Relational attribute: $\exists\forall contains(Concept\_Drone\_14)$



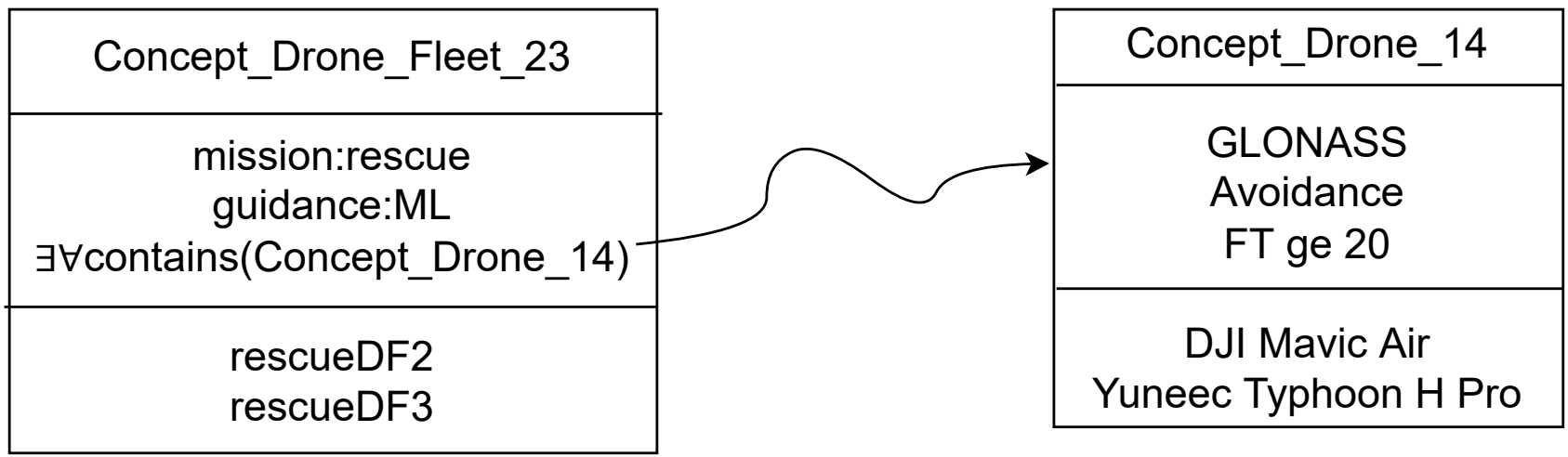
Drone fleets (Extended Object-Attribute context)

DroneFleet	mission:agri	mission:rescue	mission:Audiovisual	archi:centralized	archi:distributed	guidance:human	guidance:ML	control:leaderFollower	control:behaviorBased		$\exists\forall contains(Concept\_Drone\_14)$	
agriDF0	×			×		×		×				
agriDF1	×				×	×		×				
rescueDF2		×			×	×	×		×		×	
rescueDF3		×			×	×	×		×		×	
audiovisualDF4			×		×	×			×			

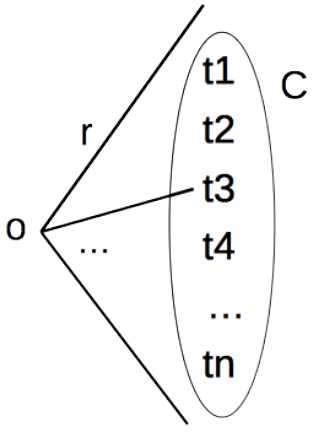
Relational attributes

`rescueDF2` and `rescueDF3` do not share concrete drone types but they share the fact that all their drones with `GLONASS`, `GPS`, `FT` $\geq$ 20, etc.

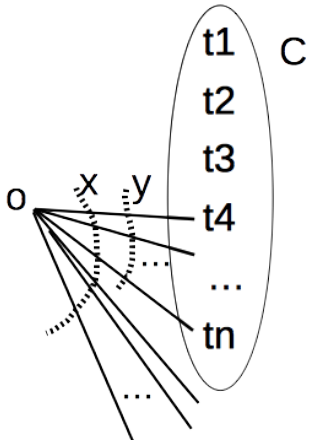
Relational attribute: $\exists\forall contains(Concept\_Drone\_14)$



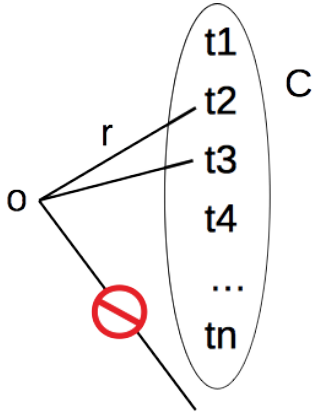
Quantifiers



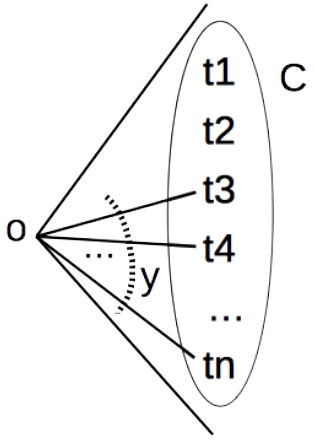
∃ r (C)



∃∀≥y*100/x% r (C)

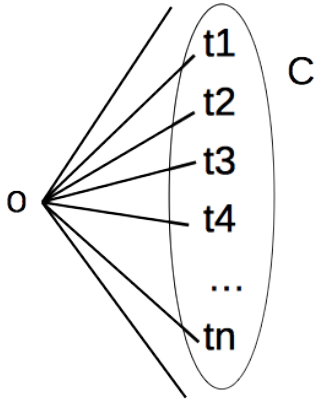


∃∀ r (C)



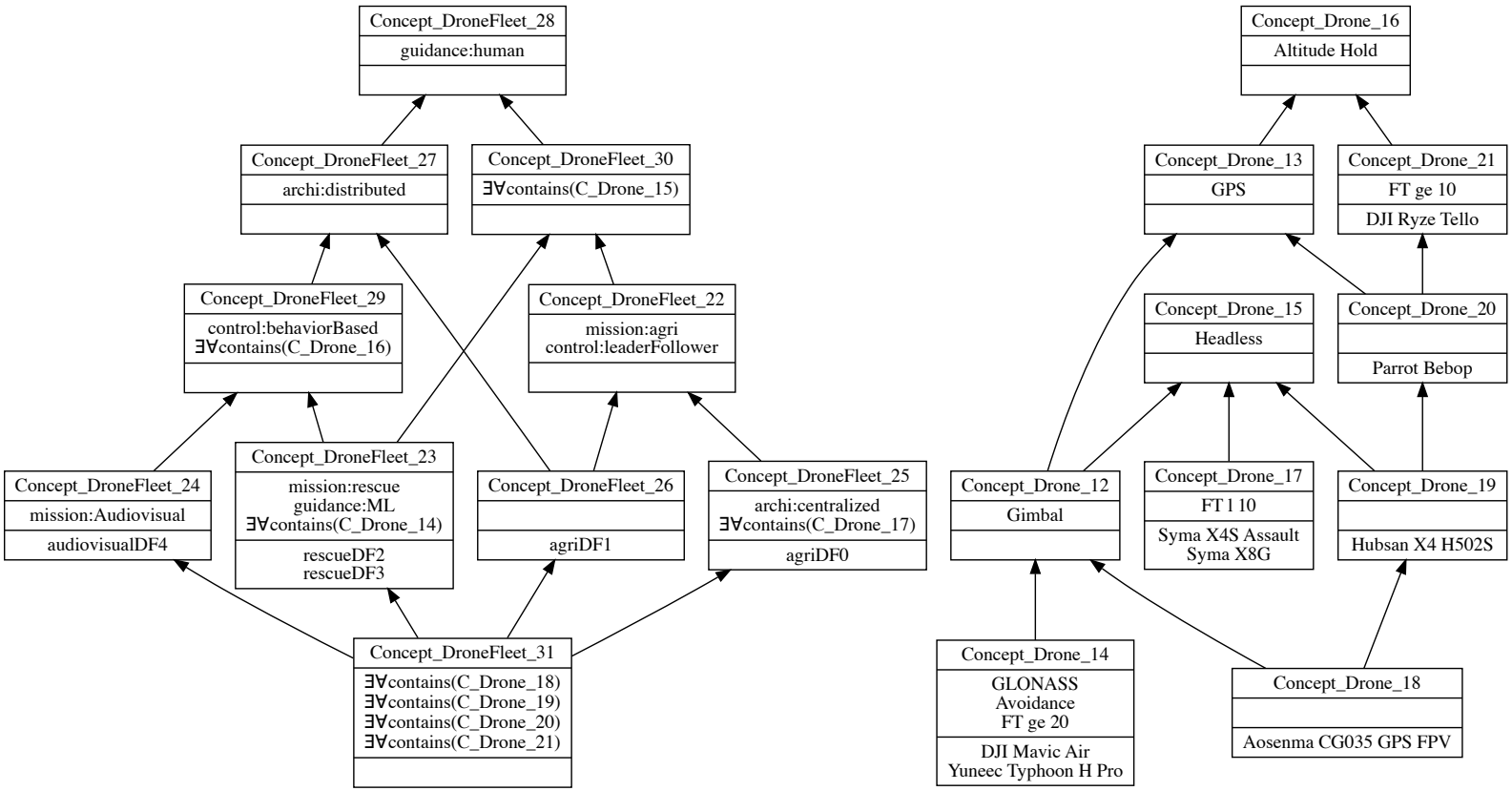
|C| = x

∃⊇≥y*100/x% r (C)

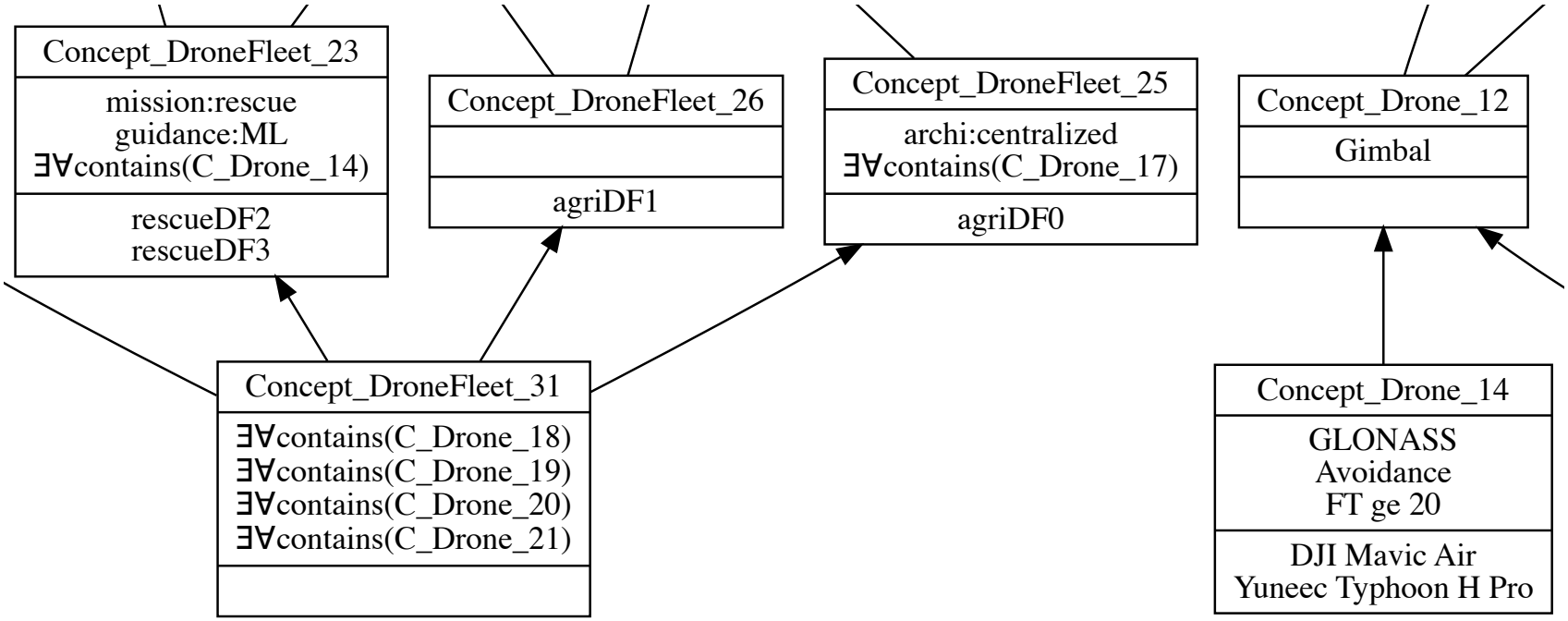


∃⊇ r (C)

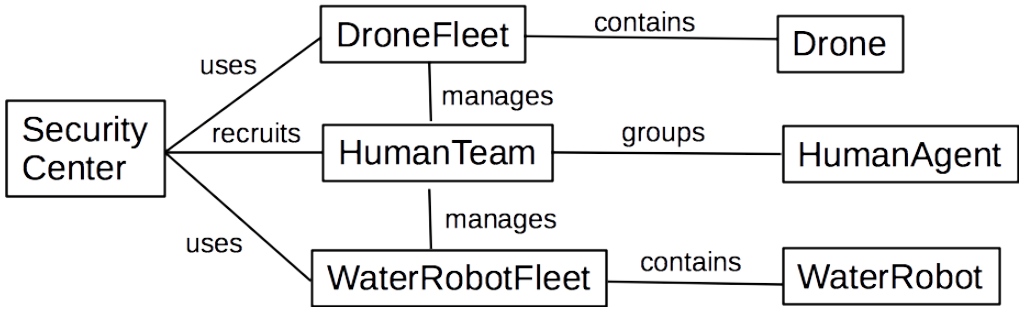
AOC-posets of drone fleets and drones



AOC-posets of drone fleets and drones (excerpt)



RCA in the general case



An iterative process

- Complex model with paths and cycles of any length
- Objects groups (concepts) are propagated along the paths and the cycles, step after step
- The process stops when no new concept appears

Agenda

FCA in Knowledge Discovery

- Knowledge structuring: ontologies, data models in KR/DB/SE Stumme, Maedche, Cimiano, Hotho, Sertkaya, Valtchev, ...
- Information retrieval, Exploratory KD: conceptual navigation, visualization, Decision Tree - like structure, Efficient index structure Ferré, Carpineto, Romano, Eklund, ...
- Data mining, Data analysis: Pattern mining, implication/association rules, functional dependencies, interestingness measures Valtchev et al., Missaoui et al., Bertet et al., Cordero et al., Valverde-Albacete et al. Baixeries et al. ...
- Knowledge elicitation: Concept discovery, concept definition, attribute exploration Ganter, Obidekov, Alam, Buzmakov, Codocedo, Napoli, ...
- Recommendation systems Ignatov, Kuznetsov, Codocedo, Napoli, ...
- Querying with answers classification Messai, Napoli, Azmeh et al., ...

Part of them in: Jonas Poelmans, Sergei O. Kuznetsov, Dmitry I. Ignatov, Guido Dedene: Formal Concept Analysis in knowledge processing: A survey on models and techniques. Expert Syst. Appl. 40(16): 6601-6623 (2013) + significant new work since

Applications (A few domains)

- Environment, biology, chemistry, health
- Linguistics, Text understanding
- Software engineering
- Communities, social network

Part of them in: Jonas Poelmans, Dmitry I. Ignatov, Sergei O. Kuznetsov, Guido Dedene: Formal concept analysis in knowledge processing: A survey on applications. Expert Syst. Appl. 40(16): 6538-6560 (2013) + significant new work since

Tools (A few examples)

More than 50 web applications, downloadable software or plugins at Uta Priss webpage `https://upriss.github.io/fca/fcasoftware.html`

- Algorithms: Conexp family, ToscanaJ, fcaR, GALACTIC
- Specializations: RCA software (Galicia, RCAexplore, FCA4J), Fuzzy (fcaR), polyadic (FCA Tools Bundle)
- IR and Search/Query engines: Credo family, Search Sleuth family, Camelis, Sparklis family
- Visualization+navigation: Latviz, RV-xplorer, ConceptCloud, RCAviz

Workshop Applications and Tools of Formal Concept Analysis@ICFCA2019
Workshop ETAFCA@CLA2022

Focus on Software Engineering

- Mostly well formatted and complete data: conceptual models, specifications, source code, traces, call graphs, git actions...
- Software engineers are proficient to understand and exploit the results

A large range of applications, e.g.

- Migration from procedural to object-oriented paradigm
- Class model / Class hierarchies refactoring from artefact description or artefact usage
- Organization of repositories of software artefacts: classes, components, web services
- Extraction of model transformation patterns
- Feature models synthesis in software product lines

Focus on class model refactoring: Motivation

Class models, ontologies

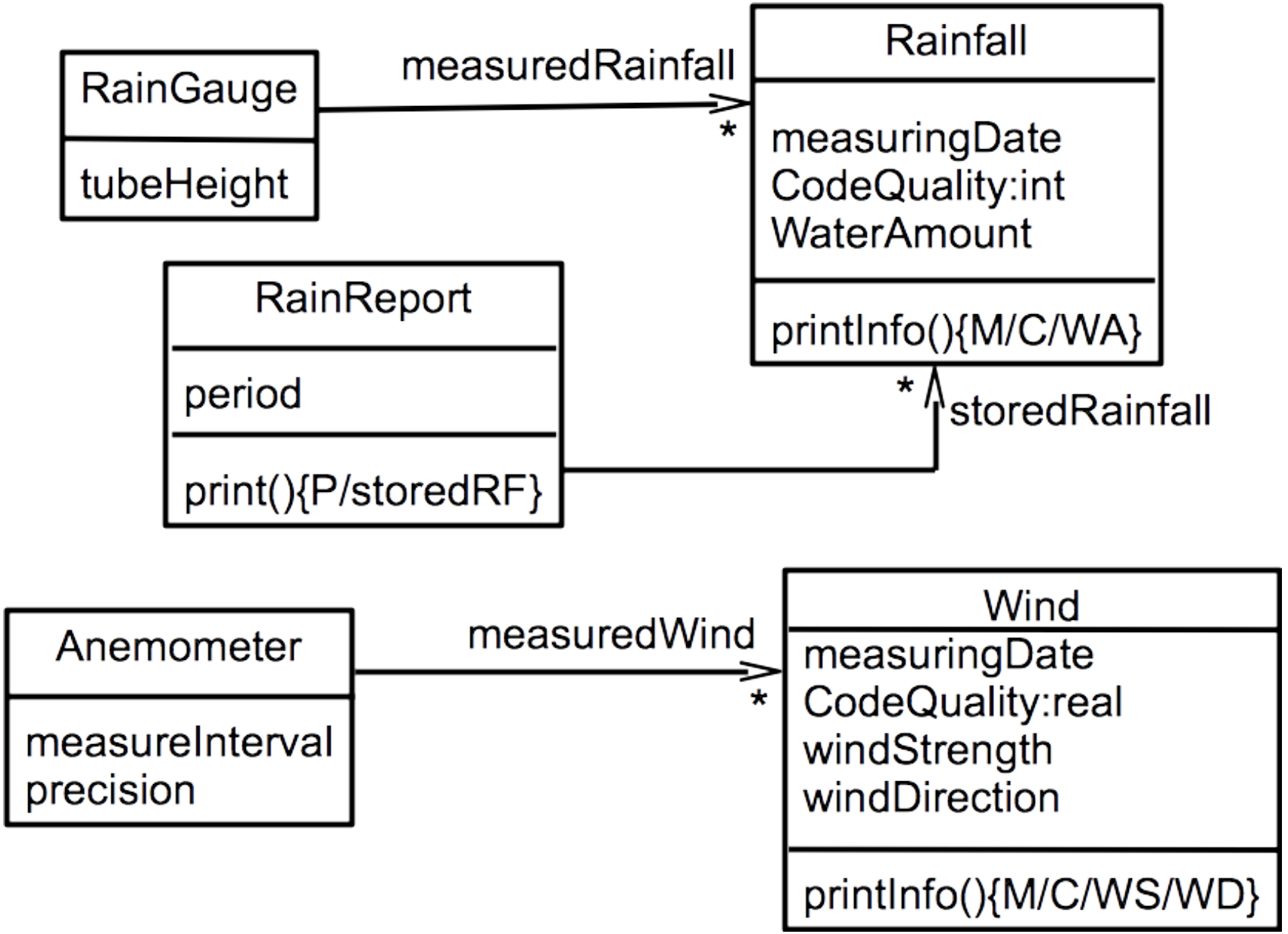
- Capturing and representing domain knowledge
- Highlight is-a relationships as the skeleton of the representation
- Encourage reuse and abstraction

Normal form model with RCA

- No redundancy
- All abstractions and specialization relationships are included
- Most compact structure

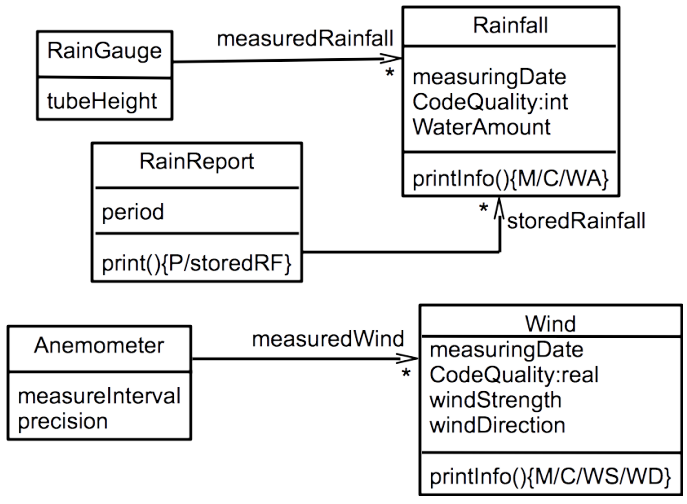
Seminal paper (FCA version): Godin & Mili, 1993

Focus on class model refactoring: Initial Model



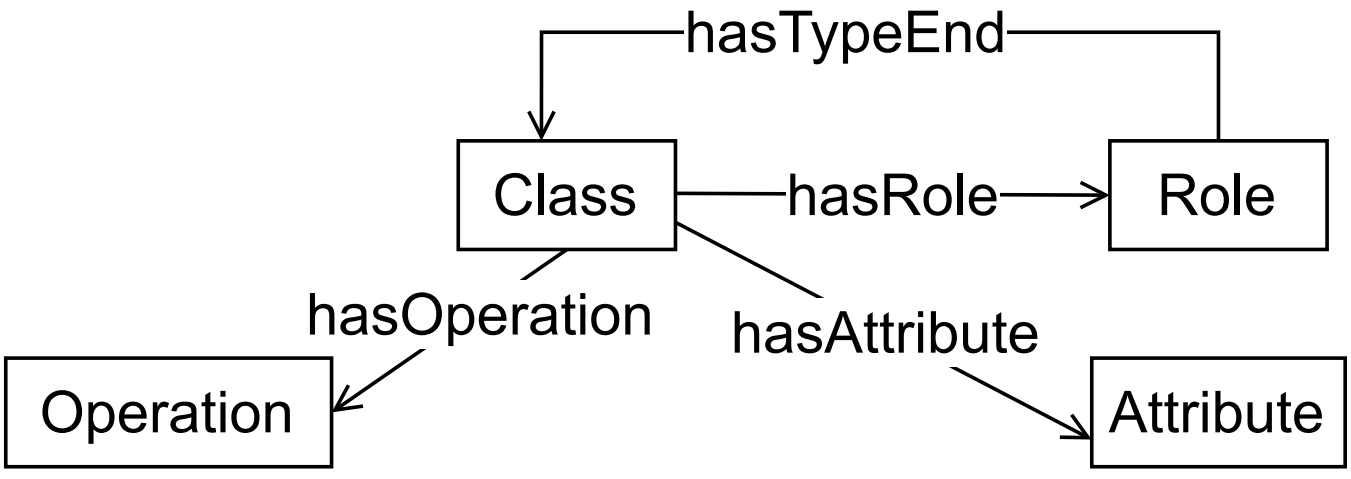
inspired by Pesticides Models, A. Miralles, IRSTEA

Focus on class model refactoring: Identified defaults



- `measuringDate codeQuality` repeated in `Rainfall` and `Wind` (missing *measure* abstraction)
- `RainGauge` et `Anemometer` are connected via `measuredXXX` (missing *Device* abstraction)
- repeated behavior in `print` methods (M/C = print measuringDate / codeQuality).

Focus on class model refactoring: Encoding principles



Kclass
RainGauge
Anemometer
Rainfall
Wind
RainReport

Kattribute	tubeHeight	measureInterval	precision	measuringDate	codeQuality	waterAmount	windStrength	windDirection	period	number	int	real
RG::tubeHeight	×											
A::measureInterval		×										
A::precision			×									
R::measuringDate				×								
W::measuringDate				×								
R::codeQuality					×					×	×	
W::codeQuality					×					×		×
R::waterAmount						×						
W::windStrength							×					
W::windDirection								×				
RR::period									×			

Krole	measuredRainfall	measuredWind	storedRainfall
RG::measuredRainfall	×		
RR::storedRainfall			×
A::measuredWind		×	

Koperation	print	MC	MCWA	MCWSWD	PstoredRF
RR::print	×				×
R::printinfo	×	×	×		
W::printinfo	×	×		×	

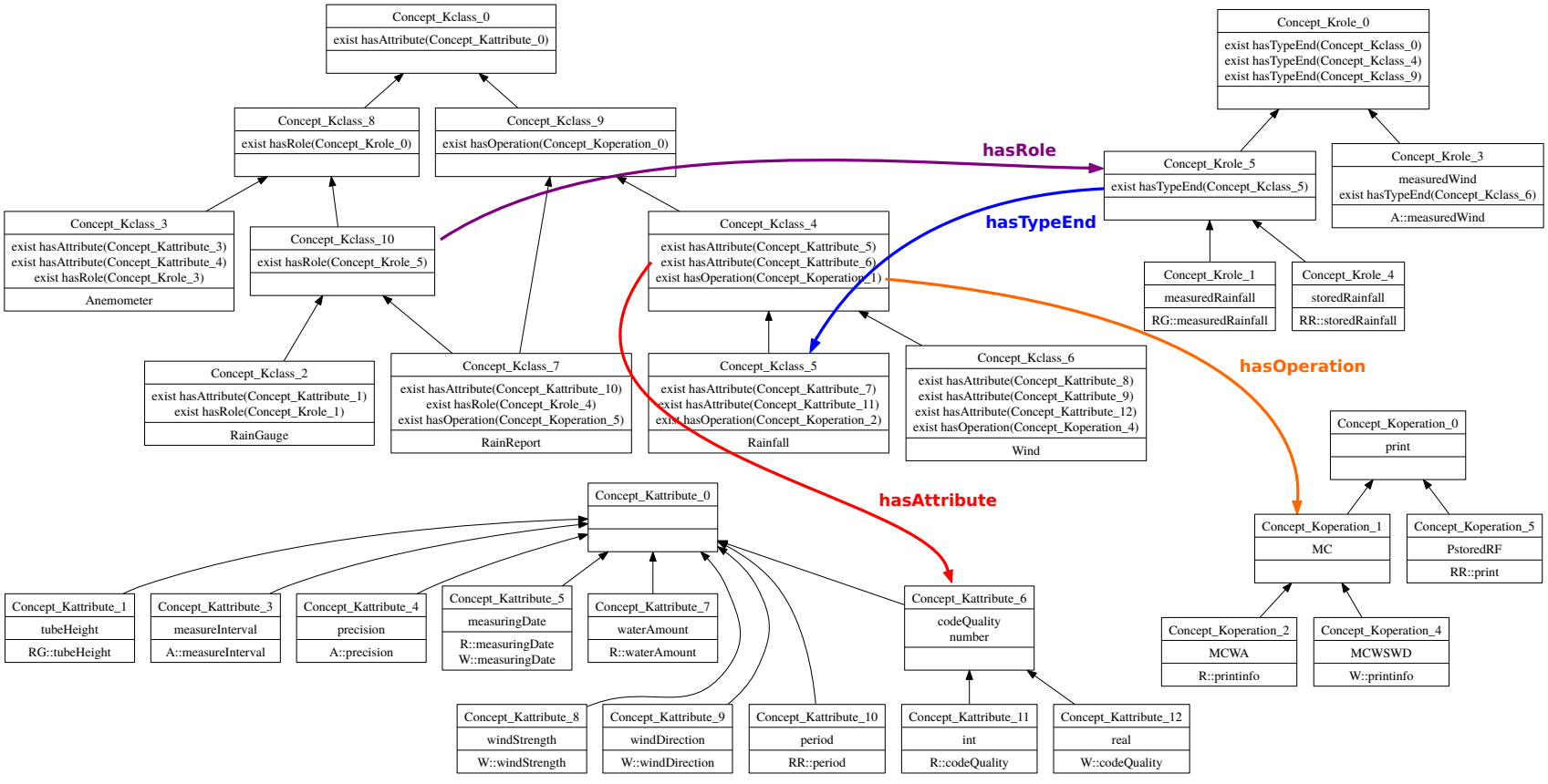
hasAttribute	RG::tubeHeight	A::measureInterval	A::precision	R::measuringDate	W::measuringDate	R::codeQuality	W::codeQuality	R::waterAmount	W::windStrength	W::windDirection	RR::period
RainGauge	×										
Anemometer		×	×								
Rainfall				×		×		×			
Wind					×		×		×	×	
RainReport											×

hasRole	RG::measuredRainfall	RR::storedRainfall	A::measuredWind
RainGauge	×		
Anemometer			×
Rainfall			
Wind			
RainReport		×	

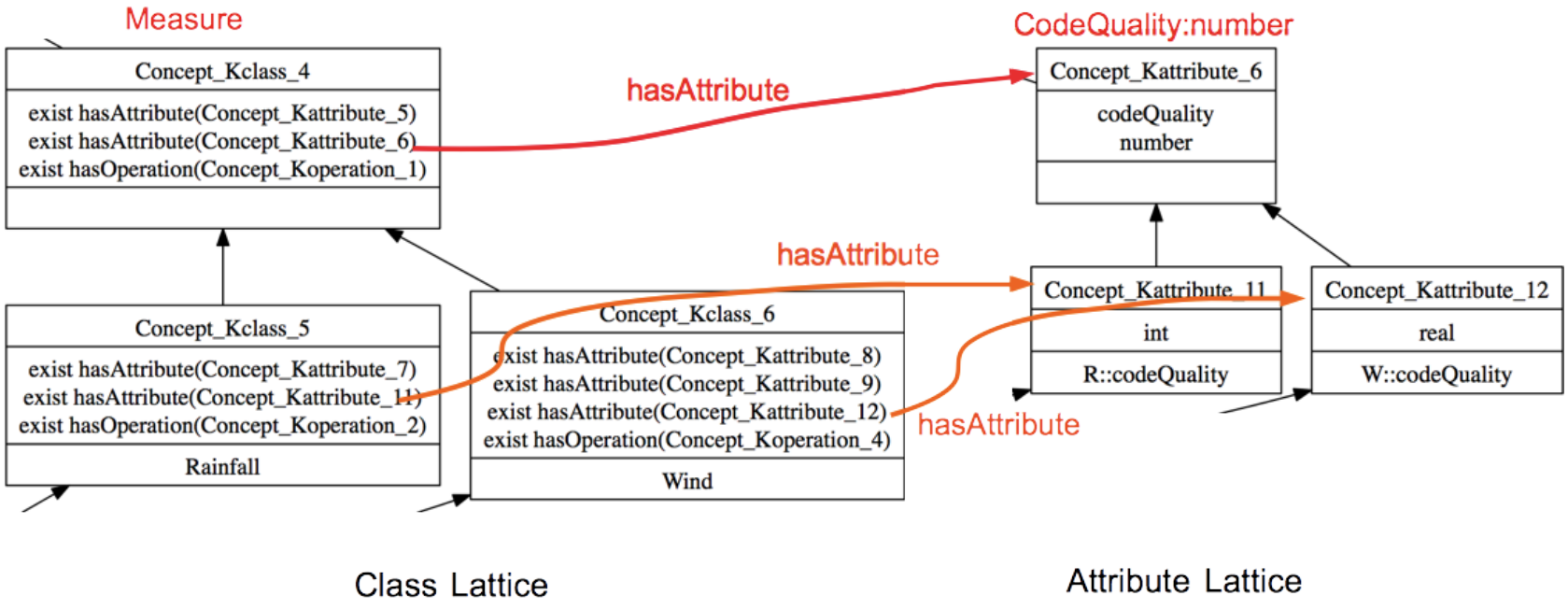
hasOperation	RR::print	R::printinfo	W::printinfo
RainGauge			
Anemometer			
Rainfall		×	
Wind			×
RainReport	×		

hasTypeEnd	Rainfall	Wind
RG::measuredRainfall	×	
RR::storedRainfall	×	
A::measuredWind		×

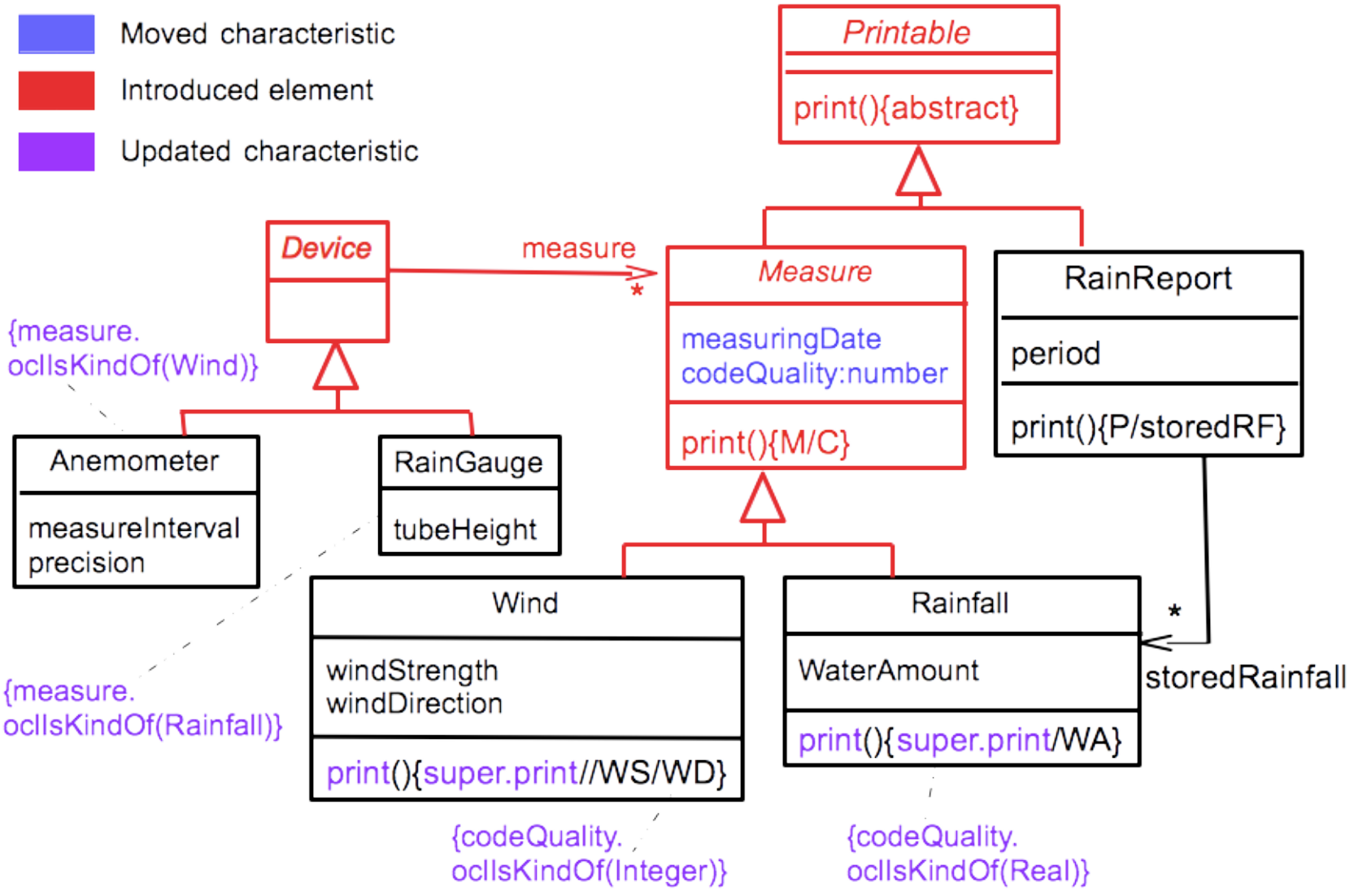
Focus on class model refactoring: Lattices



Focus on class model refactoring: Lattices excerpt



Focus on class model refactoring: Final Model



Focus on class model refactoring: references

- Design: PhDs theses of C. Roume (2004), A. Rouane Hacène (2006), J.-R. Falleri (2009), A. Osman Guédi (2013)
- Implementation and practice: France Télécom (M. Dao, 2004), IRSTEA (A. Miralles, 2012-...)
- Cousin methods: OODB schema refactoring (Lakhal et al., Missikov & Scholl, Rundensteiner), Ontology learning and refactoring (Rouane Hacène, Napoli, Valtchev et al.)

Focus on software product lines

Software Product Lines

A paradigm to discipline the construction of families of similar software (e.g. e-commerce, travel reservation), reduce the cost, increase quality and personalization

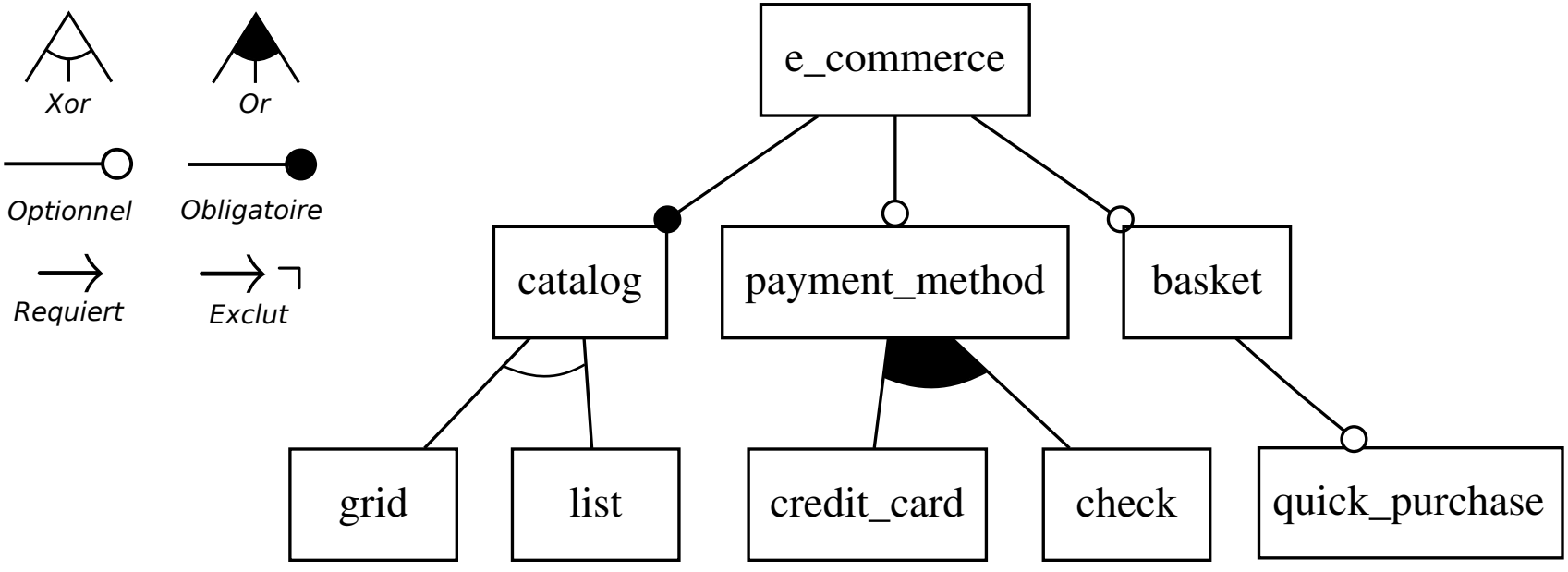
Main challenge

Capturing, representing and exploiting variability to be able to derive almost automatically software variants

Focus on software product lines: Feature Model

Feature Model

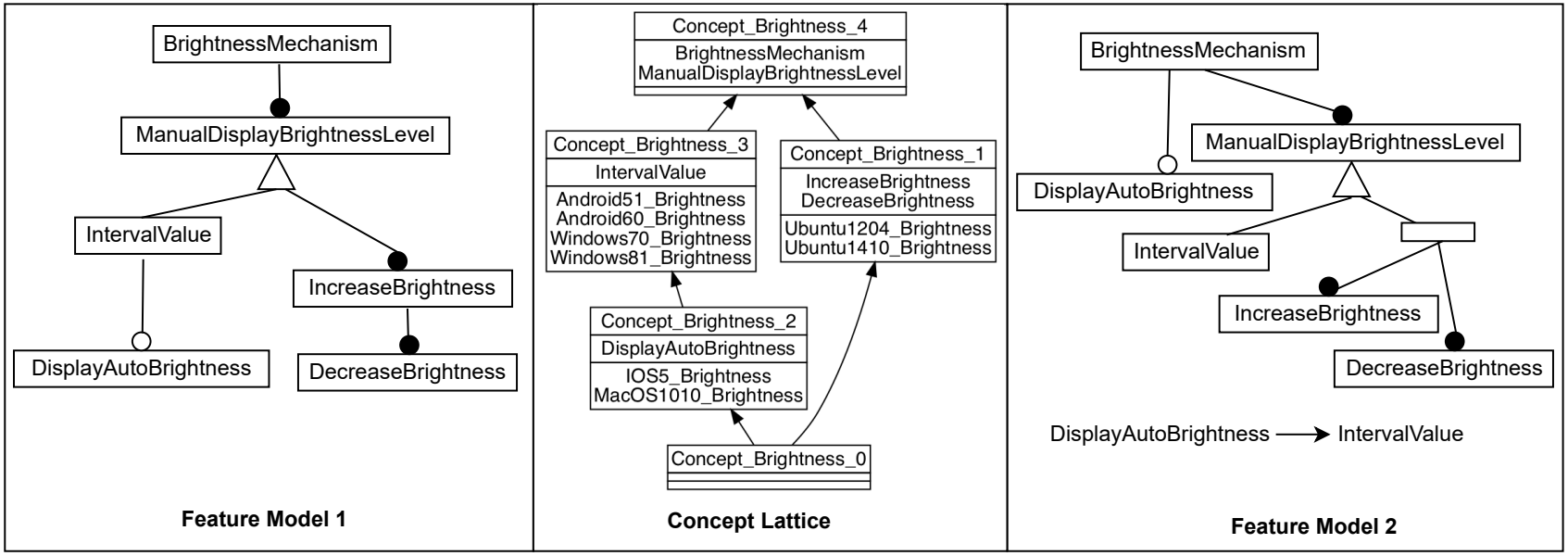
One of the most used variability models



Credits J. Galasso-Carbonnel

Focus on software product lines: FCA variability model

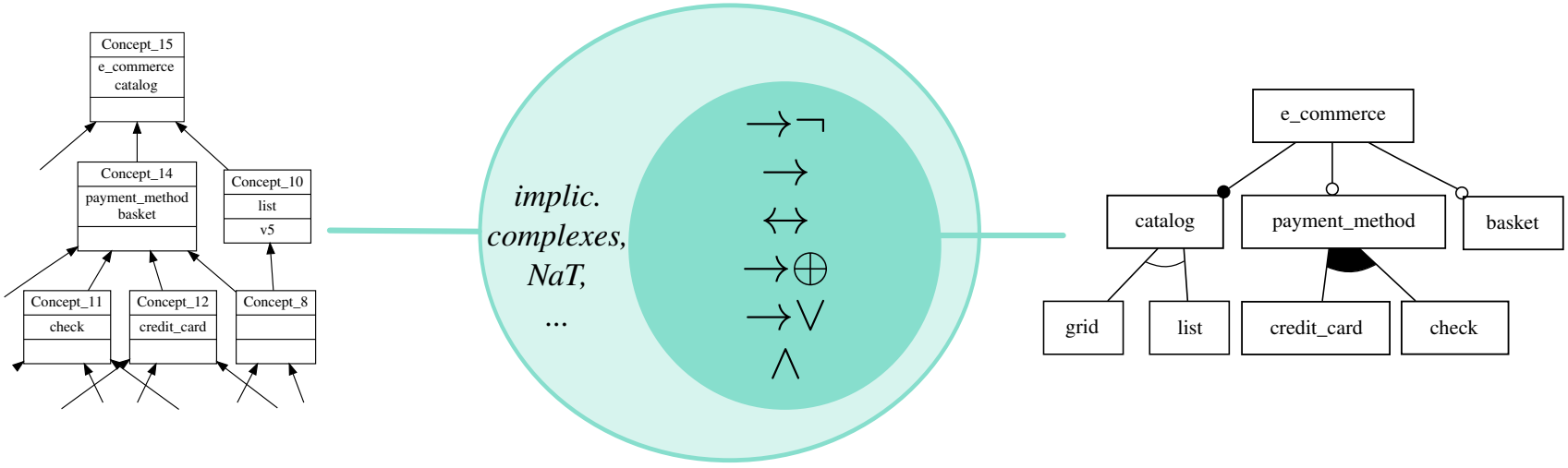
All FM with the same configuration set can be mapped or extracted from the concept lattice [Galasso-Carbonnel et al.]



Credits A. Waffo Kouhoué, Y. Bonavero

Focus on software product lines: FCA completeness

FCA is more powerful than FM: it allows to express any propositional logic formula [Galasso-Carbonnel et al.]



Credits J. Galasso-Carbonnel

Focus on software product lines:

Applications

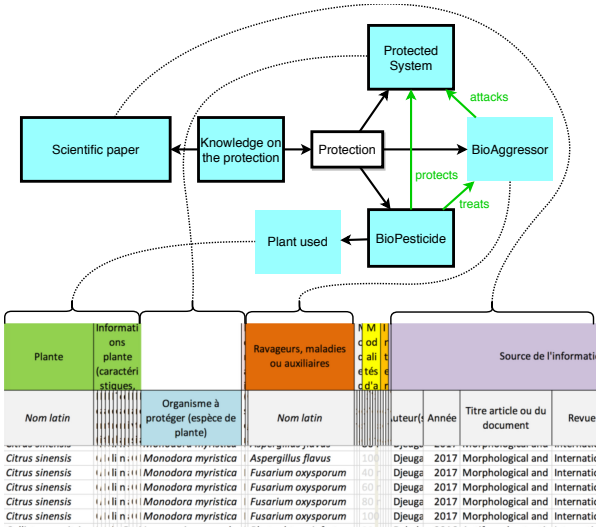
- Feature Model refactoring or synthesis (from a set of existing configurations)
- Feature Model operations: union, intersection
- Configuration recommendation

Evaluations

- SPLOT repository (`http://www.splot-research.org/`)
- Product comparison matrix (e.g. wikipedia)
- Visual accessibility options in operating systems

Focus on Agroecology: Knomana

- Find alternative solutions to chemical pesticides and antibiotics



- +45000 plant usage description
- +35 attributes
- +700 scientific papers

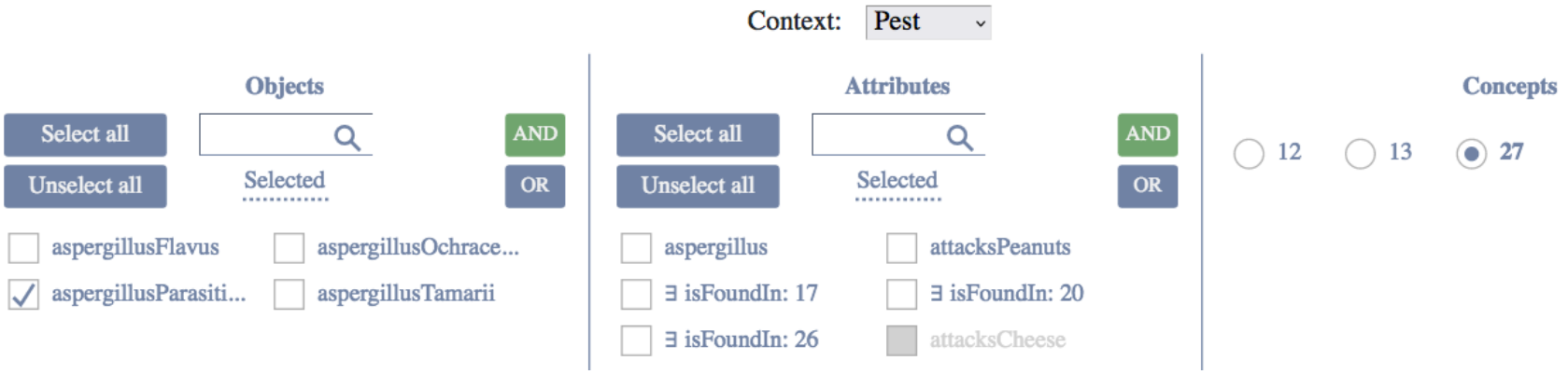
- Data exploration with RCAVIZ `rcaviz.lirmm.fr/`
- Implication Rules computed for FCA and RCA with `www.lirmm.fr/fca4j/`

Credit: P. Keip, P. Martin, P. Silvie

Focus on Agroecology: Knomana

Data exploration with RCAVIZ `rcaviz.lirmm.fr/`

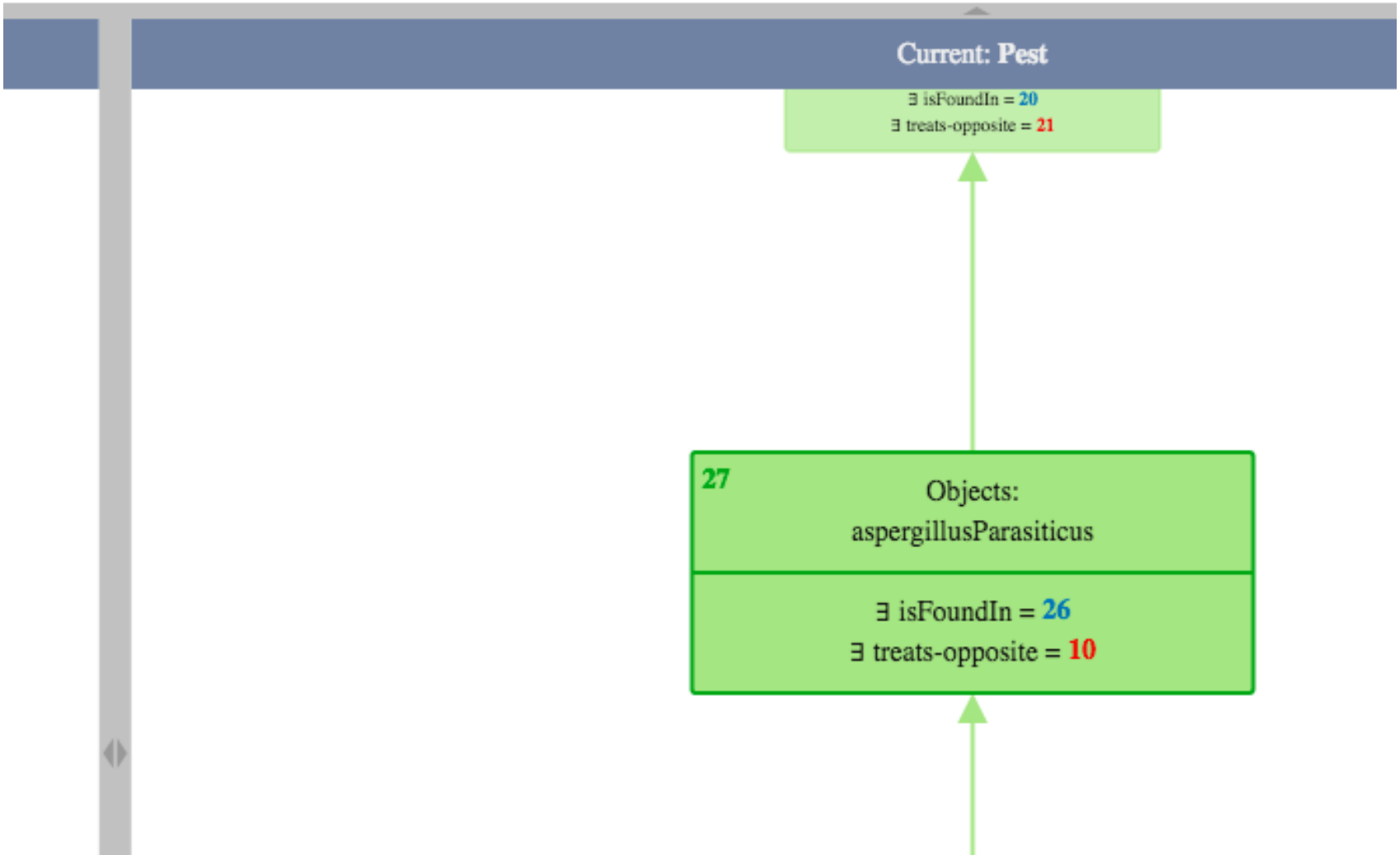
Aspergillus parasiticus attack in Benin



Credit example: A. Ouzerdine - RCAVIZ joint work with E. Muller, P. Martin, A. Sallaberry, P. Poncelet

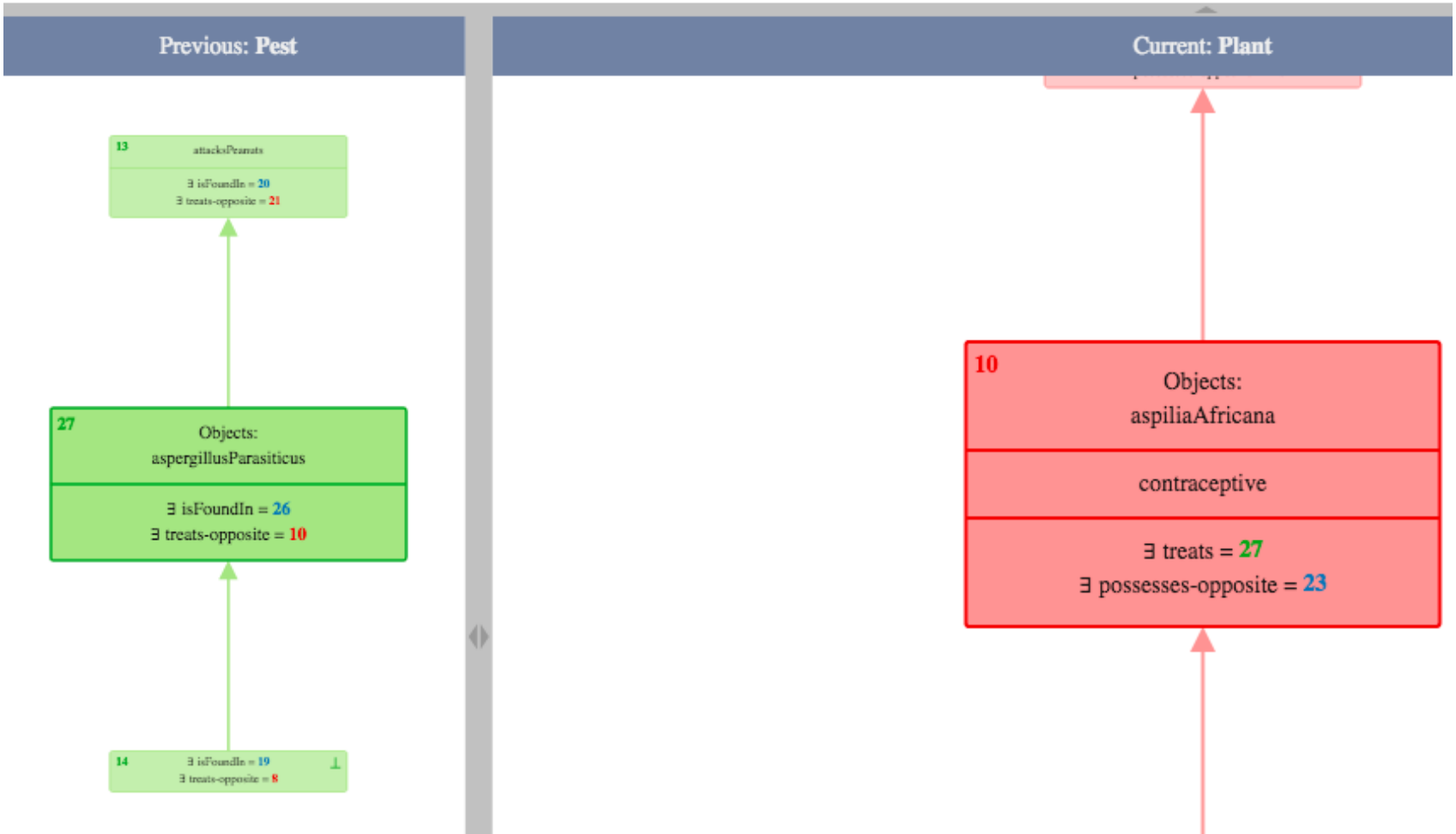
Focus on Agroecology: Knomana

Aspergillus parasiticus concept is highlighted; we follow relational attribute to Concept 10 (controlling plant)



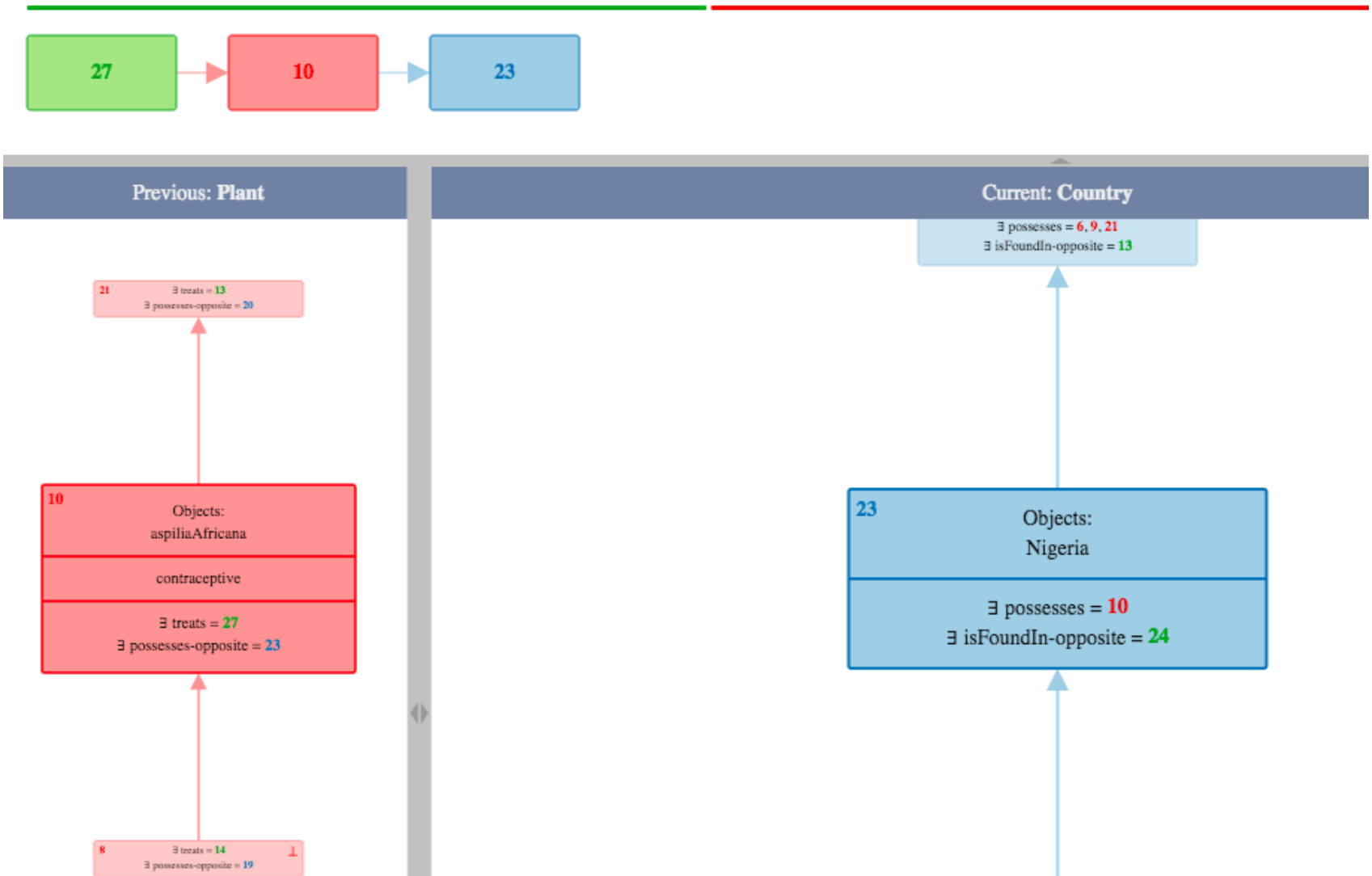
Focus on Agroecology: Knomana

Which is ... Aspilia Africana



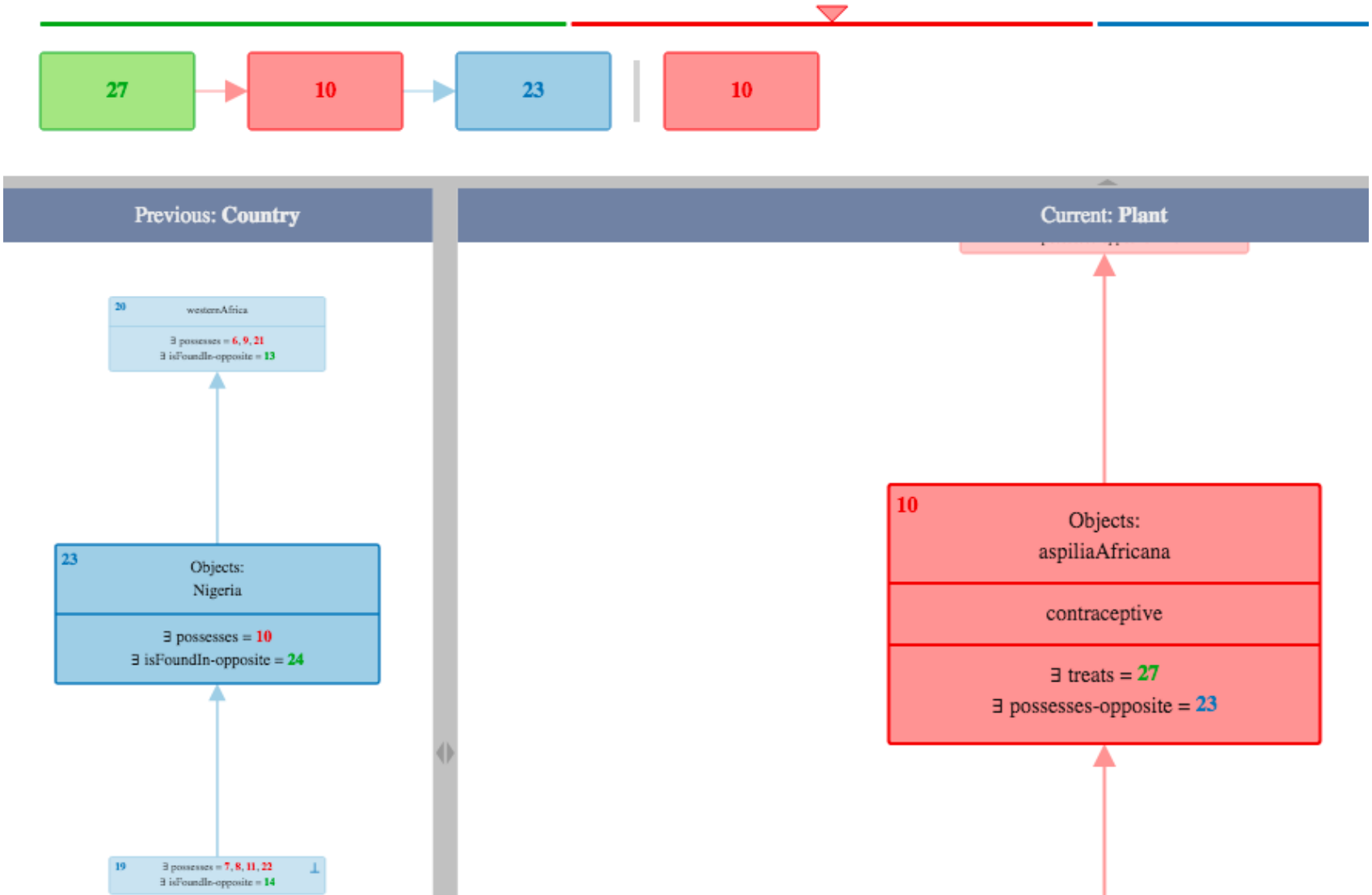
Focus on Agroecology: Knomana

Unfortunately, it can be found in Nigeria (not in Benin) - Concept 23



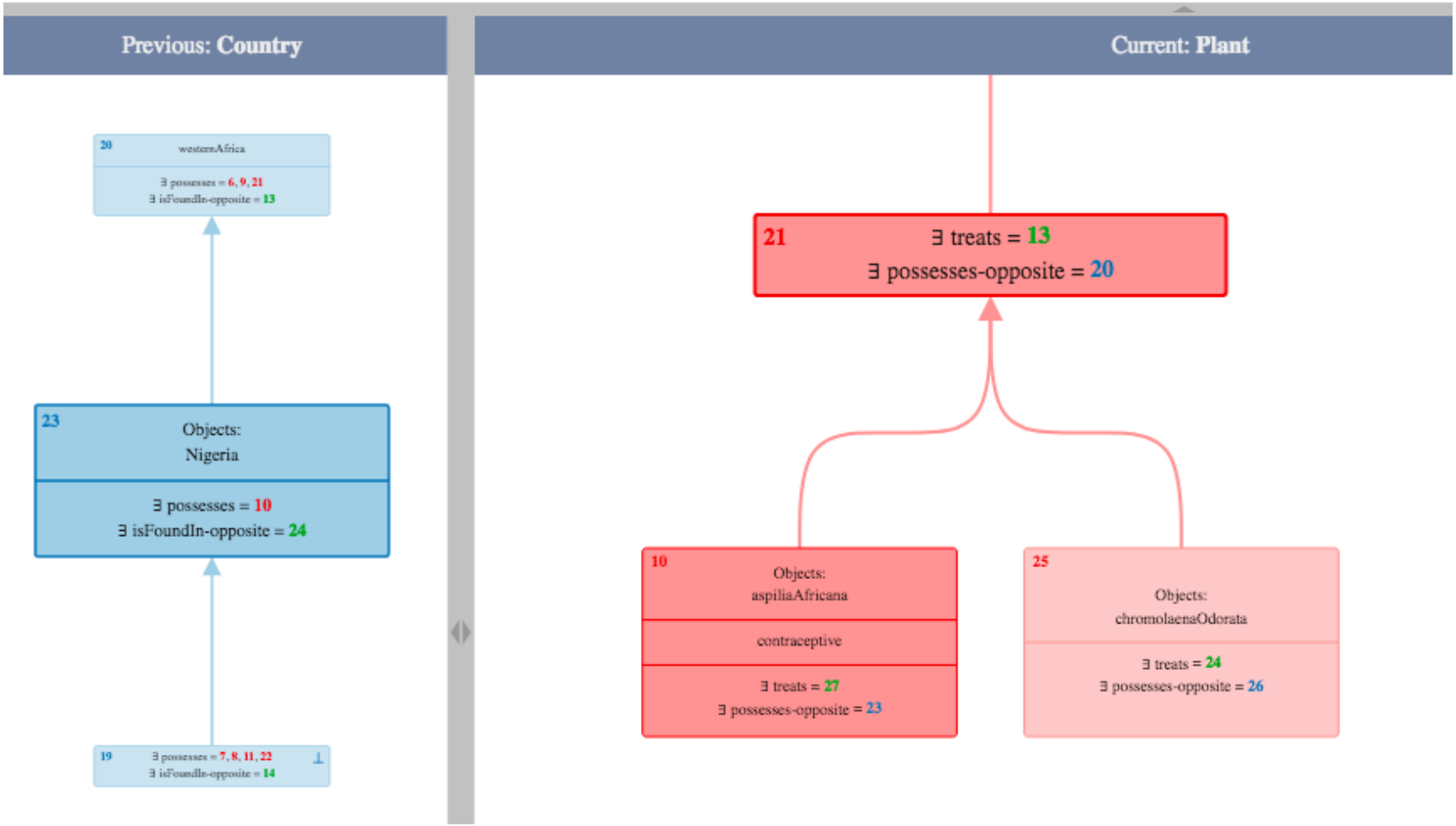
Focus on Agroecology: Knomana

Can we find in Benin an alternative to Aspilia Africana? We come back to Concept 10.



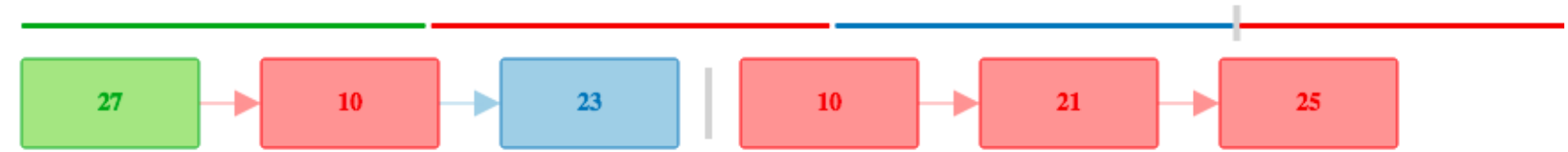
Focus on Agroecology: Knomana

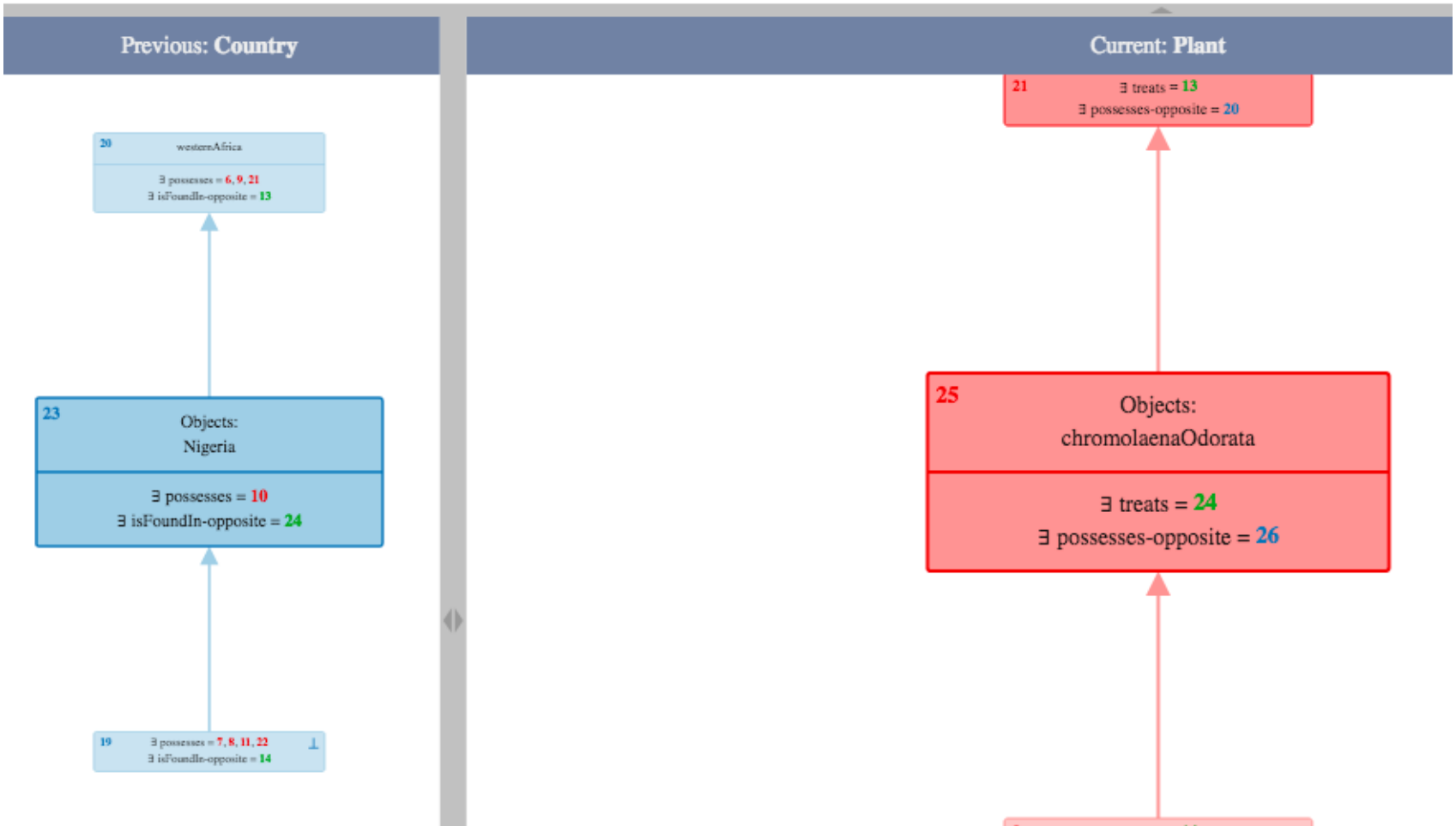
To find a cousin plant, we go up to Concept 21



Focus on Agroecology: Knomana

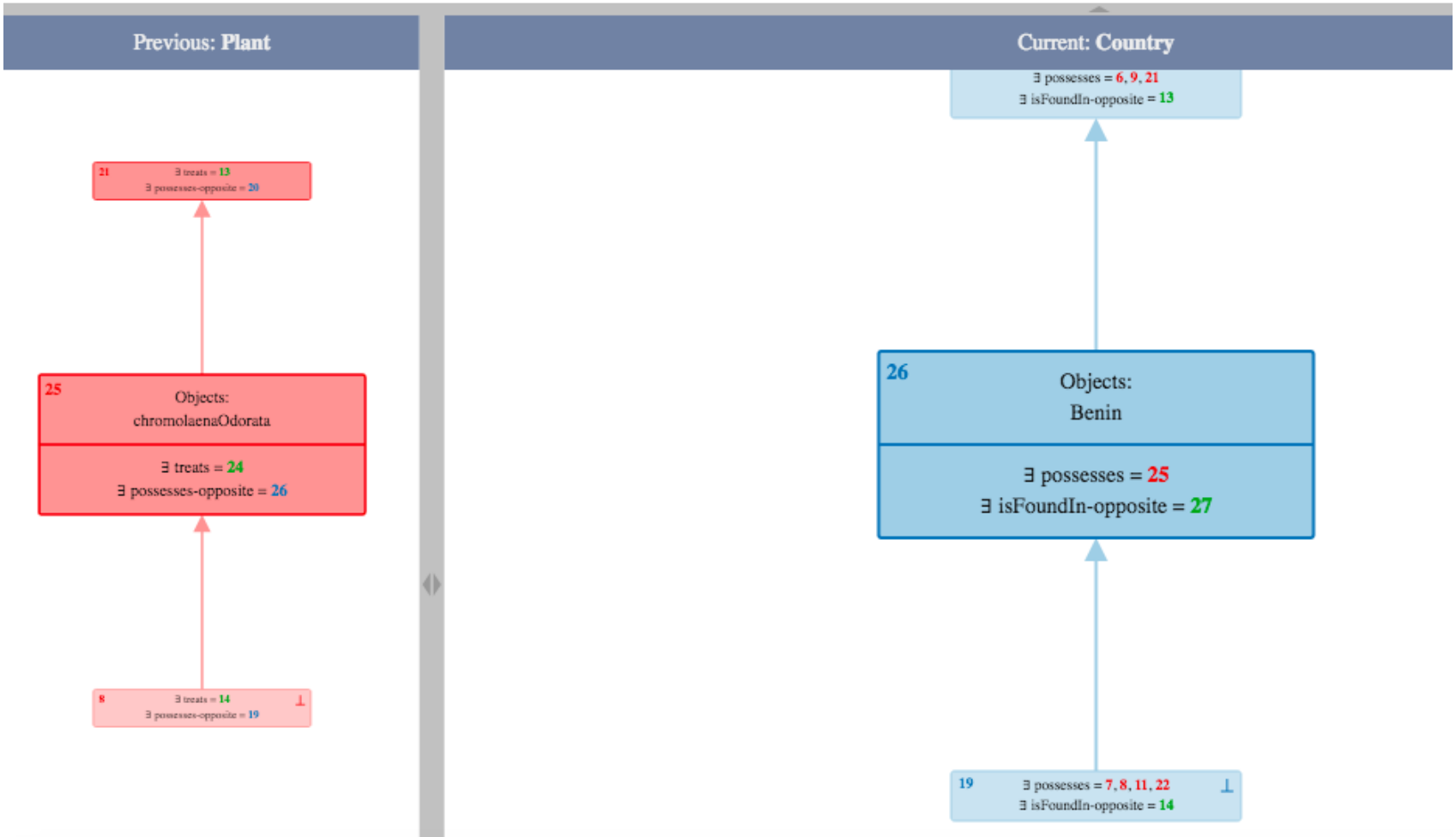
Then down to Concept 25, where we find Chromolaena odorata





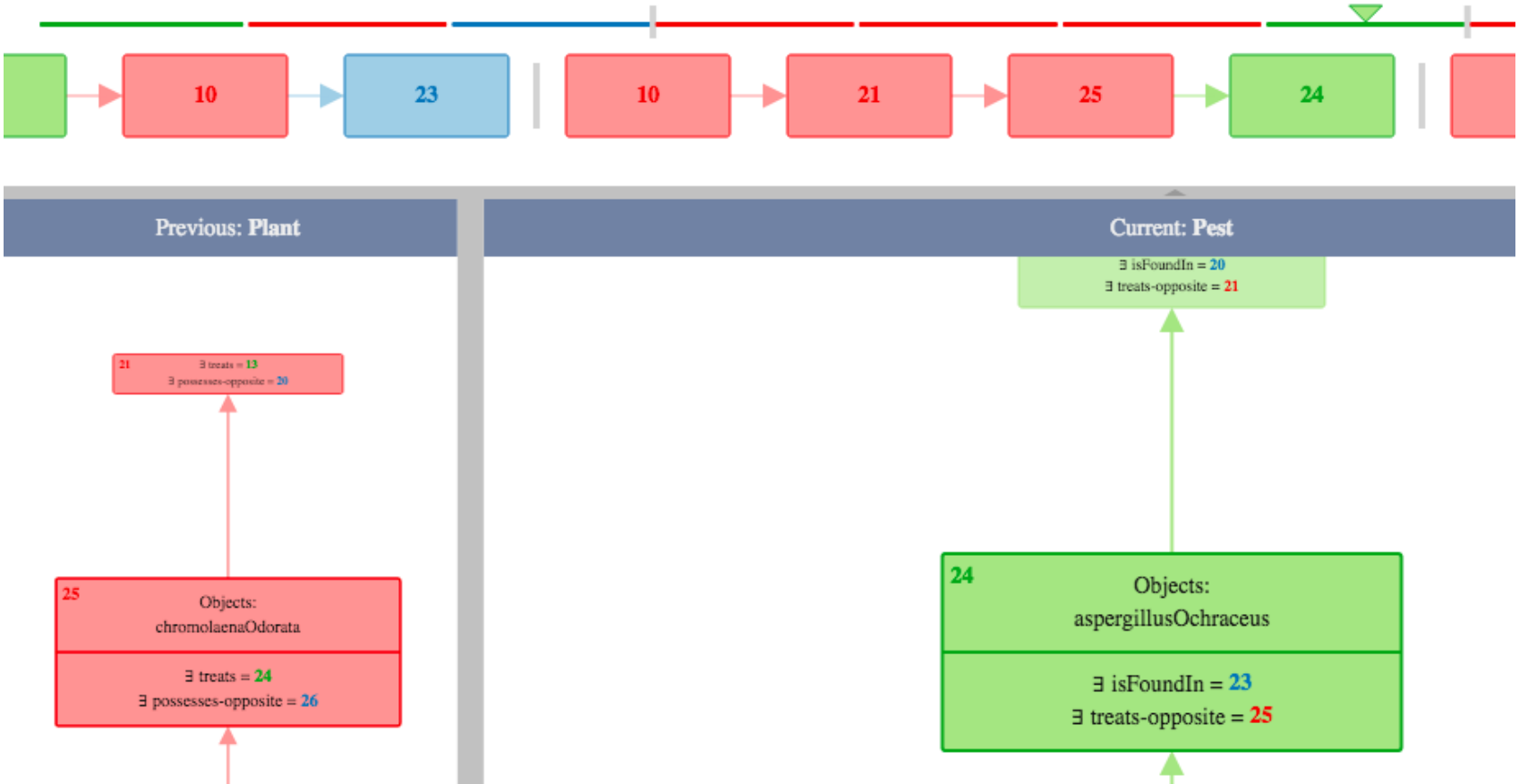
Focus on Agroecology: Knomana

Which fortunately, can be found in Benin (Concept 26)



Focus on Agroecology: Knomana

Chromolaena odorata controls Aspergillus ochraceus, which has the same genus as Aspergillus parasiticus.
This hypothesis that can be delivered to experts: i.e. conduct experiments for using Chromolaena odorata against Aspergillus parasiticus



Focus on Agroecology: Knomana

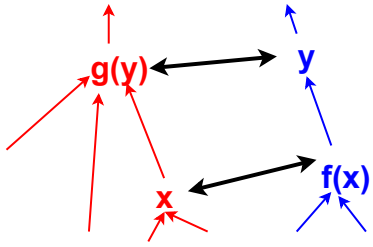
Challenges

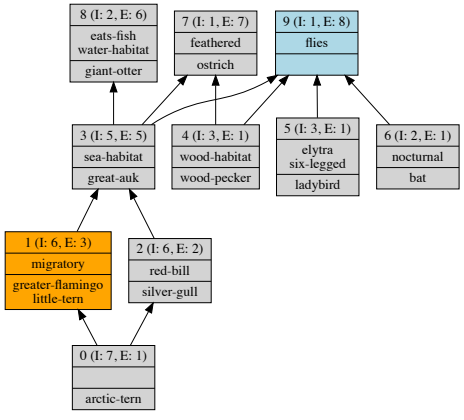
- Incomplete and imprecise data (e.g. indications like spp)
- Potential mistakes (needed cleaning)
- FCA/RCA tools still hardly used by domain experts (a data scientist is needed)

Agenda

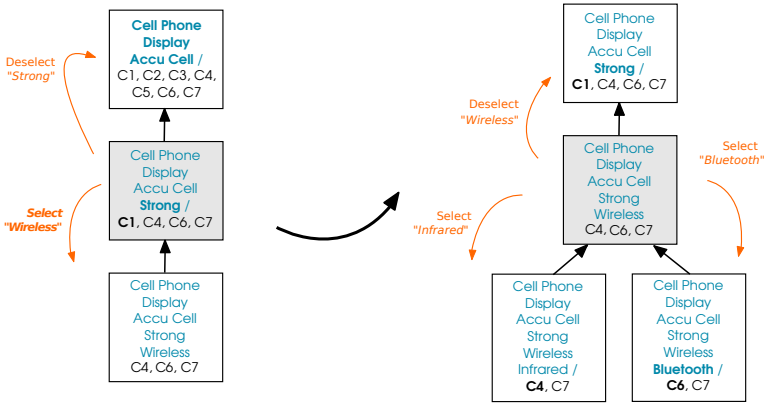
FCA: a Swiss knife for Knowledge Discovery

	flies	nocturnal	feathered	migratory	red-bill	elytra	sea-habitat	wood-habitat	six-legged	eats-fish	water-habitat
ladybird	×					×			×		
bat	×	×									
ostrich			×								
greater-flamingo	×		×	×			×			×	×
silver-gull	×		×		×		×			×	×
little-tern	×		×	×			×			×	×
great-auk	×		×				×			×	×
wood-pecker	×		×					×			
giant-otter										×	×
arctic-tern	×		×	×	×		×			×	×

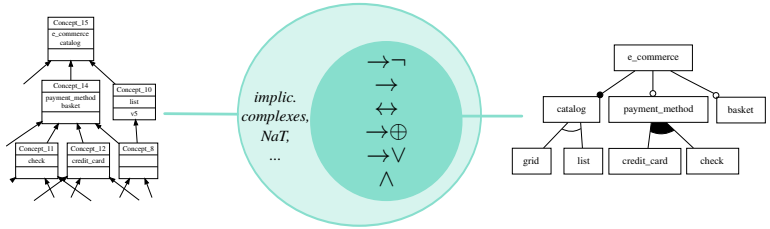




six-legged => flies,elytra
red-bill => flies,feathered,sea-habitat,eats-fish,water-habitat
feathered,eats-fish,water-habitat => flies,sea-habitat
eats-fish => water-habitat



Credits: J. Galasso-Carbonnel

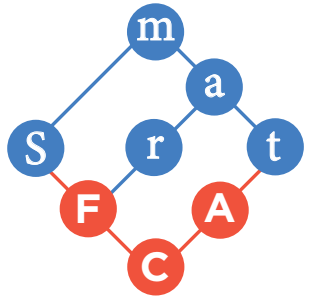


Credits: J. Galasso-Carbonnel

What's next?

- Theory: e.g. connecting the different FCA trends, connecting FCA to other KD approaches
- Algorithms: e.g. local/incremental, parallel
- Methodology: e.g. User interaction, Detecting and correcting anomalies, Classifying rules, Generalizing from applications
- Address Complex data / Big data challenges
- Using FCA/RCA as data complexity measuring framework
- Hybrid IA/KD systems; Explainable AI
- Integration in data science workflows (e.g. in Orange or Scikit-learn with subsymbolic ML, in Jupyter Notebooks)

Thank you!



Smart **FCA**

Partly supported by the ANR SmartFCA project
Grant ANR-21-CE23-0023 of the French National Research Agency
anr

Many thanks to many collaborators, and ...
Special credits to Amedeo Napoli for sharing ideas and references