A gentle introduction to Relational Concept Analysis, Tutorial ICFCA 2011

Marianne Huchard, Amedeo Napoli, Amine Mohamed Rouane Hacene, Petko Valtchev

Relational Concept Analysis
A gentle introduction

Marianne Huchard, Amedeo Napoli,
Mohamed Rouane Hacène, Petko Valtchev

August 21, 2011

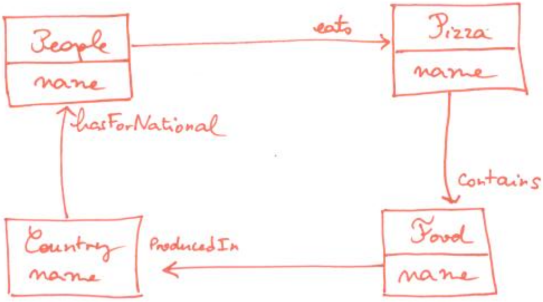
Relational Concept Analysis (RCA)

- Extend the purpose of FCA for taking into account relations between objects
- The RCA process relies on the following main points:
  - a relational model based on the entity-relationship model
  - a conceptual scaling process allowing to represent relations between objects as relational attributes
  - an iterative process for designing a concept lattice where concept intents include non-relational and relational attributes.
- The RCA process provides relational structures that can be represented as ontology concepts within a knowledge representation formalism such as description logics (DLs).

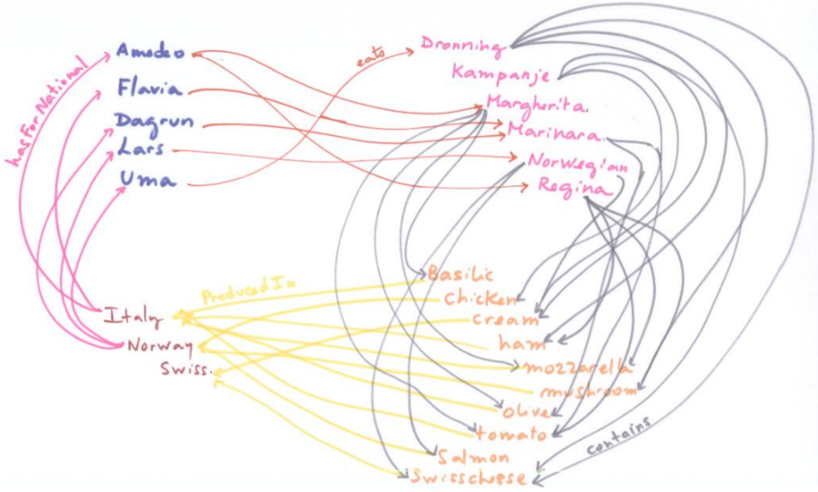
Relational Concept Analysis (RCA)

A relational model based on the entity-relationship model ...

Pizza story

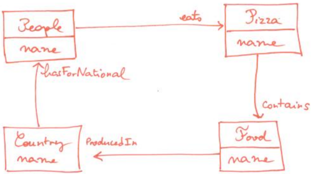


Objects and links



Relational Concept Analysis (RCA)

Pizza story



Pizza data

- four object/attribute contexts
  - $K_{People} \subset$ People $\times$ people names
  - $K_{Pizza} \subset$ Pizza $\times$ pizza names
  - $K_{Food} \subset$ Food item $\times$ food names
  - $K_{Country} \subset$ Country $\times$ country names
- four object/object contexts
  - `eats` $\subset$ People $\times$ Pizza
  - `contains` $\subset$ Pizzas $\times$ Food item
  - `producedIn` $\subset$ Food item $\times$ Country
  - `hasForNational` $\subset$ Country $\times$ People

K_{People}

	Amedeo	Flavia	Dagrun	Lars	Uma
Amedeo	×				
Flavia		×			
Dagrun			×		
Lars				×	
Uma					×

K_{Pizza}

	Dronning	Kampanje	Margherita	Marina	Norwegian	Regina
Dronning	×					
Kampanje		×				
Margherita			×			
Marina				×		
Norwegian					×	
Regina						×

K_{Food}

	basilic	chicken	cream	ham	mozzarella	mushroom	olive	tomato	salmon	swisscheese
basilic	×									
chicken		×								
cream			×							
ham				×						
mozzarella					×					
mushroom						×				
olive							×			
tomato								×		
salmon									×	
swisscheese										×

$K_{Country}$

	Italy	Norway	Switzerland
Italy	$\times$		
Norway		$\times$	
Switzerland			$\times$

R_{eats}

	Dronning	Kampanje	Margherita	Marina	Norwegian	Regina
Amedeo			$\times$			$\times$
Flavia				$\times$		
Dagrun				$\times$		$\times$
Lars					$\times$	
Uma	$\times$				$\times$	

$R_{contains}$

	basilic	chicken	cream	ham	mozzarella	mushroom	olive	tomato	salmon	swisscheese
Dronning			×	×		×				×
Kampanje		×	×							×
Margherita	×				×		×	×		
Marina							×	×		
Norwegian			×						×	×
Regina				×	×	×		×		

$R_{producedIn}$

	Italy	Norway	Switzerland
basilic	$\times$		
chicken		$\times$	
cream			$\times$
ham	$\times$		
mozzarella	$\times$		
mushroom		$\times$	
olive	$\times$		
tomato	$\times$		
salmon		$\times$	
swisscheese			$\times$

$R_{hasForNational}$

	Amedeo	Flavia	Dagrun	Lars	Uma
Italy	$\times$	$\times$			
Norway			$\times$	$\times$	$\times$
Switzerland					

Relational Context Family (RCF)

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

- K is a set of formal contexts $K_i = (O_i, A_i, I_i)$
- R is a set of relational 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 context*, K_l is the *target context*.
  - we may have $K_k = K_l$.

Pizza RCF

$K = K_{People}, K_{Pizza}, K_{Food}, K_{Country}$

$R = R_{eats}, R_{contains}, R_{producedIn}, R_{hasForNational}$

A simple approach

Concatenate relational contexts to their source formal context

E.g. $K_{People} + R_{eats}$

	Amedeo	Flavia	Dagrun	Lars	Uma	eats:Dronning	eats:Kampanje	eats:Margherita	eats:Marinara	eats:Norwegian	eats:Regina
Amedeo	×							×			×
Flavia		×							×		
Dagrun			×						×		×
Lars				×						×	
Uma					×	×				×	

A simple approach

Discovered concepts

- People who eat the marinara pizza (Flavia, Dagrun)
- People who eat the norwegian pizza (Lars, Uma)

	Amedeo	Flavia	Dagrun	Lars	Uma	eats:Dronning	eats:Kampanje	eats:Margherita	eats:Marinara	eats:Norwegian	eats:Regina
Amedeo	×							×			×
Flavia		×							×		
Dagrun			×						×		×
Lars				×						×	
Uma					×	×				×	

A simple approach (limits)

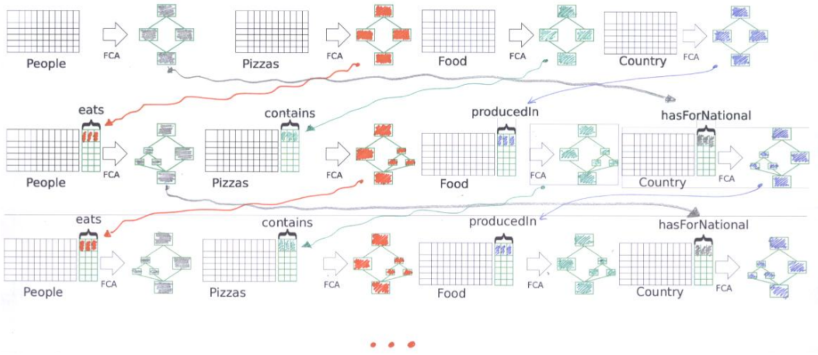
Undiscovered concepts

- People who eat the red-sauce pizzas (Amedeo, Flavia, Dagrun)
- People who eat pizzas produced with food from Norway (Amedeo, Dagrun, Lars, Uma)

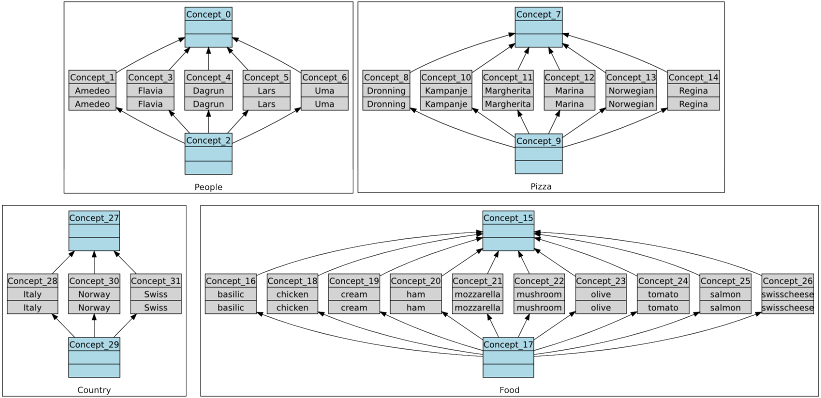
	Amedeo	Flavia	Dagrun	Lars	Uma	eats:Dronning	eats:Kampanje	eats:Margherita	eats:Marinara	eats:Norwegian	eats:Regina
Amedeo	×							×			×
Flavia		×							×		
Dagrun			×						×		×
Lars				×						×	
Uma					×	×				×	

An iterative approach (RCA)

Learned concepts are used in a next step to learn more



RCA - Step 0 - Initial Lattices



Scaling relations

Integrating concepts in the relational contexts

Amedeo eats Margherita ; Margherita $\in$ extent(Concept_11)
$\rightarrow \exists p \in$Concept_11, s.t. Amedeo eats p
$\rightarrow$ (Amedeo, eats:Concept_11)
$\rightarrow$ (Amedeo, Concept_11) belongs to the existentially scaled relation eat*, (Amedeo,$\exists$ *eat* :Concept_11) stands

	Amedeo	Flavia	Dagrun	Lars	Uma	eats : Concept_7	eats : Concept_8	eats : Concept_11	eats : Concept_12	eats : Concept_13	eats : Concept_14
Amedeo	X					X		X			X
Flavia		X				X			X		
Dagrun			X			X			X		X
Lars				X		X				X	
Uma					X	X	X			X	

RCA - Scaling relations

For each relational context $R_j = (O_k, O_l, I_j)$, a scaled context $R_j^* = (O_k, A, I_j)$ is created.

- A is the concept set of a lattice built on objects of O_l, denoted by $\mathcal{L}_l^n$
- I_j contains (o, a) iff $S(R_j(o), Extent(a))$ is true.

S is a *scaling* operator, the most used are:

- $S_\exists(R(o), Extent(a))$ is true iff $\exists x \in R(o), x \in Extent(a)$.
- $S_{\forall\exists}(R(o), Extent(a))$ is true iff $\forall x \in R(o), x \in Extent(a) \wedge \exists x \in R(o), x \in Extent(a)$

Scaling operators

Operator	Attribute form	Condition
Universal (wide)	$\forall\ r : c$	$r(o) \subseteq Ext(c)$
Includes	$\supseteq\ r : c$	$r(o) \supseteq Ext(c)$
Existential	$\exists\ r : c$	$r(o) \cap Ext(c) \neq \emptyset$
Universal strict	$\forall\exists\ r : c$	$r(o) \subseteq Ext(c)$ *and* $r(o) \neq \emptyset$
Qualified cardinality restriction	$\geq\ n\ r : c$	$r(o) \subseteq Ext(c)$ *and* $\lvert r(o) \rvert \geq\ n$
Cardinality restriction	$\geq\ n\ r : \top_{\mathcal{L}}$	$\lvert r(o) \rvert \geq\ n$

Relational scaling

Some properties of relational scaling:

- The homogeneity of concept descriptions is kept: all attributes are considered as binary (even relational attributes).
- Standard algorithms for building concept lattices can be directly reused.

Existential scaling of contains - step 1

	contains : Concept_15	contains : Concept_16	contains : Concept_18	contains : Concept_19	contains : Concept_20	contains : Concept_21	contains : Concept_22	contains : Concept_23	contains : Concept_24	contains : Concept_25	contains : Concept_26
Dronning	x			x	x		x				x
Kampanje	x		x	x							x
Margherita	x	x				x		x	x		
Marina	x							x	x		
Norwegian	x			x						x	x
Regina	x				x	x	x		x		

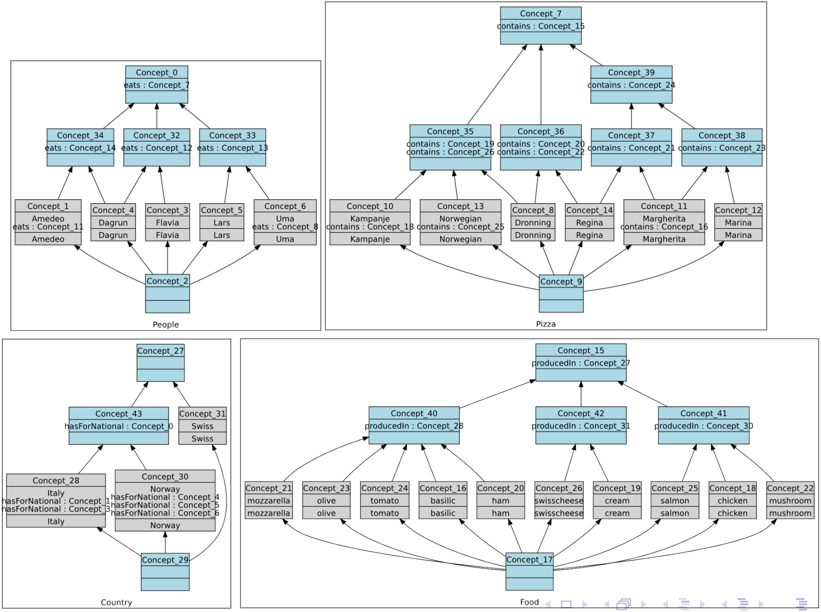
Existential scaling of producedIn - step 1

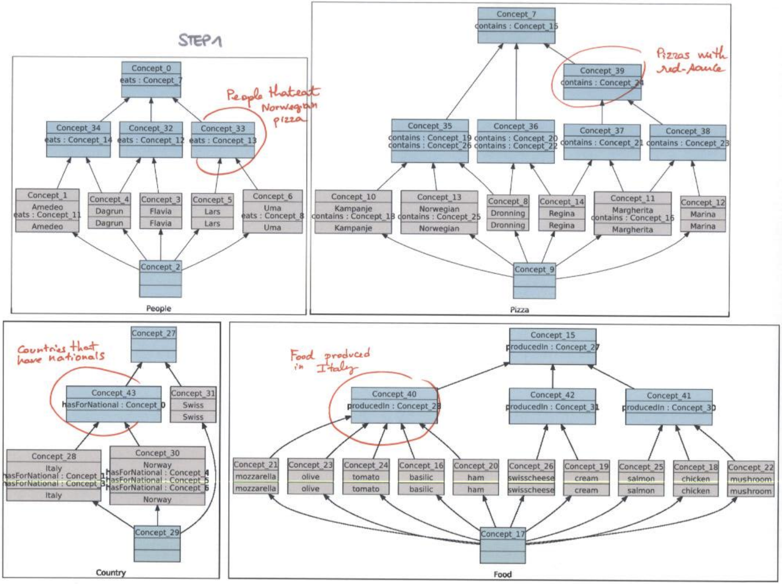
	producedIn : Concept_27	producedIn : Concept_28	producedIn : Concept_30	producedIn : Concept_31
basilic	x	x		
chicken	x		x	
cream	x			x
ham	x	x		
mozzarella	x	x		
mushroom	x		x	
olive	x	x		
tomato	x	x		
salmon	x		x	
swisscheese	x			x

Existential scaling of hasForNational - step 1

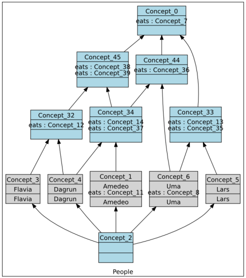
	hasForNational : Concept_0	hasForNational : Concept_1	hasForNational : Concept_3	hasForNational : Concept_4	hasForNational : Concept_5	hasForNational : Concept_6
Italy	X	X	X			
Norway	X			X	X	X
Swiss						

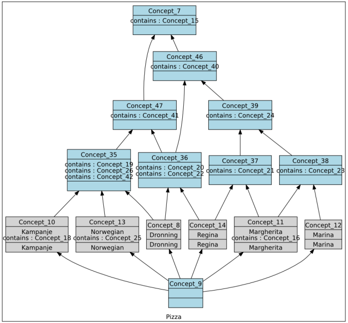
RCA - Lattices at step 1

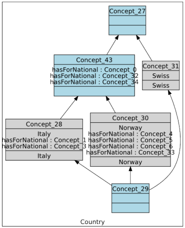


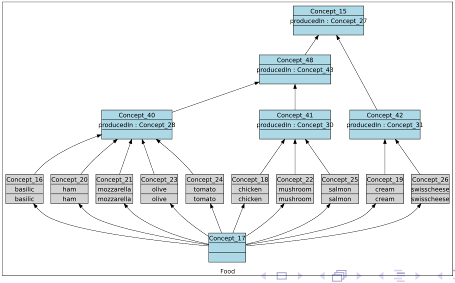

STEP 1
Concept_0
eats : Concept_7
People that eat Norwegian pizza
Concept_34
eats : Concept_14
Concept_32
eats : Concept_12
Concept_33
eats : Concept_13
Concept_1
Amedeo
eats : Concept_11
Amedeo
Concept_4
Dagrun
Dagrun
Concept_3
Flavia
Flavia
Concept_5
Lars
Lars
Concept_6
Uma
eats : Concept_8
Uma
Concept_2
People
Concept_7
contains : Concept_15
Pizzas with red-sauce
Concept_39
contains : Concept_24
Concept_35
contains : Concept_19
contains : Concept_26
Concept_36
contains : Concept_20
contains : Concept_22
Concept_37
contains : Concept_21
Concept_38
contains : Concept_23
Concept_10
Kampanje
contains : Concept_18
Kampanje
Concept_13
Norwegian
contains : Concept_25
Norwegian
Concept_8
Dronning
Dronning
Concept_14
Regina
Regina
Concept_11
Margherita
contains : Concept_16
Margherita
Concept_12
Marina
Marina
Concept_9
Pizza
Concept_27
Countries that have nationals
Concept_43
hasForNational : Concept_0
Concept_31
Swiss
Swiss
Concept_28
Italy
hasForNational : Concept_1
hasForNational : Concept_3
Italy
Concept_30
Norway
hasForNational : Concept_4
hasForNational : Concept_5
hasForNational : Concept_6
Norway
Concept_29
Country
Concept_15
producedIn : Concept_27
Food produced in Italy
Concept_40
producedIn : Concept_28
Concept_42
producedIn : Concept_31
Concept_41
producedIn : Concept_30
Concept_21
mozzarella
mozzarella
Concept_23
olive
olive
Concept_24
tomato
tomato
Concept_16
basilic
basilic
Concept_20
ham
ham
Concept_26
swisscheese
swisscheese
Concept_19
cream
cream
Concept_25
salmon
salmon
Concept_18
chicken
chicken
Concept_22
mushroom
mushroom
Concept_17
Food

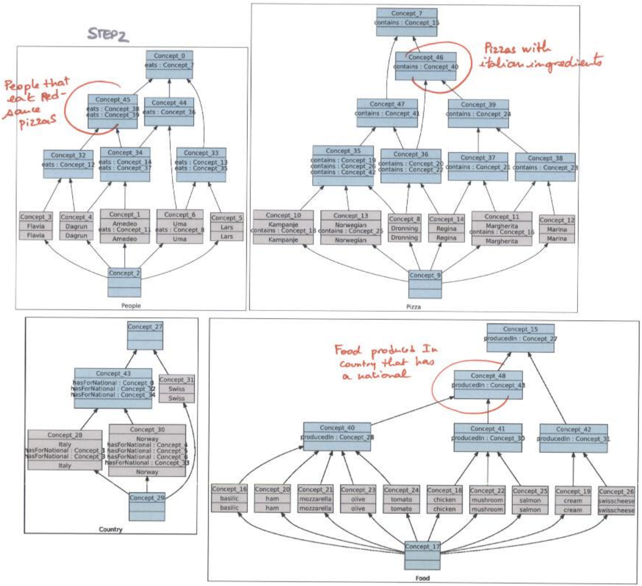
RCA - Lattices at step 2



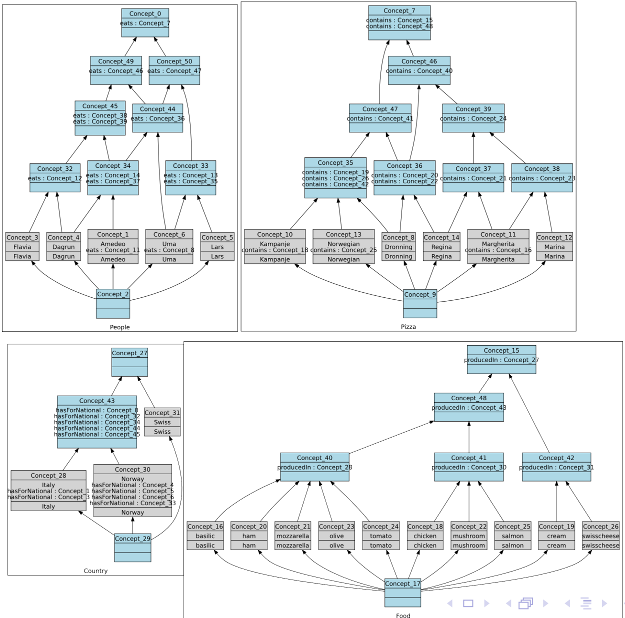


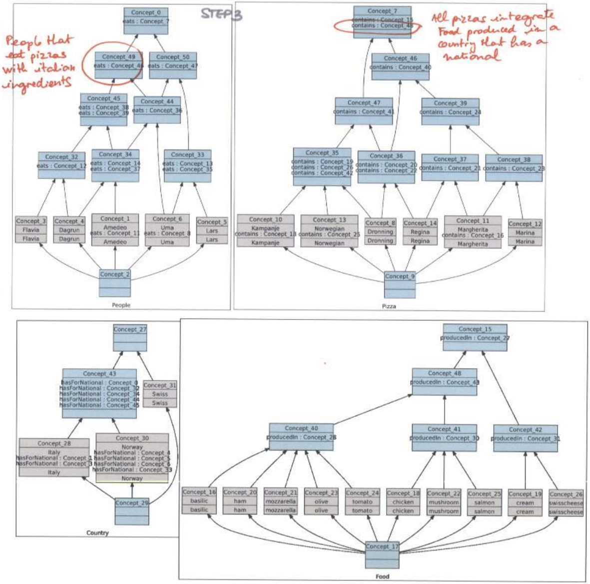




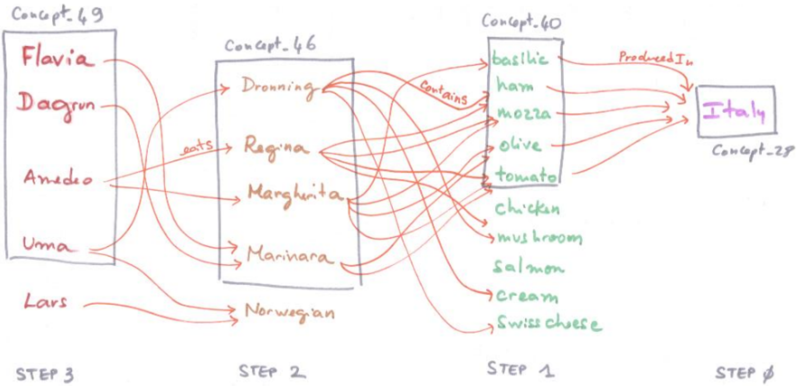

STEP2
People that eat Red-sauce pizzas
Pizzas with italian ingredients
Food produced In country that has a national
People
Pizza
Country
Food

RCA - Lattices at step 3




STEP 3
People that eat pizzas with italian ingredients
All pizzas integrate Food produced in a country that has a national
People
Pizza
Country
Food

An excerpt of the iteration



The RCA schema

Input

$RCF = (K, R)$: n formal contexts, several relational contexts

Initialization step

build, for i in $1..n$, $\mathbf{L}^0[i]$ the concept lattice of the context $\mathcal{K}_i$

Step p

▷ concatenate $\mathcal{K}_i$ with the relational contexts with source $\mathcal{K}_i$ scaled using the lattices of step $p-1$ and a given operator
▷ update lattices of step $p-1$ to build, for i in $1..n$, the lattice $\mathbf{L}^p[i]$ for the context $\mathcal{K}_i$ concatenated as explained previously

Output (fix point)

A concept lattice family is obtained when no new concepts are added

The RCA algorithm

proc Multi-Fca(In: $(\mathbf{K}, \mathbf{R})$ a RCF,
Out: $\mathbf{L}$ array $[1..n]$ of lattices)
$p \leftarrow 0$; halt $\leftarrow$ false
for i from 1 to n do
  $\mathbf{L}^0[i] \leftarrow$ Build-Lattice($\mathcal{K}_i^0$)
while not halt do
  $p++$
  for i from 1 to n do
    $\mathcal{K}_i^p \leftarrow$ Extend-Rel($\mathcal{K}_i^{p-1}$, $\mathbf{L}^{p-1}$)
    $\mathbf{L}^p[i] \leftarrow$ Update-Lattice($\mathcal{K}_i^p$, $\mathbf{L}^{p-1}[i]$)
  halt $\leftarrow \bigwedge_{i=1,n}$ Isomorphic($\mathbf{L}^p[i], \mathbf{L}^{p-1}[i]$)

A general analysis plan

Relation	Step 1	Step 2	Step 3	...
eats	$\forall\ r : c$	$\exists\ r : c$	$\exists\ r : c$	$\exists\ r : c$
contains	$\geq\ n\ r : c$	$\forall\ r : c$	$\exists\ r : c$	$\exists\ r : c$
producedIn	$\geq\ n\ r : c$	$\forall\ r : c$	$\exists\ r : c$	$\exists\ r : c$
hasForNational	$\exists\ r : c$	$\forall\ r : c$	$\exists\ r : c$	$\exists\ r : c$
...	...	...	...	

To discover for example:

- People that eat at least a pizza containing only food produced in Norway
- Countries where at least one national eats at least one pizza that contains at least one food produced in Italy

Note: at one step, several scaling operators can be applied to the same relational context (giving several scaled contexts based on a same relational context)

RCA - Arguments for convergence (finite object / attribute sets)

- the number of objects (lines) in extended contexts doesn't change, this limits the concepts number of every lattice $\mathcal{L}_i$ to be $2^{|O_i|}$
- the number of columns cannot increase indefinitely since new attributes are $r : c$, where r is a relation, for example with $target(r) = O_j$ and c is the concept of a lattice $\mathcal{L}_j$ built on O_j

A synthesis on RCA

- an iterative method to produce abstractions
- variations on scaling
- object-attribute concept posets can be built instead of lattices to limit the complexity

Tools

- Galicia: http://galicia.sourceforge.net/
- eRCA: http://code.google.com/p/erca/

Applications

- UML class diagram refactoring
  * Michel Dao, Marianne Huchard, Mohamed Rouane Hacene, Cyril Roume, Petko Valtchev: Improving Generalization Level in UML Models Iterative Cross Generalization in Practice. ICCS 2004: 346-360
  * Gabriela Arévalo, Jean-Rémy Falleri, Marianne Huchard, Clémentine Nebut: Building Abstractions in Class Models: Formal Concept Analysis in a Model-Driven Approach. MoDELS 2006: 513-527
- UML Use case diagram refactoring
  * Xavier Dolques, Marianne Huchard, Clémentine Nebut, and Philippe Reitz. Fixing generalization defects in UML use case diagrams. CLA 2010: 247-258
- Blob design defect correction
  * Naouel Moha, Amine Rouane Hacene, Petko Valtchev, Yann-Gaël Guéhéneuc: Refactorings of Design Defects Using Relational Concept Analysis. ICFCA 2008: 289-304
- Extracting architectures in object-oriented software
  * Alae-Eddine El Hamdouni, Abdelhak Seriai, Marianne Huchard Component-based Architecture Recovery from Object-Oriented Systems via Relational Concept Analysis. CLA 2010: 259-270

Applications

- Learning model Transformation patterns in MDE
  * Xavier Dolques, Marianne Huchard, and Clémentine Nebut. From transformation traces to transformation rules: Assisting model driven engineering approach with formal concept analysis. In Supplementary Proc. of ICCS 2009:15-29.
- Classification of web services
  * Zeina Azmeh, Maha Driss, Fady Hamoui, Marianne Huchard, Naouel Moha, Chouki Tibermacine, Selection of Composable Web Services Driven by User Requirements. To appear in the Application and Experience Track of ICWS 2011
- Ontology construction
  * Rokia Bendaoud, Mohamed Rouane Hacene, Yannick Toussaint, Bertrand Delecroix, and Amedeo Napoli, Text-based ontology construction using relational concept analysis. MCETECH 2008
- Ontology pattern extraction
  * Mohamed Rouane-Hacène, Marianne Huchard, Amedeo Napoli, Petko Valtchev. Using Formal Concept Analysis for discovering knowledge patterns. CLA 2010: 223-234
- Ontology restructuring
  * Mohamed Rouane-Hacene, Roger Nkambou and Petko Valtchev. Supporting ontology design through large-scale FCA-based ontology restructuring, to appearin Proc. of the ICCS 2011.

Related work

- Priss, U.: Classification of meronymy by methods of relational concept analysis. In: Online Proceedings of the 1996 Midwest Artificial Intelligence Conf., Bloomington, Indiana. (1996)
- Wille, R.: Conceptual structures of multicontexts. In: Conceptual Structures: Knowledge Representation as Interlingua, Springer (1996) 23-39
- Liquiere, M., Sallantin, J. : Structural Machine Learning with Galois Lattice and Graphs. ICML 1998: 305-313 Prediger, S., Wille, R.:The lattice of concept graphs of a relationally scaled context. In: Proc. of the 7th Intl. Conf. on Conceptual Structures (ICCS'99), Springer (1999) 401-414
- Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base, Proc. of the 9th Intl. Conf. on Conceptual Structures (ICCS'01), Stanford, CA. Volume 2120 of LNCS., Springer (2001) 129-142
- Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In: ICCS 2005. Volume 3596 of LNCS., Springer (2005) 166-180

References

- Rouane-Hacene M. , Huchard M., Napoli A., Valtchev P. A proposal for combining Formal Concept Analysis and description Logics for mining relational data, in Proceedings of ICFCA-2007, LNAI 4390, Springer, pages 51-65, 2007.
- Huchard, M., Hacene M. R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Ann. Math. Artif. Intell. 49(1-4): 39-76 (2007)