

Contribution to the Classification of Web of Data based on Formal Concept Analysis

Justine Reynaud, Yannick Toussaint and Amedeo Napoli

INRIA Nancy Grand Est. 54600 Villers-les-Nancy, France
Université de Lorraine. LORIA UMR 7503, Vandœuvre-les-Nancy, France

August 30, 2016

- 1 Introduction
- 2 Web of Data
 - Definitions
 - Toy example
- 3 FCA extensions
 - Background knowledge
 - Pattern structures
 - Related works
- 4 Our approach
 - Definition
 - Example
- 5 Conclusion

Considering the Web of Data implies dealing with complex data

Considering the Web of Data implies dealing with complex data

- Working with sets of **triples**

Considering the Web of Data implies dealing with complex data

- Working with sets of **triples**
- Taking into account some **background knowledge**

Considering the Web of Data implies dealing with complex data

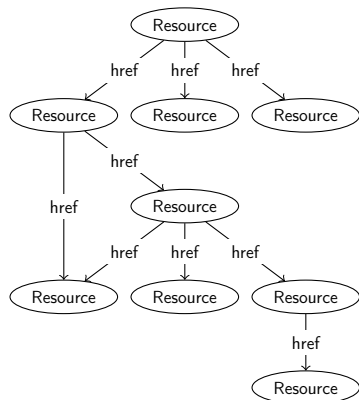
- Working with sets of **triples**
- Taking into account some **background knowledge**

Based on the work of [Alam et al., 2015] and extending this approach.

Contents

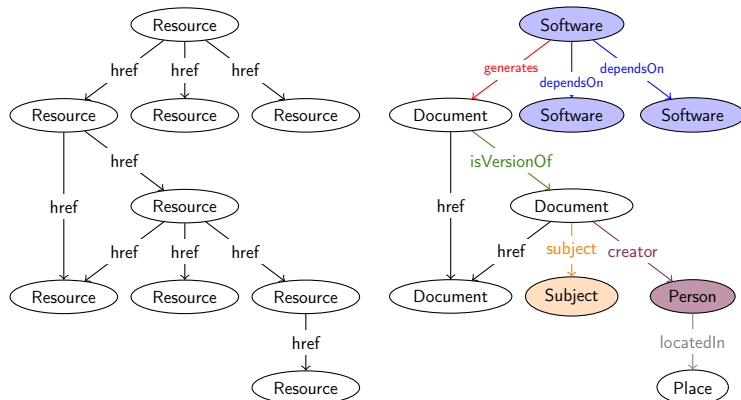
- 1 Introduction
- 2 Web of Data
 - Definitions
 - Toy example
- 3 FCA extensions
 - Background knowledge
 - Pattern structures
 - Related works
- 4 Our approach
 - Definition
 - Example
- 5 Conclusion

What is Web of Data ?



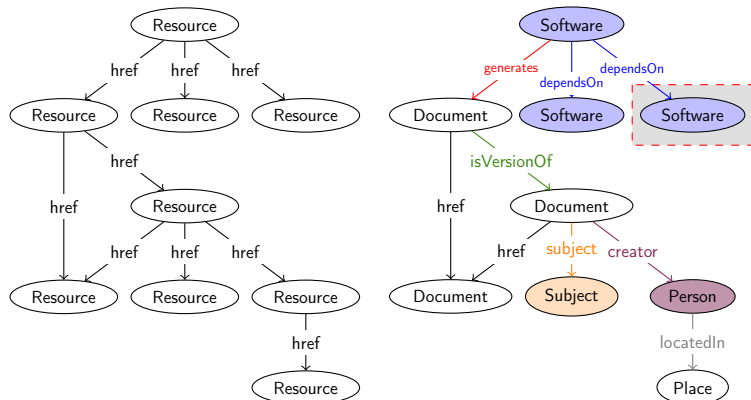
Source: <https://www.w3.org/2001/12/semweb-fin/w3csw>

What is Web of Data ?



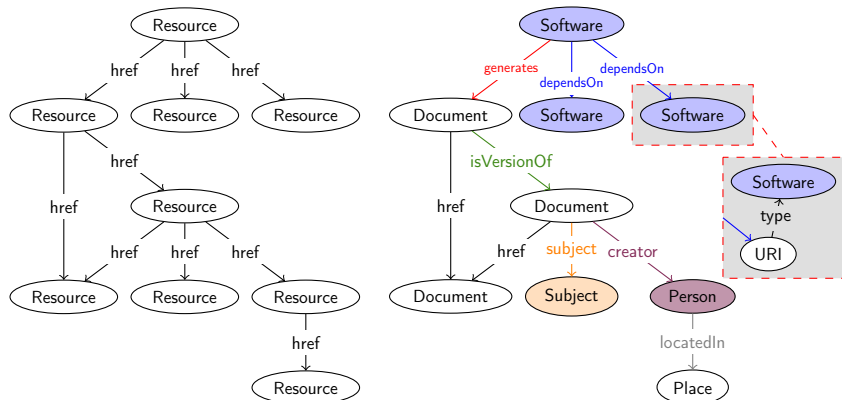
Source: <https://www.w3.org/2001/12/semweb-fin/w3csw>

What is Web of Data ?



Source: <https://www.w3.org/2001/12/semweb-fin/w3csw>

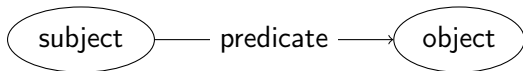
What is Web of Data ?



Source: <https://www.w3.org/2001/12/semweb-fin/w3csw>

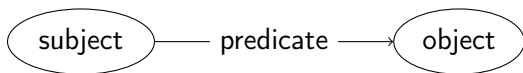
Structuring the web of data

Graph of the linked open data is represented with triples



Structuring the web of data

Graph of the linked open data is represented with triples



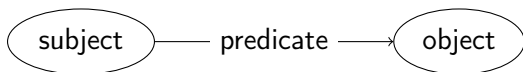
RDF Resources and their relations

Obama hasBirthPlace *Honolulu*

Honolulu rdf:type Capital

Structuring the web of data

Graph of the linked open data is represented with triples



RDF Resources and their relations

Obama hasBirthPlace *Honolulu*

Honolulu rdf:type Capital

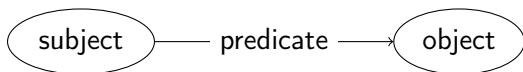
RDFS Structure of resources

Capital rdfs:subClassOf City

isCapitalOf rdfs:subPropertyOf isCityOf

Structuring the web of data

Graph of the linked open data is represented with triples



RDF Resources and their relations

Obama hasBirthPlace *Honolulu*

Honolulu rdf:type Capital

RDFS Structure of resources

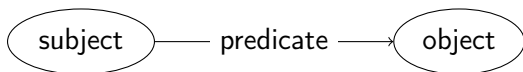
Capital rdfs:subClassOf City

isCapitalOf rdfs:subPropertyOf isCityOf

Properties: express binary relationships between any two entities

Structuring the web of data

Graph of the linked open data is represented with triples



RDF Resources and their relations

Obama hasBirthPlace *Honolulu*

Honolulu rdf:type Capital

RDFS Structure of resources

Capital rdfs:subClassOf City

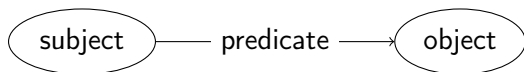
isCapitalOf rdfs:subPropertyOf isCityOf

Properties: express binary relationships between any two entities

Classes: represent sets of entities

Structuring the web of data

Graph of the linked open data is represented with triples



RDF Resources and their relations

Obama hasBirthPlace *Honolulu*

Honolulu rdf:type Capital

RDFS Structure of resources

Capital rdfs:subClassOf City

isCapitalOf rdfs:subPropertyOf isCityOf

Properties: express binary relationships between any two entities

Classes: represent sets of entities

Instances: entities that belong to a class

- *Instance* : the instance itself
- Instance : the class of the instance

Toy example

Inspired by [Ferré, 2015]

JR, YT, AN (INRIA / LORIA)

Classifying WOD with FCA



August 30, 2016

7 / 21

Toy example

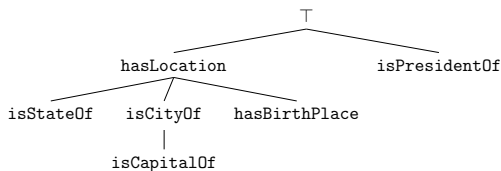
- A set of RDF triples (subject, predicate, object)

```
Obama isPresidentOf USA .  
Obama hasBirthPlace Honolulu .  
Honolulu isCapitalOf Hawaii .  
Hawaii isStateOf USA .  
Clinton isPresidentOf USA .  
Clinton hasBirthPlace Hope .  
Hope isCityOf Arkansas .  
Arkansas isStateOf USA .
```

Toy example

- A set of RDF triples (subject, predicate, object)
- A hierarchy on properties (subP)

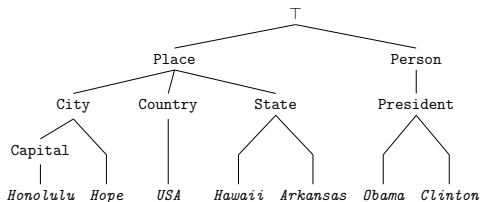
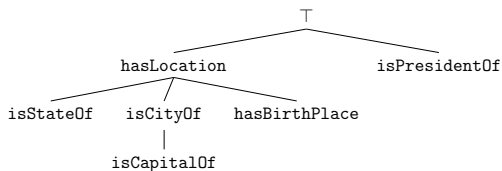
Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Honolulu isCapitalOf *Hawaii* .
Hawaii isStateOf *USA* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .
Hope isCityOf *Arkansas* .
Arkansas isStateOf *USA* .



Toy example

- A set of RDF triples (subject, predicate, object)
- A hierarchy on properties (subP)
- A hierarchy on classes (subC)

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Honolulu isCapitalOf *Hawaii* .
Hawaii isStateOf *USA* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .
Hope isCityOf *Arkansas* .
Arkansas isStateOf *USA* .



Contents

- 1 Introduction
- 2 Web of Data
 - Definitions
 - Toy example
- 3 FCA extensions
 - Background knowledge
 - Pattern structures
 - Related works
- 4 Our approach
 - Definition
 - Example
- 5 Conclusion

[Carpineto and Romano, 2004]

Taking into account background knowledge.

- Objects (documents)
- Attributes (keywords)
- Background knowledge (thesaurus)

Taking into account background knowledge.

- Objects (documents)
- Attributes (keywords)
- Background knowledge (thesaurus)

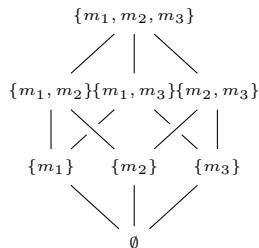
“The intersection of the intent of two concepts (X_1, Y_1) and (X_2, Y_2) is obtained by finding, for each pair (m_1, m_2) , $m_1 \in Y_1, m_2 \in Y_2$, the most specific attributes in M^* that are most general than m_1 and m_2 , and then retaining only the most specific elements of the set of attributes generated this way ”

Pattern structures: Definition [Ganter and Kuznetsov, 2001]

Pattern structures: Definition [Ganter and Kuznetsov, 2001]

- From a (G, M, I) context...

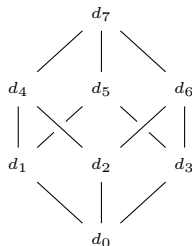
| | m_1 | m_2 | m_3 |
|-------|-------|-------|-------|
| g_1 | × | × | × |
| g_2 | × | × | |
| g_3 | | × | × |
| g_4 | × | | |



Pattern structures: Definition [Ganter and Kuznetsov, 2001]

- From a (G, M, I) context...
- ... to a $(G, (\mathcal{P}(M), \cap), \delta)$ pattern structure

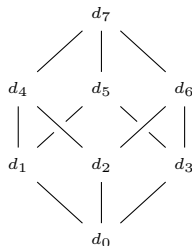
| | m_1 | m_2 | m_3 | | description | name |
|-------|-------|-------|-------|-------|---------------------|-------|
| g_1 | × | × | × | g_1 | $\{m_1, m_2, m_3\}$ | d_7 |
| g_2 | × | × | | g_2 | $\{m_1, m_2\}$ | d_4 |
| g_3 | | × | × | g_3 | $\{m_2, m_3\}$ | d_5 |
| g_4 | × | | | g_4 | $\{m_1\}$ | d_1 |



Pattern structures: Definition [Ganter and Kuznetsov, 2001]

- From a (G, M, I) context...
- ... to a $(G, (\mathcal{P}(M), \cap), \delta)$ pattern structure

| | m_1 | m_2 | m_3 | | description | name |
|-------|-------|-------|-------|-------|---------------------|-------|
| g_1 | × | × | × | g_1 | $\{m_1, m_2, m_3\}$ | d_7 |
| g_2 | × | × | | g_2 | $\{m_1, m_2\}$ | d_4 |
| g_3 | | × | × | g_3 | $\{m_2, m_3\}$ | d_5 |
| g_4 | × | | | g_4 | $\{m_1\}$ | d_1 |

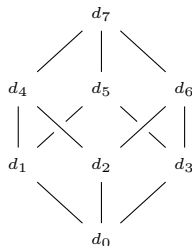


- Each object is mapped to its description : $\delta(g_2) = d_4$, $\delta(g_3) = d_5$

Pattern structures: Definition [Ganter and Kuznetsov, 2001]

- From a (G, M, I) context...
- ... to a $(G, (\mathcal{P}(M), \cap), \delta)$ pattern structure

| | m_1 | m_2 | m_3 | | description | name |
|-------|-------|-------|-------|-------|---------------------|-------|
| g_1 | × | × | × | g_1 | $\{m_1, m_2, m_3\}$ | d_7 |
| g_2 | × | × | | g_2 | $\{m_1, m_2\}$ | d_4 |
| g_3 | | × | × | g_3 | $\{m_2, m_3\}$ | d_5 |
| g_4 | × | | | g_4 | $\{m_1\}$ | d_1 |

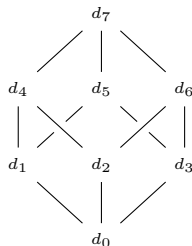


- Each object is mapped to its description : $\delta(g_2) = d_4$, $\delta(g_3) = d_5$
- Descriptions are partially ordered : $d_4 \sqsubseteq d_1$

Pattern structures: Definition [Ganter and Kuznetsov, 2001]

- From a (G, M, I) context...
- ... to a $(G, (\mathcal{P}(M), \cap), \delta)$ pattern structure

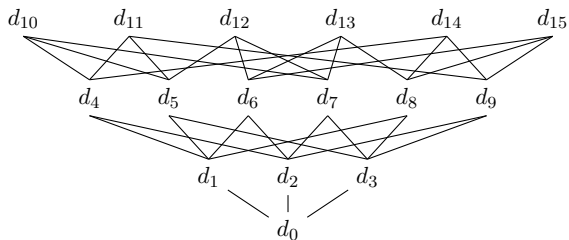
| | m_1 | m_2 | m_3 | | description | name |
|-------|-------|-------|-------|-------|---------------------|-------|
| g_1 | × | × | × | g_1 | $\{m_1, m_2, m_3\}$ | d_7 |
| g_2 | × | × | | g_2 | $\{m_1, m_2\}$ | d_4 |
| g_3 | | × | × | g_3 | $\{m_2, m_3\}$ | d_5 |
| g_4 | × | | | g_4 | $\{m_1\}$ | d_1 |



- Each object is mapped to its description : $\delta(g_2) = d_4$, $\delta(g_3) = d_5$
- Descriptions are partially ordered : $d_4 \sqsubseteq d_1$
- Similarity between objects is the part of their descriptions they share :
 $\delta(g_2) \cap \delta(g_3) = d_4 \cap d_5 = d_1$

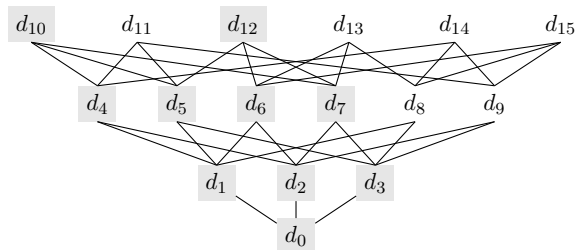
Pattern structures with sets of descriptions

Each object has a set of descriptions.



Pattern structures with sets of descriptions

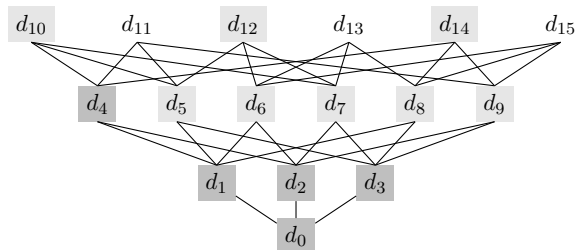
Each object has a set of descriptions.



- $\delta(g_1) = \{d_{10}, d_{12}\}$

Pattern structures with sets of descriptions

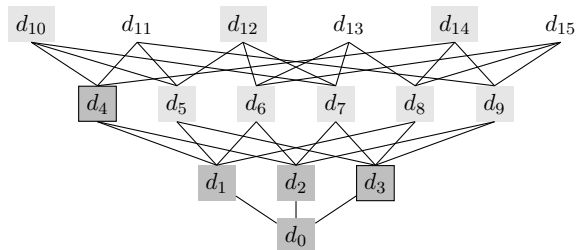
Each object has a set of descriptions.



- $\delta(g_1) = \{d_{10}, d_{12}\}$
- $\delta(g_2) = \{d_{14}\}$

Pattern structures with sets of descriptions

Each object has a set of descriptions.

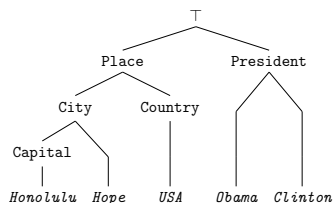


- $\delta(g_1) = \{d_{10}, d_{12}\}$
- $\delta(g_2) = \{d_{14}\}$
- $\delta(g_1) \sqcap \delta(g_2) = \{d_4, d_3\}$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .

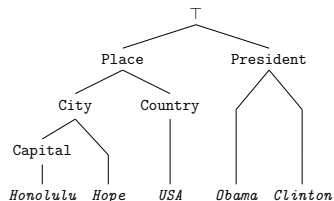


Given the toy example, we have :

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



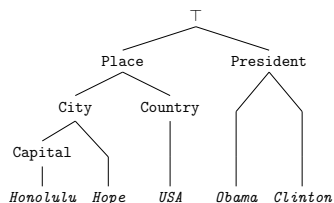
Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



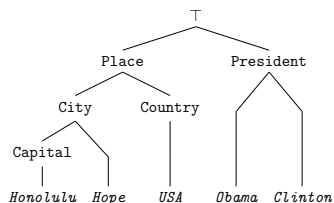
Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$
$$\delta(\text{Clinton}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Hope}\})\}$$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



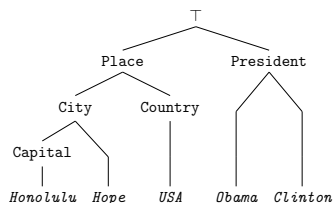
Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$
$$\delta(\text{Clinton}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Hope}\})\}$$
$$\delta(\text{Obama}) \sqcap \delta(\text{Clinton})$$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



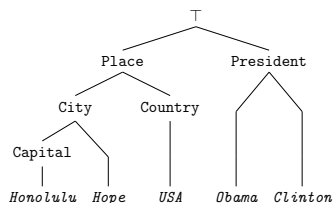
Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$
$$\delta(\text{Clinton}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Hope}\})\}$$
$$\delta(\text{Obama}) \sqcap \delta(\text{Clinton})$$
$$= \{(\text{isPresidentOf}, \{\text{lca}(\text{USA}, \text{USA})\}), (\text{hasBirthPlace}, \{\text{lca}(\text{Honolulu}, \text{Hope})\})\}$$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



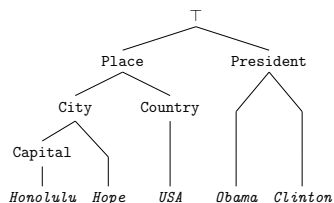
Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$
$$\delta(\text{Clinton}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Hope}\})\}$$
$$\delta(\text{Obama}) \sqcap \delta(\text{Clinton})$$
$$= \{(\text{isPresidentOf}, \{\text{lca}(\text{USA}, \text{USA})\}), (\text{hasBirthPlace}, \{\text{lca}(\text{Honolulu}, \text{Hope})\})\}$$
$$= \{(\text{isPresidentOf}, \{\text{lca}(\text{Country}, \text{Country})\}), (\text{hasBirthPlace}, \{\text{lca}(\text{Capital}, \text{City})\})\}$$

Pattern structures : Application [Alam et al., 2015]

Classifying subjects given the triples they are involved in.

Obama isPresidentOf *USA* .
Obama hasBirthPlace *Honolulu* .
Clinton isPresidentOf *USA* .
Clinton hasBirthPlace *Hope* .



Given the toy example, we have :

$$\delta(\text{Obama}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Honolulu}\})\}$$
$$\delta(\text{Clinton}) = \{(\text{isPresidentOf}, \{\text{USA}\}), (\text{hasBirthPlace}, \{\text{Hope}\})\}$$
$$\delta(\text{Obama}) \sqcap \delta(\text{Clinton})$$
$$= \{(\text{isPresidentOf}, \{\text{lca}(\text{USA}, \text{USA})\}), (\text{hasBirthPlace}, \{\text{lca}(\text{Honolulu}, \text{Hope})\})\}$$
$$= \{(\text{isPresidentOf}, \{\text{lca}(\text{Country}, \text{Country})\}), (\text{hasBirthPlace}, \{\text{lca}(\text{Capital}, \text{City})\})\}$$
$$= \{(\text{isPresidentOf}, \{\text{Country}\}), (\text{hasBirthPlace}, \{\text{City}\})\}$$

Related works

- Background knowledge [Carpineto and Romano, 2004]
- Pattern structures [Ganter and Kuznetsov, 2001]
- Application of PS to web of data [Alam et al., 2015]

Related works

- Background knowledge [Carpineto and Romano, 2004]
- Pattern structures [Ganter and Kuznetsov, 2001]
- Application of PS to web of data [Alam et al., 2015]

But also

- Triadic analysis [Lehmann and Wille, 1995]
- Graph-based approaches
 - Graph-FCA [Ferré, 2015]
 - FCA for RDF graphs [Kötters, 2015]

Contents

- 1 Introduction
- 2 Web of Data
 - Definitions
 - Toy example
- 3 FCA extensions
 - Background knowledge
 - Pattern structures
 - Related works
- 4 Our approach**
 - Definition
 - Example
- 5 Conclusion

Defining the pattern structure

Defining the pattern structure

Extent An identifier of the triple

Defining the pattern structure

Extent An identifier of the triple

Intent The triple itself where instances are replaced by their classes

Defining the pattern structure

Extent An identifier of the triple

Intent The triple itself where instances are replaced by their classes

Similarity $(s_i, p_i, o_i) \sqcap (s_j, p_j, o_j) = (lca(s_i, s_j), lca(p_i, p_j), lca(o_i, o_j))$

Defining the pattern structure

Extent An identifier of the triple

Intent The triple itself where instances are replaced by their classes

Similarity $(s_i, p_i, o_i) \sqcap (s_j, p_j, o_j) = (lca(s_i, s_j), lca(p_i, p_j), lca(o_i, o_j))$

Example

$(\text{Hope}, \text{cityOf}, \text{Arkansas}) \sqcap (\text{Honolulu}, \text{capitalOf}, \text{Hawaii}) = (\text{City}, \text{cityOf}, \text{State})$

Generalising to set of triples

With sets of triples, the similarity is computed element by element.
The result is a set of triples.

Generalising to set of triples

With sets of triples, the similarity is computed element by element.
The result is a set of triples.

Description $\Delta(\{t_1, \dots, t_n\}) = \{\delta(t_1), \dots, \delta(t_n)\}$

Generalising to set of triples

With sets of triples, the similarity is computed element by element.
The result is a set of triples.

Description $\Delta(\{t_1, \dots, t_n\}) = \{\delta(t_1), \dots, \delta(t_n)\}$

Similarity

$$\Delta(\{t_1, \dots, t_n\}) \sqcap \Delta(\{t_1, \dots, t_m\}) = \min_{\substack{i \in \{1, \dots, n\} \\ j \in \{1, \dots, m\}}} \{\delta(t_i) \sqcap \delta(t_j)\}$$

Where min is the minimal element considering the order on descriptions:

$$(s_i, p_i, o_i) \sqsubseteq (s_j, p_j, o_j) \Leftrightarrow s_j \text{ subC } s_i, p_j \text{ subP } p_i, o_j \text{ subC } o_i$$

Generalising to set of triples

With sets of triples, the similarity is computed element by element.
The result is a set of triples.

Description $\Delta(\{t_1, \dots, t_n\}) = \{\delta(t_1), \dots, \delta(t_n)\}$

Similarity

$$\Delta(\{t_1, \dots, t_n\}) \sqcap \Delta(\{t_1, \dots, t_m\}) = \min_{\substack{i \in \{1, \dots, n\} \\ j \in \{1, \dots, m\}}} \{\delta(t_i) \sqcap \delta(t_j)\}$$

Where min is the minimal element considering the order on descriptions:

$$(s_i, p_i, o_i) \sqsubseteq (s_j, p_j, o_j) \Leftrightarrow s_j \text{ subC } s_i, p_j \text{ subP } p_i, o_j \text{ subC } o_i$$

Example

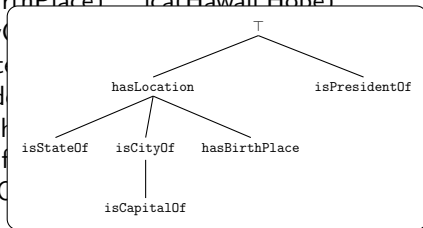
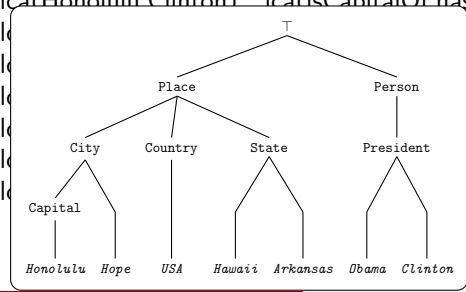
| | |
|--|--|
| <i>Obama</i> isPresidentOf <i>USA</i> . | <i>Clinton</i> isPresidentOf <i>USA</i> . |
| <i>Obama</i> hasBirthPlace <i>Honolulu</i> . | <i>Clinton</i> hasBirthPlace <i>Hope</i> . |
| <i>Honolulu</i> isCapitalOf <i>Hawaii</i> . | <i>Hope</i> isCityOf <i>Arkansas</i> . |
| <i>Hawaii</i> isStateOf <i>USA</i> . | <i>Arkansas</i> isStateOf <i>USA</i> . |

Example

| | | |
|------------------------|----------------------------------|------------------------|
| lca(Obama,Clinton) | lca(isPresidentOf,isPresidentOf) | lca(USA,USA) |
| lca(Obama,Clinton) | lca(isPresidentOf,hasBirthPlace) | lca(Honolulu,Hope) |
| lca(Obama,Hope) | lca(isPresidentOf,isCityOf) | lca(Honolulu,Arkansas) |
| lca(Obama,Arkansas) | lca(isPresidentOf,isStateOf) | lca(USA,USA) |
| lca(Obama,Clinton) | lca(hasBirthPlace,isPresidentOf) | lca(Honolulu,USA) |
| lca(Obama,Clinton) | lca(hasBirthPlace,hasBirthPlace) | lca(Honolulu,Hope) |
| lca(Obama,Hope) | lca(hasBirthPlace,isCityOf) | lca(Honolulu,Arkansas) |
| lca(Obama,Arkansas) | lca(hasBirthPlace,isStateOf) | lca(Honolulu,USA) |
| lca(Honolulu,Clinton) | lca(isCapitalOf,isPresidentOf) | lca(Hawaii,USA) |
| lca(Honolulu,Clinton) | lca(isCapitalOf,hasBirthPlace) | lca(Hawaii,Hope) |
| lca(Honolulu,Hope) | lca(isCapitalOf,isCityOf) | lca(Hawaii,Arkansas) |
| lca(Honolulu,Arkansas) | lca(isCapitalOf,isStateOf) | lca(Hawaii,USA) |
| lca(Hawaii,Clinton) | lca(isStateOf,isPresidentOf) | lca(USA,USA) |
| lca(Hawaii,Clinton) | lca(isStateOf,hasBirthPlace) | lca(USA,Hope) |
| lca(Hawaii,Hope) | lca(isStateOf,isCityOf) | lca(USA,Arkansas) |
| lca(Hawaii,Arkansas) | lca(isStateOf,isStateOf) | lca(USA,USA) |

Example

| | | |
|-------------------------------|--|--------------------------------|
| <i>lca(Obama, Clinton)</i> | <i>lca(isPresidentOf, isPresidentOf)</i> | <i>lca(USA, USA)</i> |
| <i>lca(Obama, Clinton)</i> | <i>lca(isPresidentOf, hasBirthPlace)</i> | <i>lca(Honolulu, Hope)</i> |
| <i>lca(Obama, Hope)</i> | <i>lca(isPresidentOf, isCityOf)</i> | <i>lca(Honolulu, Arkansas)</i> |
| <i>lca(Obama, Arkansas)</i> | <i>lca(isPresidentOf, isStateOf)</i> | <i>lca(USA, USA)</i> |
| <i>lca(Obama, Clinton)</i> | <i>lca(hasBirthPlace, isPresidentOf)</i> | <i>lca(Honolulu, USA)</i> |
| <i>lca(Obama, Clinton)</i> | <i>lca(hasBirthPlace, hasBirthPlace)</i> | <i>lca(Honolulu, Hope)</i> |
| <i>lca(Obama, Hope)</i> | <i>lca(hasBirthPlace, isCityOf)</i> | <i>lca(Honolulu, Arkansas)</i> |
| <i>lca(Obama, Arkansas)</i> | <i>lca(hasBirthPlace, isStateOf)</i> | <i>lca(Honolulu, USA)</i> |
| <i>lca(Honolulu, Clinton)</i> | <i>lca(isCapitalOf, isPresidentOf)</i> | <i>lca(Hawaii, USA)</i> |
| <i>lca(Honolulu, Clinton)</i> | <i>lca(isCapitalOf, hasBirthPlace)</i> | <i>lca(Hawaii, Hope)</i> |



Example

President

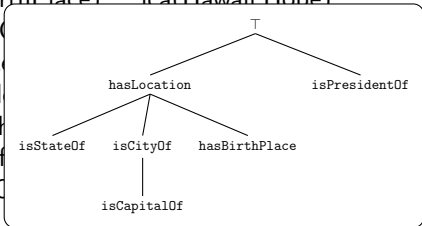
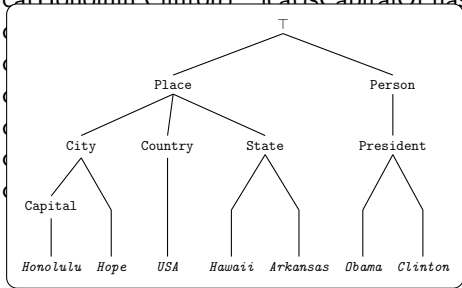
lca(Obama, Clinton)
lca(Obama, Hope)
lca(Obama, Arkansas)
lca(Obama, Clinton)
lca(Obama, Clinton)
lca(Obama, Hope)
lca(Obama, Arkansas)
lca(Honolulu, Clinton)
lca(Honolulu, Clinton)

lca(isPresidentOf, isPresidentOf)

lca(isPresidentOf, hasBirthPlace)
lca(isPresidentOf, isCityOf)
lca(isPresidentOf, isStateOf)
lca(hasBirthPlace, isPresidentOf)
lca(hasBirthPlace, hasBirthPlace)
lca(hasBirthPlace, isCityOf)
lca(hasBirthPlace, isStateOf)
lca(isCapitalOf, isPresidentOf)
lca(isCapitalOf, hasBirthPlace)

lca(USA, USA)

lca(Honolulu, Hope)
lca(Honolulu, Arkansas)
lca(USA, USA)
lca(Honolulu, USA)
lca(Honolulu, Hope)
lca(Honolulu, Arkansas)
lca(Honolulu, USA)
lca(Hawaii, USA)
lca(Hawaii, Hope)



Example

President

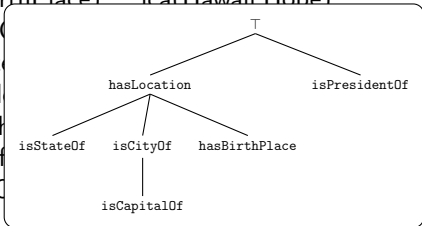
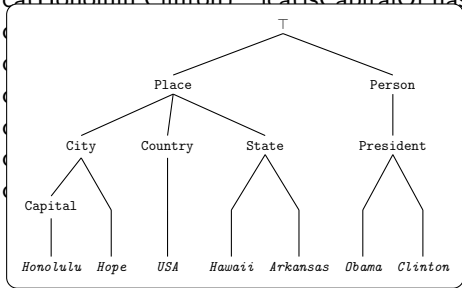
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Obama,Clinton)
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Honolulu,Clinton)
lca(Honolulu,Clinton)

isPresidentOf

lca(isPresidentOf,hasBirthPlace)
lca(isPresidentOf,isCityOf)
lca(isPresidentOf,isStateOf)
lca(hasBirthPlace,isPresidentOf)
lca(hasBirthPlace,hasBirthPlace)
lca(hasBirthPlace,isCityOf)
lca(hasBirthPlace,isStateOf)
lca(isCapitalOf,isPresidentOf)
lca(isCapitalOf,hasBirthPlace)

lca(USA,USA)

lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(USA,USA)
lca(Honolulu,USA)
lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(Honolulu,USA)
lca(Hawaii,USA)
lca(Hawaii,Hope)



Example

President

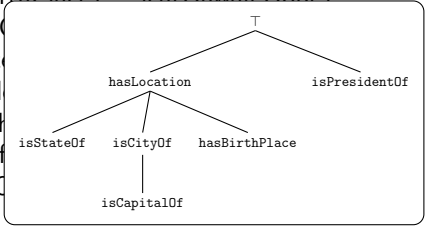
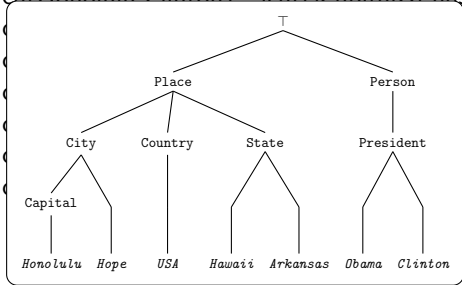
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Obama,Clinton)
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Honolulu,Clinton)
lca(Honolulu,Clinton)

isPresidentOf

lca(isPresidentOf,hasBirthPlace)
lca(isPresidentOf,isCityOf)
lca(isPresidentOf,isStateOf)
lca(hasBirthPlace,isPresidentOf)
lca(hasBirthPlace,hasBirthPlace)
lca(hasBirthPlace,isCityOf)
lca(hasBirthPlace,isStateOf)
lca(isCapitalOf,isPresidentOf)
lca(isCapitalOf,hasBirthPlace)

Country

lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(USA,USA)
lca(Honolulu,USA)
lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(Honolulu,USA)
lca(Hawaii,USA)
lca(Hawaii,Hope)



Example

President

Ica(Obama, Clinton)

Ica(Obama, Hope)

Ica(Obama, Arkansas)

Ica(Obama, Clinton)

Ica(Obama, Clinton)

Ica(Obama, Hope)

Ica(Obama, Arkansas)

Ica(Honolulu, Clinton)

Ica(Honolulu, Clinton)

isPresidentOf

Ica(isPresidentOf, hasBirthPlace)

Ica(isPresidentOf, isCityOf)

Ica(isPresidentOf, isStateOf)

Ica(hasBirthPlace, isPresidentOf)

Ica(hasBirthPlace, hasBirthPlace)

Ica(hasBirthPlace, isCityOf)

Ica(hasBirthPlace, isStateOf)

Ica(isCapitalOf, isPresidentOf)

Ica(isCapitalOf, hasBirthPlace)

Country

Ica(Honolulu, Hope)

Ica(Honolulu, Arkansas)

Ica(USA, USA)

Ica(Honolulu, USA)

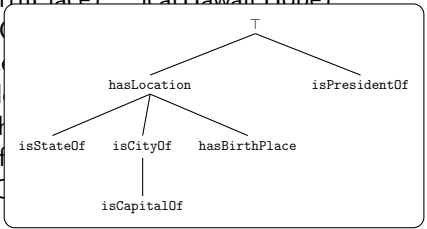
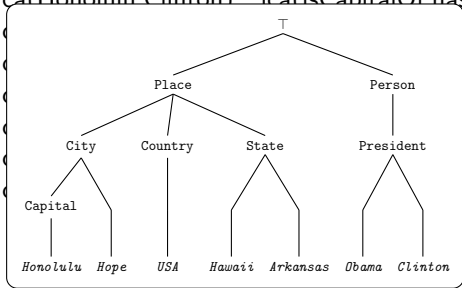
Ica(Honolulu, Hope)

Ica(Honolulu, Arkansas)

Ica(Honolulu, USA)

Ica(Hawaii, USA)

Ica(Hawaii, Hope)



Example

President

President

lca(Obama,Hope)

lca(Obama,Arkansas)

lca(Obama,Clinton)

lca(Obama,Clinton)

lca(Obama,Hope)

lca(Obama,Arkansas)

lca(Honolulu,Clinton)

lca(Honolulu,Clinton)

isPresidentOf

lca(isPresidentOf,hasBirthPlace)

lca(isPresidentOf,isCityOf)

lca(isPresidentOf,isStateOf)

lca(hasBirthPlace,isPresidentOf)

lca(hasBirthPlace,hasBirthPlace)

lca(hasBirthPlace,isCityOf)

lca(hasBirthPlace,isStateOf)

lca(isCapitalOf,isPresidentOf)

lca(isCapitalOf,hasBirthPlace)

Country

lca(Honolulu,Hope)

lca(Honolulu,Arkansas)

lca(USA,USA)

lca(Honolulu,USA)

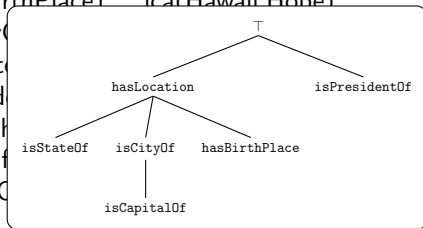
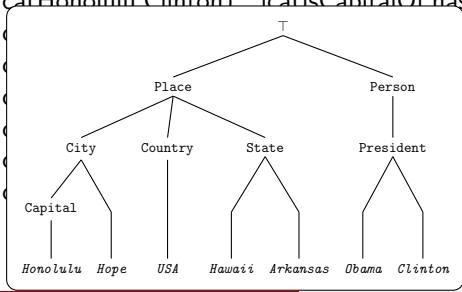
lca(Honolulu,Hope)

lca(Honolulu,Arkansas)

lca(Honolulu,USA)

lca(Hawaii,USA)

lca(Hawaii,Hope)



Example

President

President

lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Obama,Clinton)
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Honolulu,Clinton)
lca(Honolulu,Clinton)

isPresidentOf

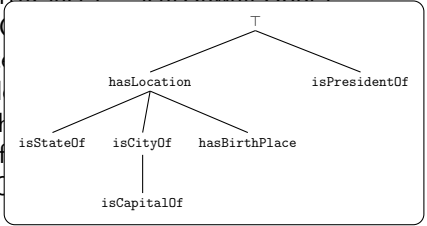
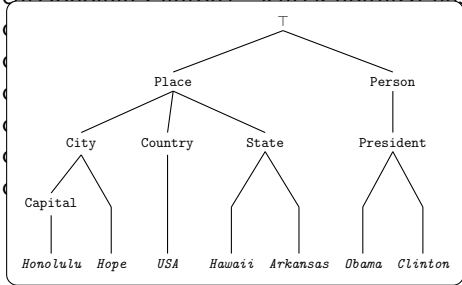
T

lca(isPresidentOf,isCityOf)
lca(isPresidentOf,isStateOf)
lca(hasBirthPlace,isPresidentOf)
lca(hasBirthPlace,hasBirthPlace)
lca(hasBirthPlace,isCityOf)
lca(hasBirthPlace,isStateOf)
lca(isCapitalOf,isPresidentOf)
lca(isCapitalOf,hasBirthPlace)

Country

lca(Honolulu,Hope)

lca(Honolulu,Arkansas)
lca(USA,USA)
lca(Honolulu,USA)
lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(Honolulu,USA)
lca(Hawaii,USA)
lca(Hawaii,Hope)



Example

President

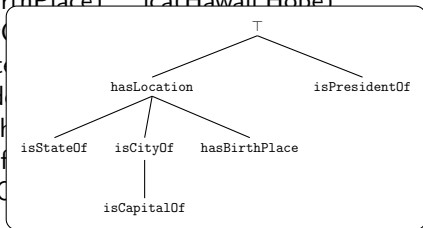
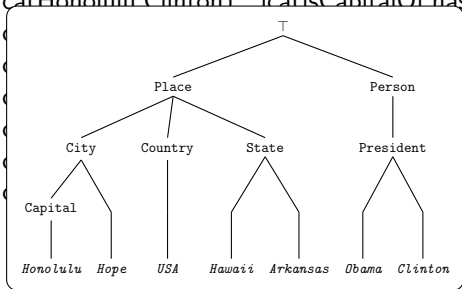
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Obama,Clinton)
lca(Obama,Clinton)
lca(Obama,Hope)
lca(Obama,Arkansas)
lca(Honolulu,Clinton)
lca(Honolulu,Clinton)

isPresidentOf

lca(isPresidentOf,isCityOf)
lca(isPresidentOf,isStateOf)
lca(hasBirthPlace,isPresidentOf)
lca(hasBirthPlace,hasBirthPlace)
lca(hasBirthPlace,isCityOf)
lca(hasBirthPlace,isStateOf)
lca(isCapitalOf,isPresidentOf)
lca(isCapitalOf,hasBirthPlace)

Country

lca(Honolulu,Arkansas)
lca(USA,USA)
lca(Honolulu,USA)
lca(Honolulu,Hope)
lca(Honolulu,Arkansas)
lca(Honolulu,USA)
lca(Hawaii,USA)
lca(Hawaii,Hope)



Example

President

lca(Obama,Clinton)

lca(Honolulu,Hope)

lca(Honolulu,Arkansas)

lca(Hawaii,Hope)

lca(Hawaii,Arkansas)

isPresidentOf

lca(hasBirthPlace,hasBirthPlace)

lca(isCapitalOf,isCityOf)

lca(isCapitalOf,isStateOf)

lca(isStateOf,isCityOf)

lca(isStateOf,isStateOf)

Country

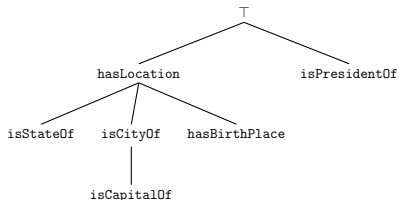
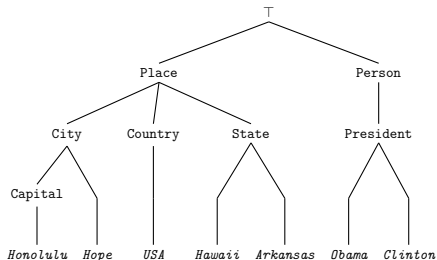
lca(Honolulu,Hope)

lca(Hawaii,Arkansas)

lca(Hawaii,USA)

lca(USA,Arkansas)

lca(USA,USA)

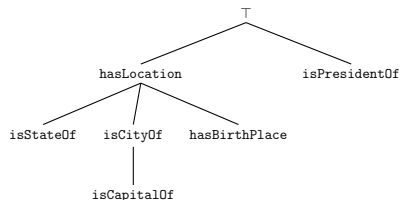
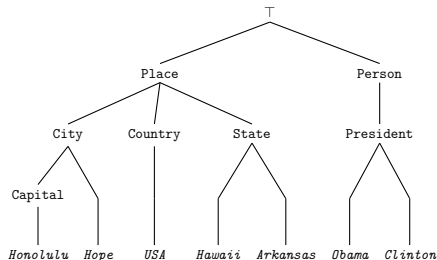


Example

President
President
City
Place
Place
State

isPresidentOf
hasBirthPlace
isCityOf
hasLocation
hasLocation
isStateOf

Country
City
State
Place
Place
Country

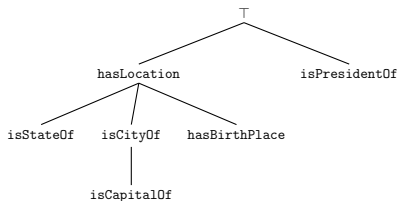
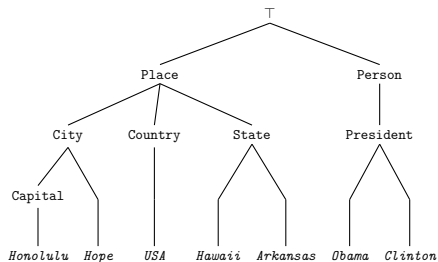


Example

President
President
City
Place
State

isPresidentOf
hasBirthPlace
isCityOf
hasLocation
isStateOf

Country
City
State
Place
Country






Contents

- 1 Introduction
- 2 Web of Data
 - Definitions
 - Toy example
- 3 FCA extensions
 - Background knowledge
 - Pattern structures
 - Related works
- 4 Our approach
 - Definition
 - Example
- 5 Conclusion

- Implementation of pattern structures with a taxonomy
- Experiments with outputs from FRED
<http://wit.istc.cnr.it/stlab-tools/fred/>

Références I

-  Alam, M., Buzmakov, A., Napoli, A., and Sailanbayev, A. (2015). Revisiting pattern structures for structured attribute sets. In *Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015.*, pages 241–252.
-  Carpineto, C. and Romano, G. (2004). *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, Chichester, UK.
-  Ferré, S. (2015). A Proposal for Extending Formal Concept Analysis to Knowledge Graphs. In *Formal Concept Analysis*, volume LNCS 9113, pages 271–286, Nerja, Spain.

-  Ganter, B. and Kuznetsov, S. O. (2001).
Pattern structures and their projections.
In Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings, pages 129–142.
-  Kötters, J. (2015).
Concept lattices of rdf graphs.
Formal Concept Analysis and Applications FCA&A 2015, page 81.
-  Lehmann, F. and Wille, R. (1995).
A triadic approach to formal concept analysis.
Springer.