Linkex: A Tool for Link Key Discovery Based on Pattern Structures

Nacira Abbas*, Jérôme David**, Amedeo Napoli*

*Université de Lorraine, CNRS, Inria, Loria, Nancy, France.

**Université de Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, Grenoble, France.

Journée thématique EGC & IA

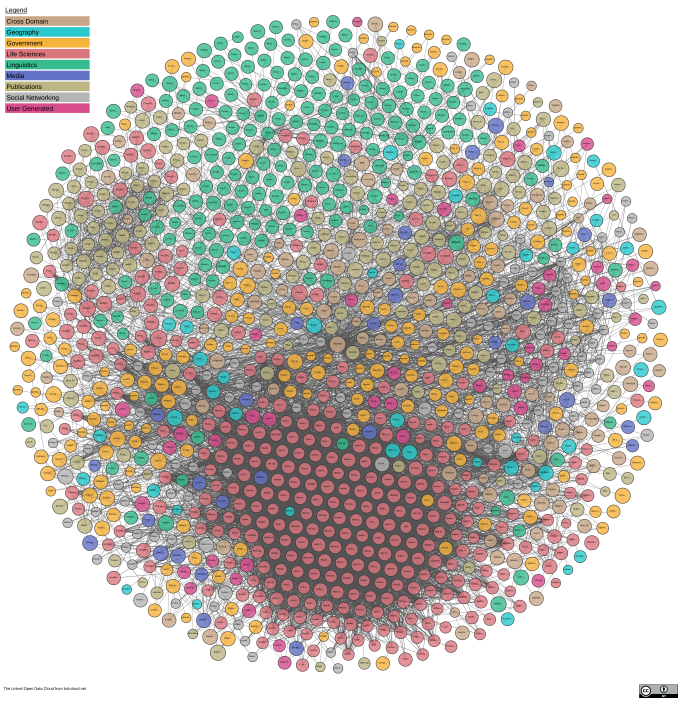
May 10, 2019

ELKER-Enhancing Link Keys: Extraction and Reasoning ANR 2017 Project (ANR-17-CE23-0007-01)

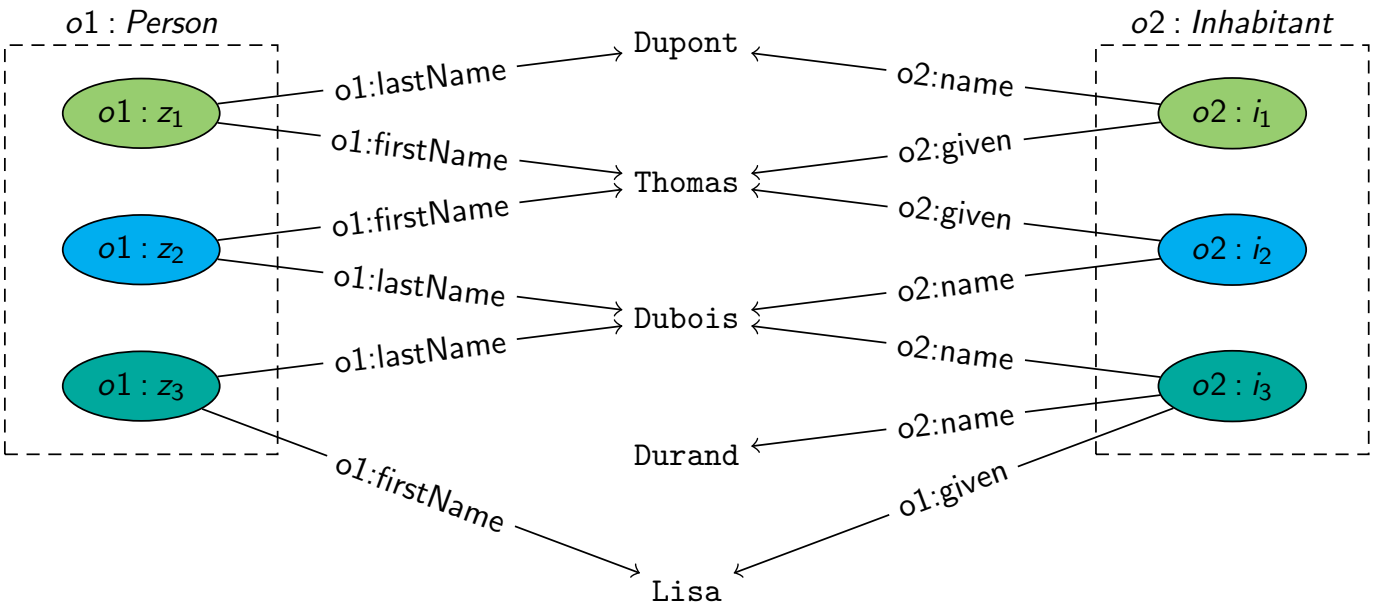
Outline

Data interlinking
Motivations

- Many organisations publish their data on the web.
- These datasets need to be linked to other datasets.
- same-as links identify the same entity in different datasets.
- **Data interlinking** is the task of finding the same entities described in different RDF (Resource Description Framework) datasets.



Data interlinking



Same-as links: $\langle z_1, i_1 \rangle$, $\langle z_2, i_2 \rangle$ and $\langle z_3, i_3 \rangle$.

Data interlinking
Approaches

There are two main approaches to data interlinking:

- similarity-based.
- rule-based (symbolic).

Data interlinking
Approaches

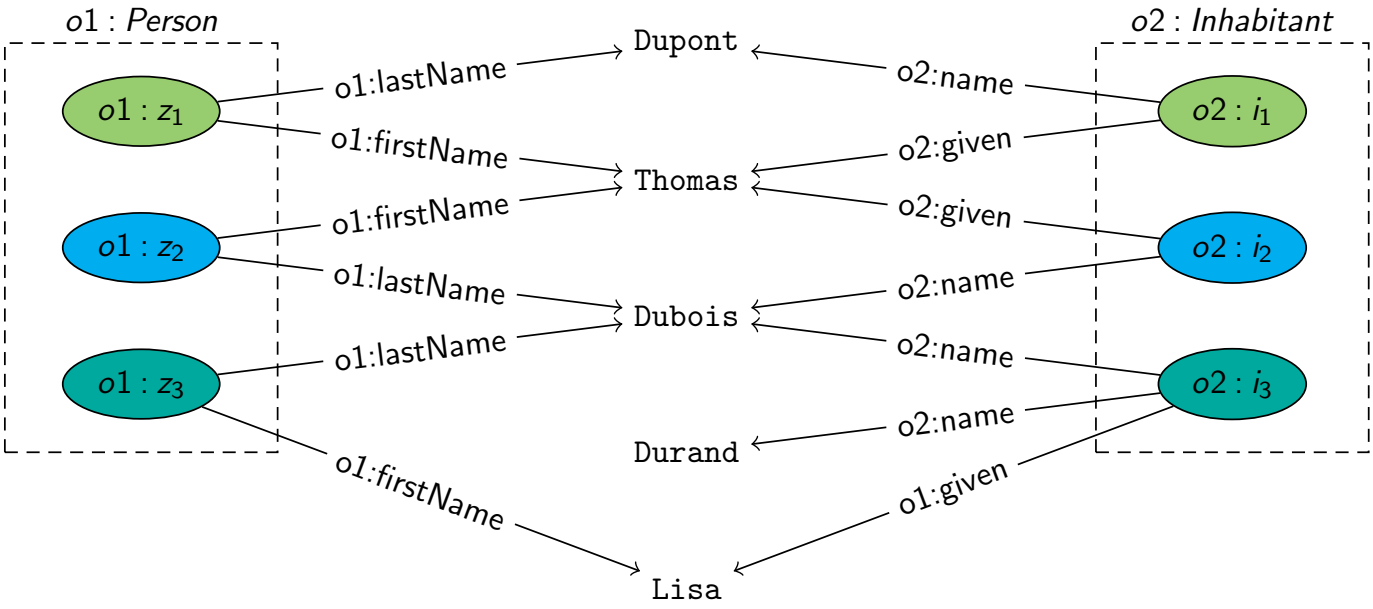
There are two main approaches to data interlinking:

- similarity-based.
- rule-based (symbolic).
  - Link keys which generalise keys from relational databases.

Link Keys

Example of a link key:

$$\underbrace{\{\langle \text{firstName}, \text{given}\rangle\}}_{\text{EQ}}\underbrace{\{\langle \text{lastName}, \text{name}\rangle\}}_{\text{IN}} \; \textit{linkkey} \; \underbrace{\langle \text{Person}, \text{Inhabitant}\rangle}_{\text{Classes}}$$



Link Keys

3 forms

1. **Link key expression**: syntactic form of a link key

$$\underbrace{\{\langle p_1, p_1^{'}\rangle, .., \langle p_k, p_k^{'}\rangle\}}_{\text{EQ}} \underbrace{\{\langle p_1, p_1^{'}\rangle, .., \langle p_k, p_k^{'}\rangle, .., \langle p_n, p_n^{'}\rangle\}}_{\text{IN}}, \langle C, C^{'}\rangle$$

2. **Link key candidate**: link key expression which identifies at least one link and it is maximal.

3. **Link key**: link key candidate which identifies all the correct links among two classes $\langle C, C^{'}\rangle$

$$\underbrace{\{\langle p_1, p_1^{'}\rangle, .., \langle p_k, p_k^{'}\rangle\}}_{\text{EQ}} \underbrace{\{\langle p_1, p_1^{'}\rangle, .., \langle p_k, p_k^{'}\rangle, .., \langle p_n, p_n^{'}\rangle\}}_{\text{IN}} \; \textit{linkkey} \; \langle C, C^{'}\rangle$$

$D, D^{'}$ are two RDF datasets.
C and $C^{'}$ classes, P and $P^{'}$ properties from $D, D^{'}$ respectively.
$p_1, .., p_n \in P$, $p_1^{'}, .., p_n^{'} \in P^{'}$

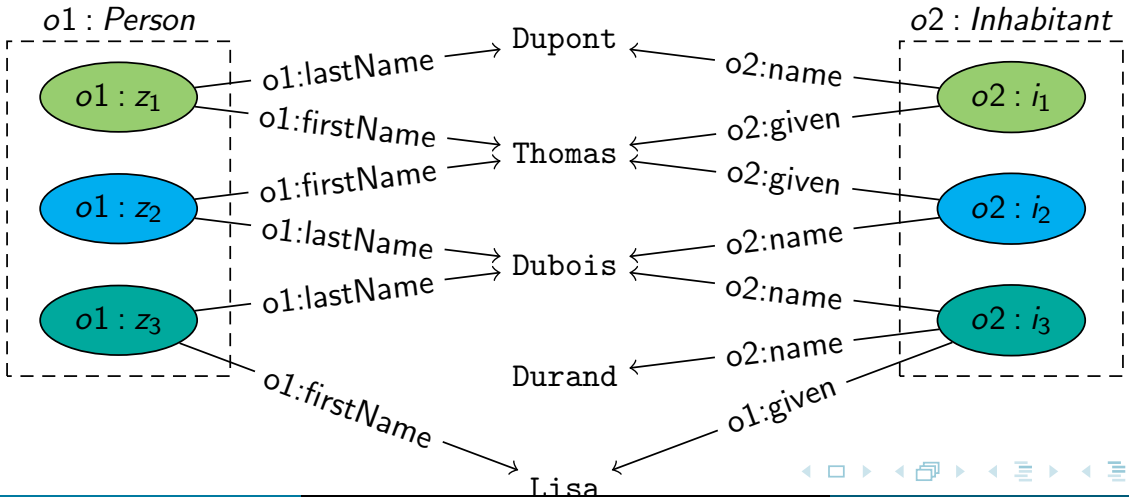
Link keys

3 forms

Link key expression: syntactic form of a link key

Generates links: $\underbrace{\{\langle \text{firstName}, \text{given}\rangle\}}_{\text{EQ}}\underbrace{\{\langle \text{lastName}, \text{name}\rangle\}}_{\text{IN}}$

Do not generates links: $\underbrace{\{\langle \text{firstName}, \text{name}\rangle\}}_{\text{EQ}}\underbrace{\{\langle \text{lastName}, \text{given}\rangle\}}_{\text{IN}}$



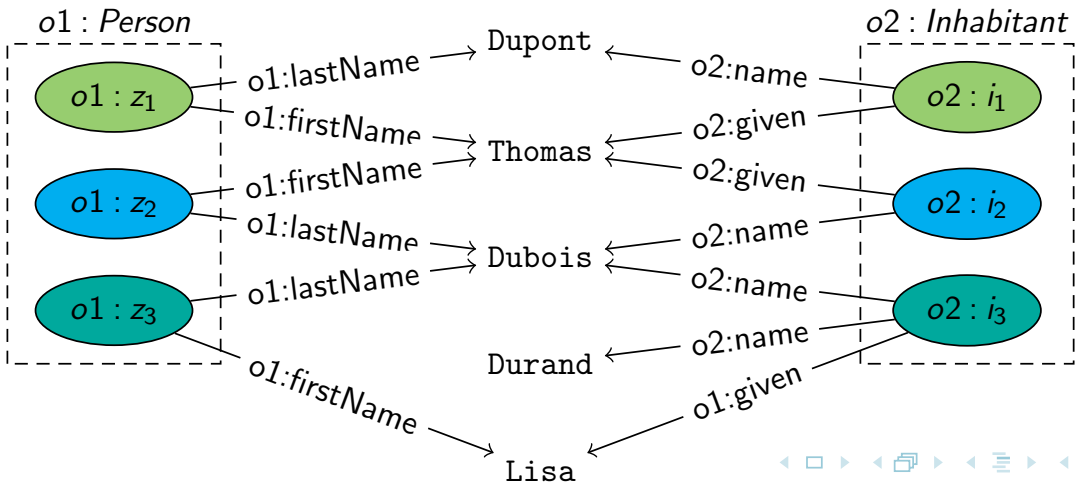
Link Keys

Link key candidate

Link key candidate: link key expression which allows to identify at least one link and it is maximal

Link key candidate: $\underbrace{\{\langle \text{firstName}, \text{given}\rangle\}}_{\text{EQ}}\underbrace{\{\langle \text{lastName}, \text{name}\rangle\}}_{\text{IN}}$

Not a link key candidate: $\underbrace{\{\langle \text{firstName}, \text{name}\rangle\}}_{\text{EQ}}\underbrace{\{\langle \text{lastName}, \text{given}\rangle\}}_{\text{IN}}$



Link Keys
Lattice of link key expression

Lattice of link key expression

Link key expressions set D is a lattice $(D, \sqcap, \sqcup)$

$K_1 = Eq_1, In_1$ and $K_2 = Eq_2, In_2$ two link key expressions.

- Meet : $K_1 \sqcap K_2 = Eq_1 \cap Eq_2, In_1 \cap In_2$
- Join: $K_1 \sqcup K_2 = Eq_1 \cup Eq_2, In_1 \cup In_2$
- Partial order: $K_1 \sqsubseteq K_2$ iff $K_1 \sqcap K_2 = K_1$

Link Keys
Lattice of link key expression

Lattice of link key expression

Link key expressions set D is a lattice $(D, \sqcap, \sqcup)$

$K_1 = Eq_1, In_1$ and $K_2 = Eq_2, In_2$ two link key expressions.

- Meet : $K_1 \sqcap K_2 = Eq_1 \cap Eq_2, In_1 \cap In_2$
- Join: $K_1 \sqcup K_2 = Eq_1 \cup Eq_2, In_1 \cup In_2$
- Partial order: $K_1 \sqsubseteq K_2$ iff $K_1 \sqcap K_2 = K_1$

For example we have:

$K_1 = \{\langle lastName, name\rangle, \langle firstName, given\rangle\}\{..\}$

$K2 = \{\langle firstName, given\rangle\}\{..\}$

$K_1 \sqcap K_2 = \{\langle firstName, given\rangle\}\{..\}$

$K_1 \sqcap K_2 = K_2,\ K_2 \sqsubseteq K_1.$

$K_1 \sqcup K_2 = \{\langle lastName, name\rangle, \langle firstName, given\rangle\}\{..\}.$

Link Keys

Satisfaction of a link key expression

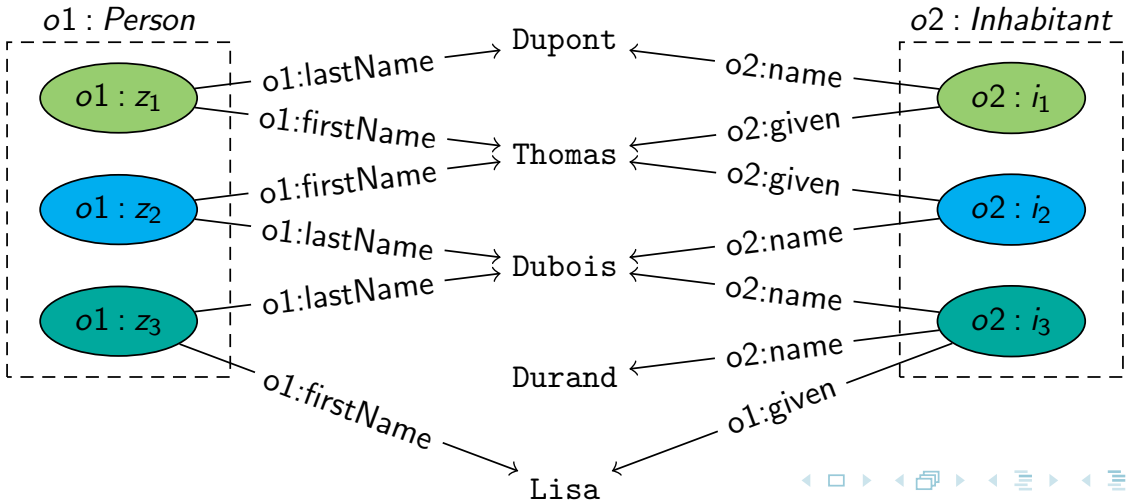
Satisfaction of a link key expression

A link $\langle s, s' \rangle \in C \times C'$ satisfies the link key expression $K = Eq, In$ and we write $K \models \langle s, s' \rangle$ if

$\forall \langle p, p' \rangle \in Eq \Longrightarrow p(s) = p'(s') \neq \emptyset$ and $\forall \langle p, p' \rangle \in In \Longrightarrow p(s) \cap p'(s') \neq \emptyset$

Example: $\langle z_3, i_3 \rangle$ satisfies $\{\langle firstName, given \rangle\}, \{\langle lastName, name \rangle\}$

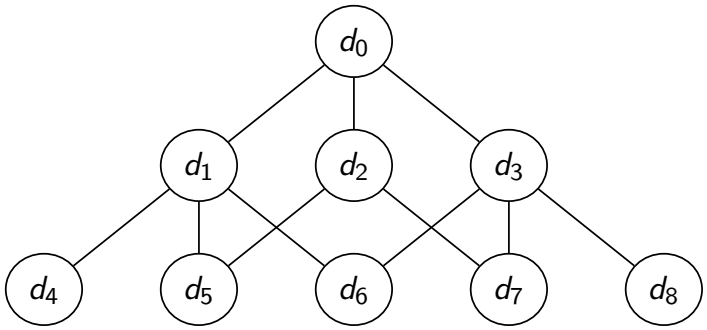
$\langle z_3, i_3 \rangle$ do not satisfy $\{\langle firstName, given \rangle, \langle lastName, name \rangle\}, \{\}$



Pattern Structures

Pattern Structures (PS) are a generalisation of Formal Concept Analysis (FCA) to deal with complex data such as graphs or intervals.

objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



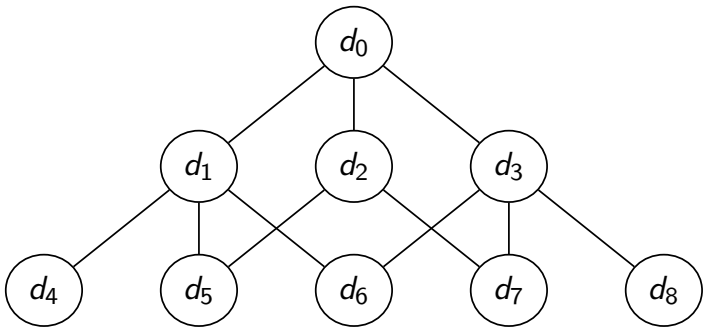
Pattern Structures

A Pattern Structure is a triple $(G, (D, \sqcap), \delta)$

- G is a set of objects.
- $(D, \sqcap)$ a semilattice of descriptions.
- The mapping $\delta : G \rightarrow D$ maps an object to a description.

$\delta(g_1) = d_8$

objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



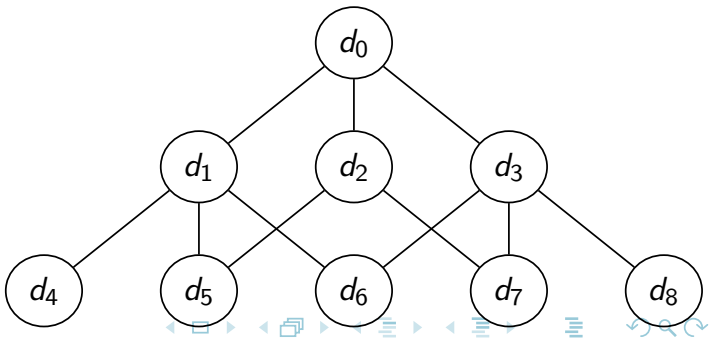
Pattern Structures

Descriptions are partially ordered in a meet semilattice by the partial order $\sqsubseteq$ defined w.r.t. the similarity operator $\sqcap$.

If a and b are two descriptions then $a \sqcap b = a \Leftrightarrow a \sqsubseteq b$ and a is subsumed by b.

$d_1 \sqcap d_4 = d_1 \Leftrightarrow d_1 \sqsubseteq d_4$ and d_1 is subsumed by d_4.

objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



Pattern Structures

Derivation operators $.^{\square}$

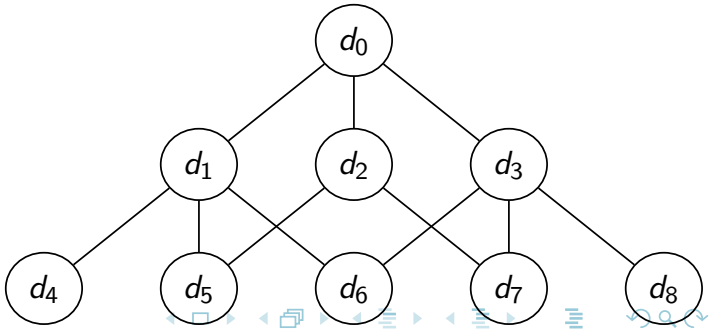
$$A^{\square} = \bigsqcap_{g \in A} \delta(g) \qquad A \subseteq G$$

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad d \in D$$

$\delta(g_3) = d_1$ $\delta(g_4) = d_4$ $d_1 \sqcap d_4 = d_1$

$\{g_3, g_4\}^{\square} = d_1$ $d_1^{\square} = \{g_3, g_4\}$

objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



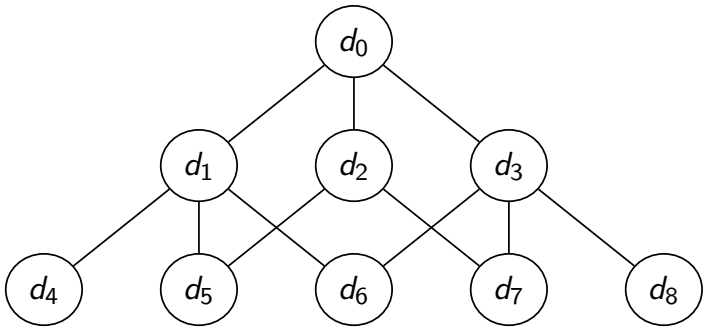
Pattern Structures

Pattern Concept (A, d) is a *Pattern Concept* if $A^{\square} = d$ and $d^{\square} = A$.
$A^{\square\square} = A$ and $d^{\square\square} = d$ (closed)

$(\{g_3, g_4\}, d_1)$ is a pattern concept .

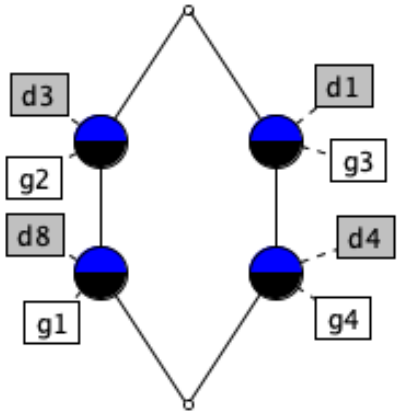
$\{g_3, g_4\}^{\square} = d_1 \qquad d_1^{\square} = \{g_3, g_4\}$

objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



A *Pattern Concept* (A_1, d_1) is subsumed by another *Pattern Concept* (A_2, d_2) if $A_1 \subseteq A_2$ (or equivalently if $d_2 \sqsubseteq d_1$).

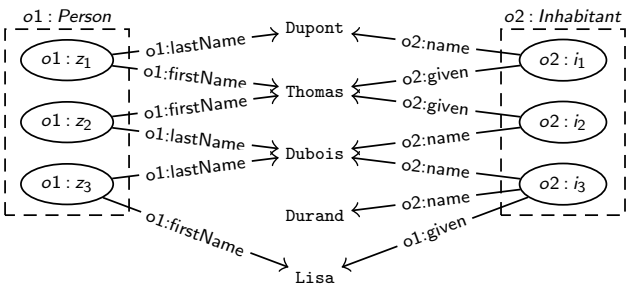
objects	descriptions
g_1	d_8
g_2	d_3
g_3	d_1
g_4	d_4



Link Key Discovery Based on Pattern Structures

A Pattern Structure for link key extraction $(C \times C', (D, \sqcap, \sqcup), \delta)$

- $C \times C'$ is the set a pair of instances $\langle s, s' \rangle \in C \times C'$.
- D is a set of link key expressions.
- $\delta(\langle s, s' \rangle) = \sqcup\{K | K \models \langle s, s' \rangle\}$

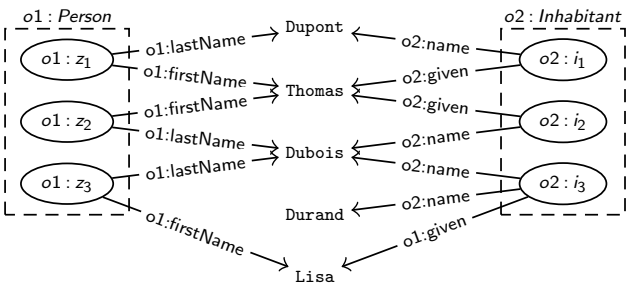


	Eq	In
$\langle z_1, i_1 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_1, i_2 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_1 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_2 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_3 \rangle$	$\emptyset$	$\{\langle lastName, name \rangle\}$
$\langle z_3, i_2 \rangle$	$\{\langle lastName, name \rangle\}$	$\{..\}$
$\langle z_3, i_3 \rangle$	$\{\langle firstName, given \rangle\}$	$\{\langle lastName, name \rangle\}$

Link Key Discovery Based on Pattern Structures

$\langle z_3, i_3 \rangle$ satisfies the link key expressions:
$\{\langle firstName, given \rangle\}$ $\{\langle firstName, given \rangle\}$
$\{\langle firstName, given \rangle\}$ $\{\langle firstName, given \rangle, \langle lastName, name \rangle\}$

$\delta(\langle z_3, i_3 \rangle) = \{\langle firstName, given \rangle\}$ $\{\langle firstName, given \rangle, \langle lastName, name \rangle\}$



	Eq	*In*
$\langle z_1, i_1 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_1, i_2 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_1 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_2 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_3 \rangle$	$\emptyset$	$\{\langle lastName, name \rangle\}$
$\langle z_3, i_2 \rangle$	$\{\langle lastName, name \rangle\}$	$\{..\}$
$\langle z_3, i_3 \rangle$	$\{\langle firstName, given \rangle\}$	$\{\langle lastName, name \rangle\}$

Link Key Discovery Based on Pattern Structures

Derivation operators $.^{\square}$ are defined as follows for $A \subseteq C \times C^{'}$ and $K \in D$:

$A^{\square} = \sqcap_{\langle s,s' \rangle \in A} \delta(\langle s, s^{'} \rangle)$ and $K^{\square} = \{\langle s, s^{'} \rangle \mid K \sqsubseteq \delta(\langle s, s^{'} \rangle)\}$

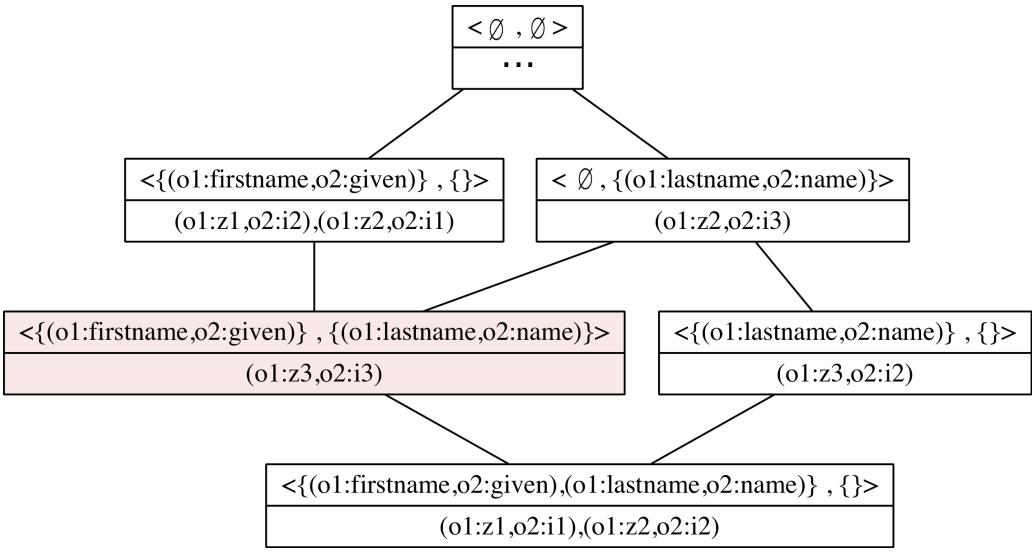
$\{\langle z_1, i_1 \rangle, \langle z_2, i_2 \rangle, \langle z_3, i_3 \rangle\}^{\square} = \{\langle firstName, given \rangle\}\{\langle lastName, name \rangle\}$

$\{\langle firstName, given \rangle\}\{\langle lastName, name \rangle\}^{\square} = \{\langle z_1, i_1 \rangle, \langle z_2, i_2 \rangle, \langle z_3, i_3 \rangle\}$

	Eq	In
$\langle z_1, i_1 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_1, i_2 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_1 \rangle$	$\{\langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_2 \rangle$	$\{\langle lastName, name \rangle, \langle firstName, given \rangle\}$	$\{..\}$
$\langle z_2, i_3 \rangle$	$\emptyset$	$\{\langle lastName, name \rangle\}$
$\langle z_3, i_2 \rangle$	$\{\langle lastName, name \rangle\}$	$\{..\}$
$\langle z_3, i_3 \rangle$	$\{\langle firstName, given \rangle\}$	$\{\langle lastName, name \rangle\}$

Link Key Discovery Based on Pattern Structures

If (A, K) is a pattern concept, then its intent K represents a link key candidate and its extent is the link set generated by this link key candidate.



Linkex

Prototype tool called `Linkex`[1] performs the following treatments:

1. **From RDF datasets to pattern structure**: value normalization, remove non-discriminating properties and properties with weak support, compute the description of each pair of instances and construct the Pattern Structure.
2. **Building the pattern concept lattice**: generates the pattern concept lattice using a modified version of `AddIntent` algorithm.
3. Visualize link key candidates in a nice way i.e. organized in a lattice represented by a Hasse diagram.

[1] https://gitlab.inria.fr/moex/linkex

Linkex
Experimentation results

Bibliographical datasets:

1. BnF[2], "Bibliothèque nationale de France" .
2. IdRef[3], "Agence Bibliographique de l'Enseignement Supérieur" .

Samples:

1. **Variant 1**: authors instances that have a combination firstname, name which is in the top 1000 most frequent homonyms.
2. **Variant 2**: authors instance that have a name starting with letter 'A'.

[2] https://data.bnf.fr
[3] https://www.idref.fr

Linkex
Experimentation results

Table: Experimentation results

Variant	#BnF	#IdRef	#Descriptions	#PatternConcepts	Time
Variant 1	15,421	8,162	1,564,495	155	2m10
Variant 2	142,571	18,637	12,348,012	186	6m50

Machine: MacBookPro11,3, Intel Core i7 @2,3 GHz with 10GB RAM allocated to the Java virtual machine.

Conclusions and perspectives

In this work, we:

1. Formalize the problem of extracting link key candidates using Pattern Structures.
2. Propose a tool allowing to automatically build a pattern concept lattice where each pattern concept intent is a candidate link key.

We aim to extend this work to :

1. Consider interdependent classes i.e when rdf objects are instances of other classes.
2. Relax equality constraints used to compare literals by considering similarity measures instead.

Thank you!

Questions?