Experimenting link key extraction between BnF, ONOMA, and Abes datasets

Jérôme David & Aude Le Moullec-Rieu

Jerome.David@inria.fr
http://moex.inria.fr

{BnF

aude.le-moullec-rieu@bnf.fr

November 22, 2018

Objective

Explore the possibility to use link key extraction for linking data from BnF, Abes and Ministère de la Culture.

Outline of the project:

- Define samples of data for the experiments
- Experiments: execute algorithm, analyze results and make it evolve
- Final analysis of the results
- Deliver the code link extraction algorithm (under LGPL)

Team

- BnF
  - Anila Angjeli, cheffe de projet Fichier national d'entités
  - Aude Le Moullec-Rieu, adjointe à la cheffe du service Diffusion des métadonnées
- Ministère de la Culture
  - Katell Briatte, Cheffe du département des systèmes d'information patrimoniaux
  - Marie-Véronique Leroi, Département de l'innovation numérique (SG/SCPCI)
- Abes
  - Aline Le Provost
- Inria
  - Jérôme David, Enseignant-Chercheur, projet Moex

What is linked data?

- Structured data expressed with semantic web technologies (RDF, OWL, etc.)
- Published on the web (deferenceable URIs, online SPARQL endpoints), and
- **Linked**: same resources in different datasets have to be identified and related through `owl:sameAs` links

Many examples available: dbpedia, data.bnf.fr, FAO, Genebank, Open street map, etc.

What is RDF?

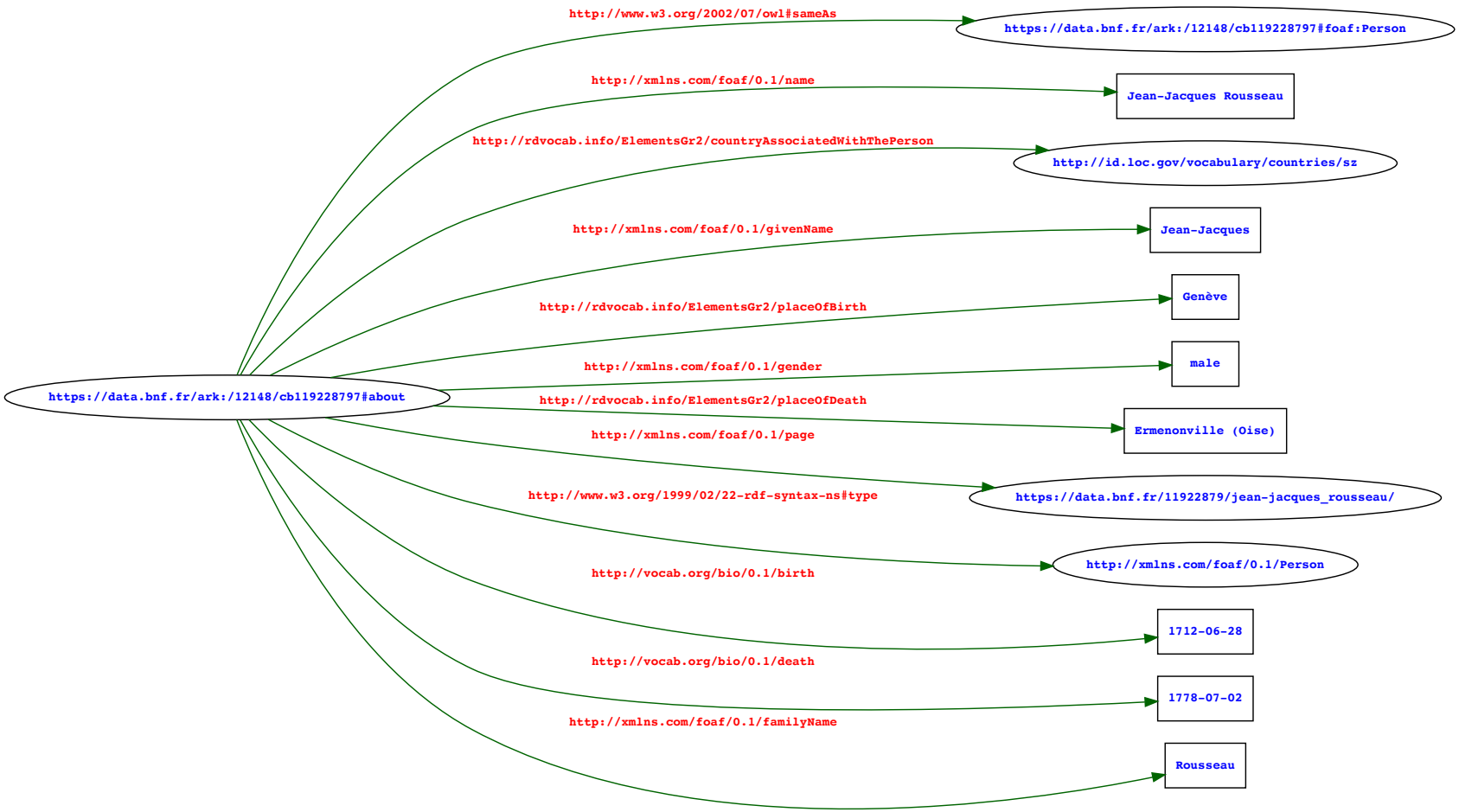
RDF is used to describe data on the semantic web.

- It expresses data as set of triples:
  ⟨`subject, predicate, object`⟩
- for instance:

  `https://data.bnf.fr/ark:/12148/cb119228797#about a foaf:Person`

  this resource is an instance of the class `foaf:Person`
- it can be represented as graph...

Example of an RDF graph



data.bnf.fr

BnF publishes data in RDF with the platform `data.bnf.fr`.

`data.bnf.fr` allows to:

- dereference URIs : `https://data.bnf.fr/ark:/12148/cb11928016k#about`
- make content negociation: HTML, RDF-XML, NT, N3
- query data using sparql
- download dumps

Data interlinking

Data interlinking is the task of finding the same entities within different datasets (RDF graphs).
For instance identifying authors between BnF and BNE.

There are two main approaches to data interlinking:

- similarity-based: resources are compared through a similarity measure and if they are similar enough, they are the same.
- rule/key-based (symbolic): logical rules expressing sufficient conditions for two resources to be the same are used to deduce same entities

Data interlinking

Data interlinking is the task of finding the same entities within different datasets (RDF graphs).
For instance identifying authors between BnF and BNE.

There are two main approaches to data interlinking:

- similarity-based: resources are compared through a similarity measure and if they are similar enough, they are the same.
- rule/key-based (symbolic): logical rules expressing sufficient conditions for two resources to be the same are used to deduce same entities

Data interlinking process

Data interlinking process can be decomposed into two phases :

1. Specify how links will be generated
   - It consists in defining similarity-based linkage rules, link keys, logical rules, etc.
   - It can be done manually or (semi-)automatically

Data interlinking process

Data interlinking process can be decomposed into two phases :

1. Specify how links will be generated
   - It consists in defining similarity-based linkage rules, link keys, logical rules, etc.
   - It can be done manually or (semi-)automatically
2. Generate links using specifications
   - single pass: all rules are applied in one single pass (via SPARQL query or link generation engine (SILK/Limes)
   - saturation/inference: all rules applied until no new links are generated (using some inference engine)

Symbolic approaches for (RDF) data interlinking

Why to use symbolic approaches ?

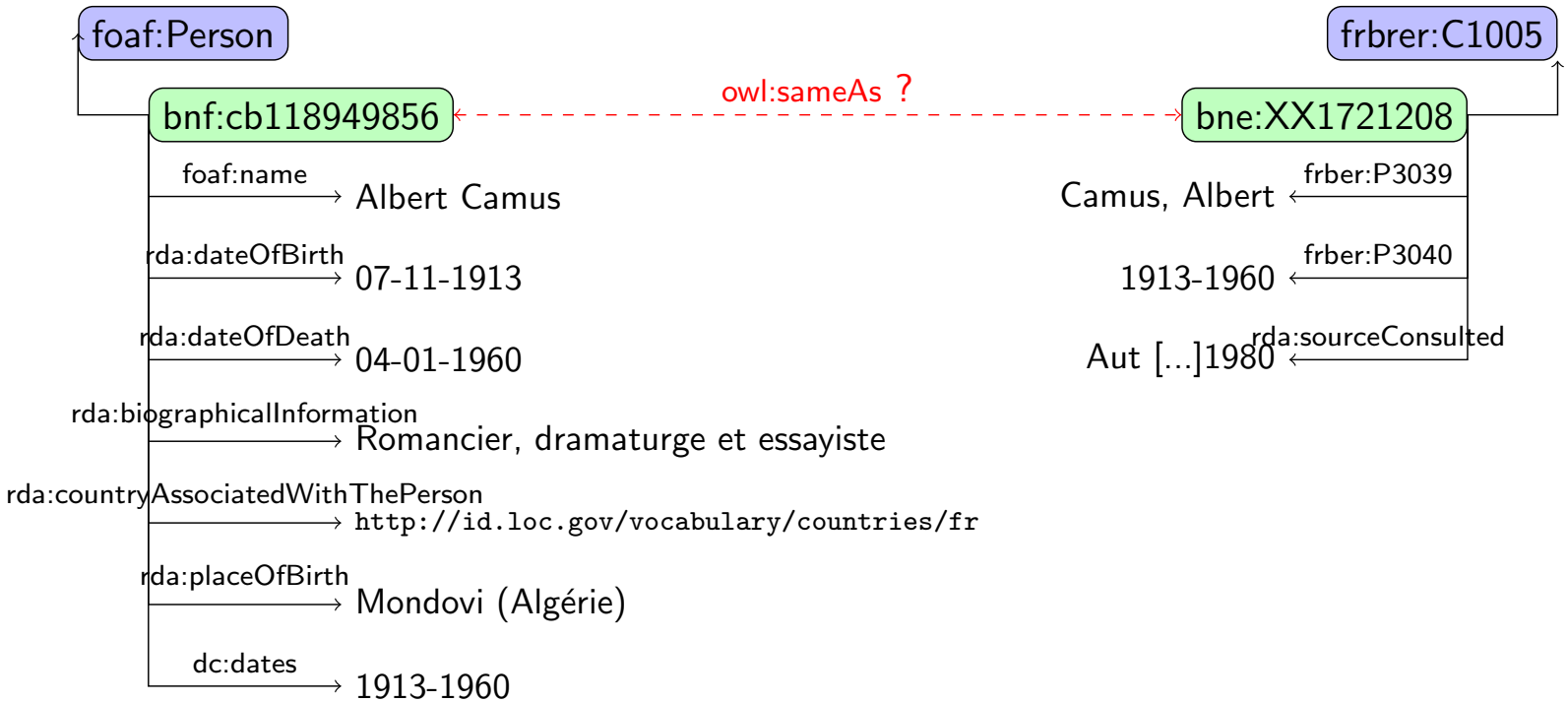
- They can be expressed as ontological constraints / rules that can be used for inferring new links
  - useful when data evolves continuously
  - can help to reduce redundancy
- They are meaningful for the user/domain expert
- They usually produce high quality links
  - precision is usually very high
  - but they are more sensitive to the quality of data (low recall)

Intuition of what is a link key

Problem: Are the resources bnf:cb118949856 and bne:XX1721208 the same?

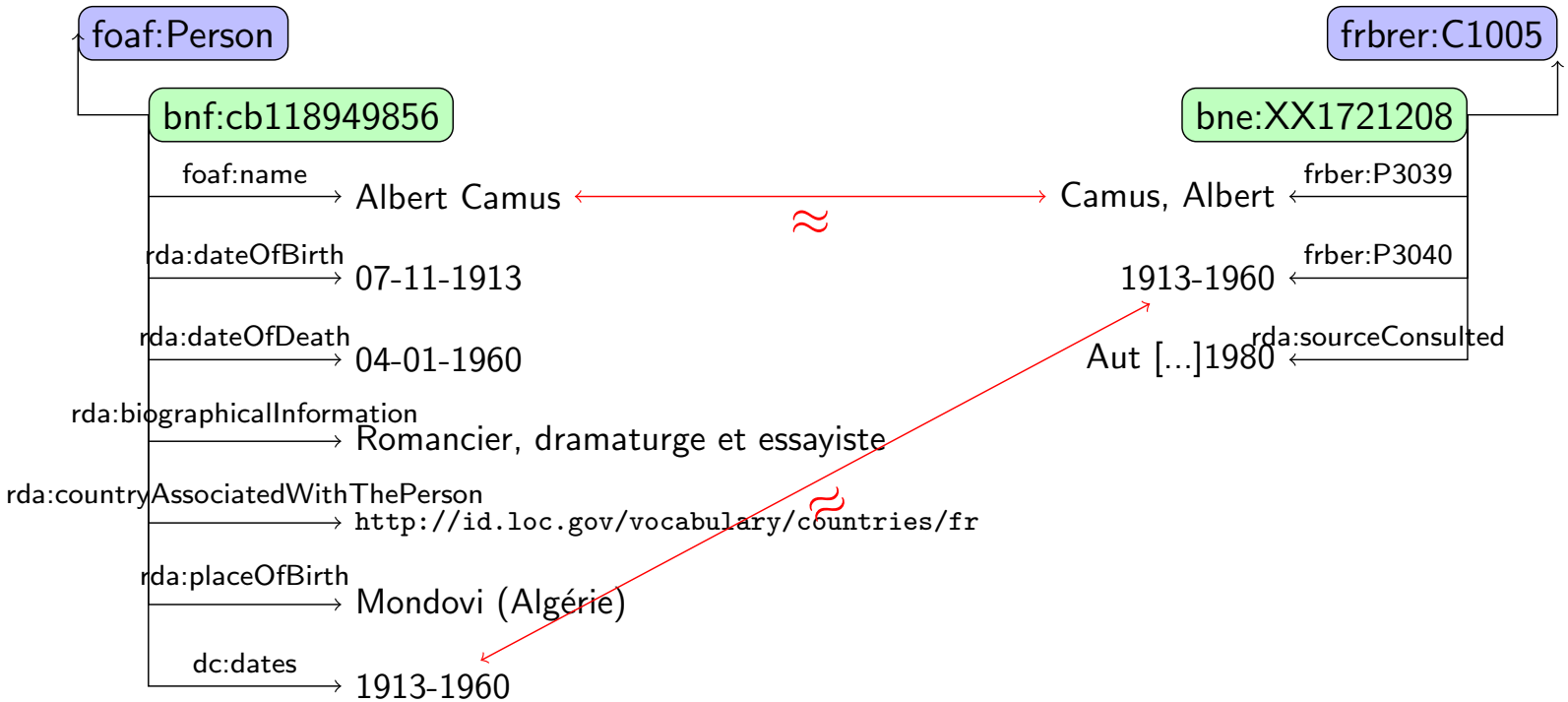
Intuition of what is a link key

Problem: Are the resources bnf:cb118949856 and bne:XX1721208 the same?



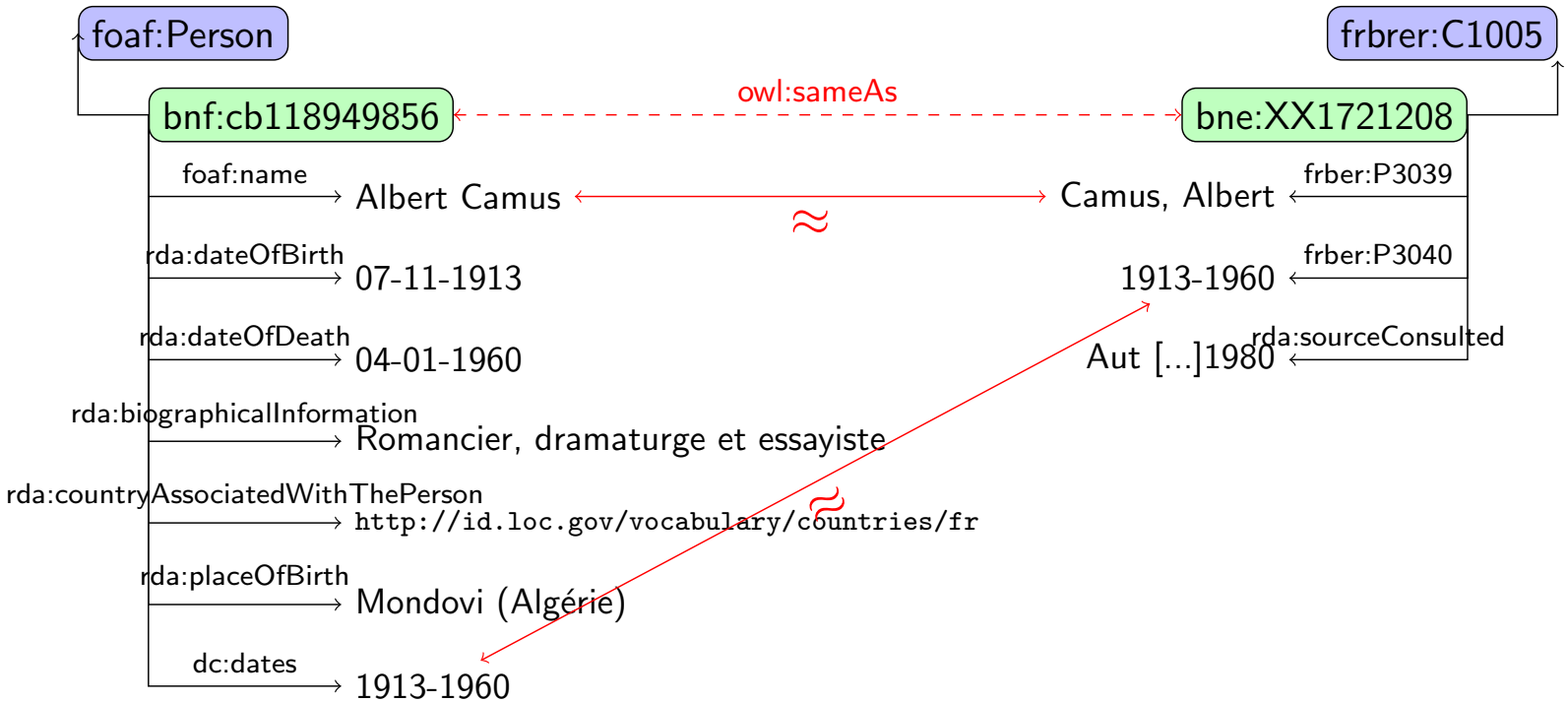
Intuition of what is a link key

Problem: Are the resources bnf:cb118949856 and bne:XX1721208 the same?



Intuition of what is a link key

Problem: Are the resources bnf:cb118949856 and bne:XX1721208 the same?



On this example a link key could be:

$\langle \{ \langle \text{foaf:name}, \text{frbr:P3039} \rangle, \langle \text{dc:dates}, \text{frbr:P3040} \rangle \}$ linkkey $\langle \text{foaf:Person}, \text{frbr:C1005} \rangle \rangle$

Link key (the full definition)

A *link key*

$$\langle \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}\ \{\langle p'_1, q'_1\rangle, \ldots, \langle p'_l, q'_l\rangle\} \text{ linkkey } \langle c, d\rangle\rangle$$

holds iff
$\forall a; \mathcal{O} \models c(a)$, $\forall b; \mathcal{O}' \models d(b)$,

$$\left.\begin{array}{rl} \text{if} & \forall i \in 1, \ldots, k, p_i(a) \cap q_i(b) \neq \varnothing \\ \text{and} & \forall i \in 1, \ldots, l, p'_i(a) = q'_i(b) \neq \varnothing \end{array}\right\} \text{ then } \langle a, \texttt{owl:sameAs}, b\rangle \text{ holds}$$

$p(s) = \{o | O \models \langle s, p, o\rangle\}$

Link key extraction

Problem: How to induce such link keys from data?

The number of set of pairs of properties is exponential

Link key extraction

Problem: How to induce such link keys from data?

The number of set of pairs of properties is exponential

Our approach:

Link key extraction

Problem: How to induce such link keys from data?

The number of set of pairs of properties is exponential

Our approach:

- compare every pair of instances and see what they share

Link key extraction

Problem: How to induce such link keys from data?

The number of set of pairs of properties is exponential

Our approach:

- compare every pair of instances and see what they share
- the maximal pairs of properties shared by pairs of instances are called candidates

Link key extraction

Problem: How to induce such link keys from data?

The number of set of pairs of properties is exponential

Our approach:

- compare every pair of instances and see what they share
- the maximal pairs of properties shared by pairs of instances are called candidates
- we evaluate candidates in order to select only the "good" ones

Candidate link key selection

- We have an algorithm for extracting them;
- But which candidate is the best?

Unsupervised selection measures

When no reference link is available.

Idea: measuring how close the extracted links would be from one-to-one and total.

Definition (Discriminability)

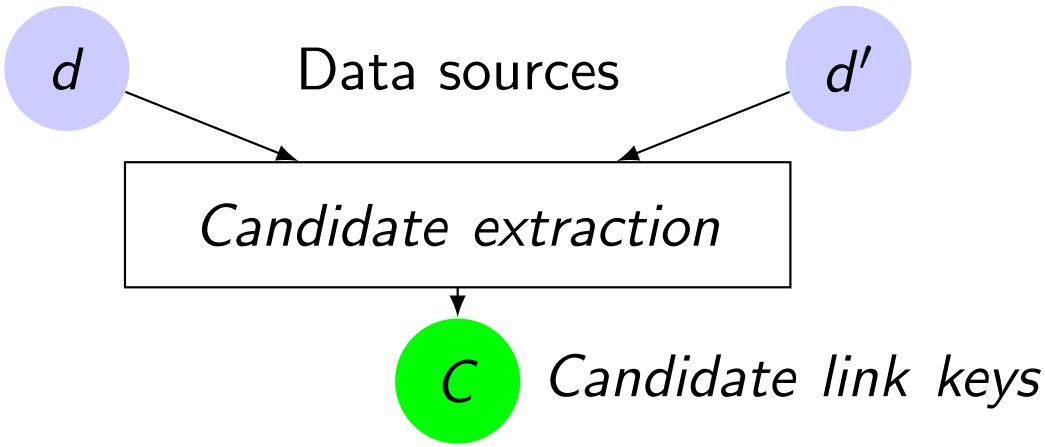
$$\mathrm{disc}(K, D, D') = \frac{\min(|\{a : \langle a, b\rangle \in L_{D,D'}(K)\}|, |\{b : \langle a, b\rangle \in L_{D,D'}(K)\}|)}{|L_{D,D'}(K)|}$$

Definition (Coverage)

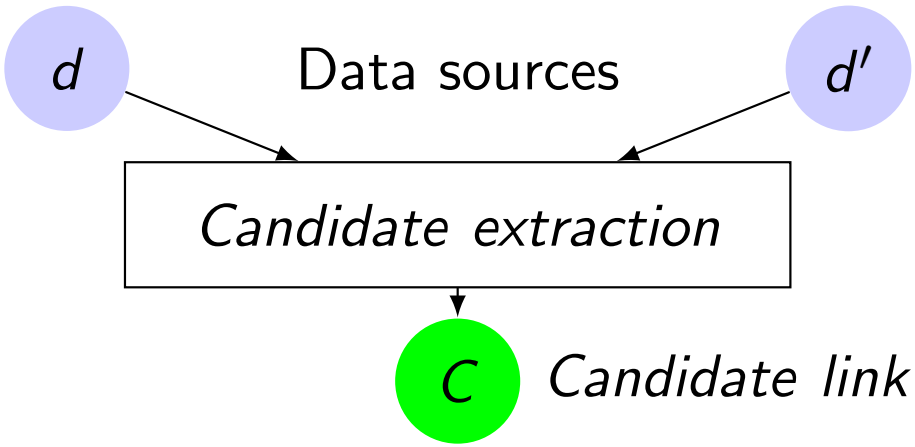
$$\mathrm{cov}(K, D, D') = \frac{|\{a : \langle a, b\rangle \in L_{D,D'}(K)\} \cup \{b : \langle a, b\rangle \in L_{D,D'}(K)\}|}{|\{a : c(a) \in D\} \cup \{b : d(b) \in D'\}|}$$

Data interlinking process

Data interlinking process



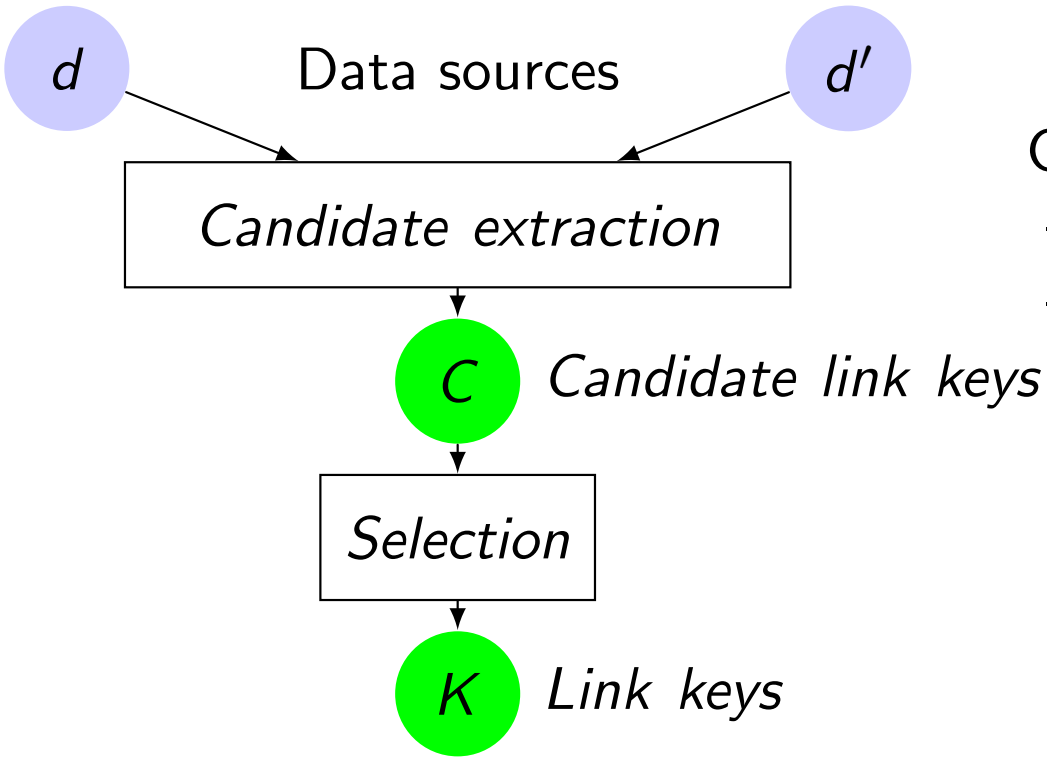
Data interlinking process



Candidate link keys:

- generate links, and
- are maximal

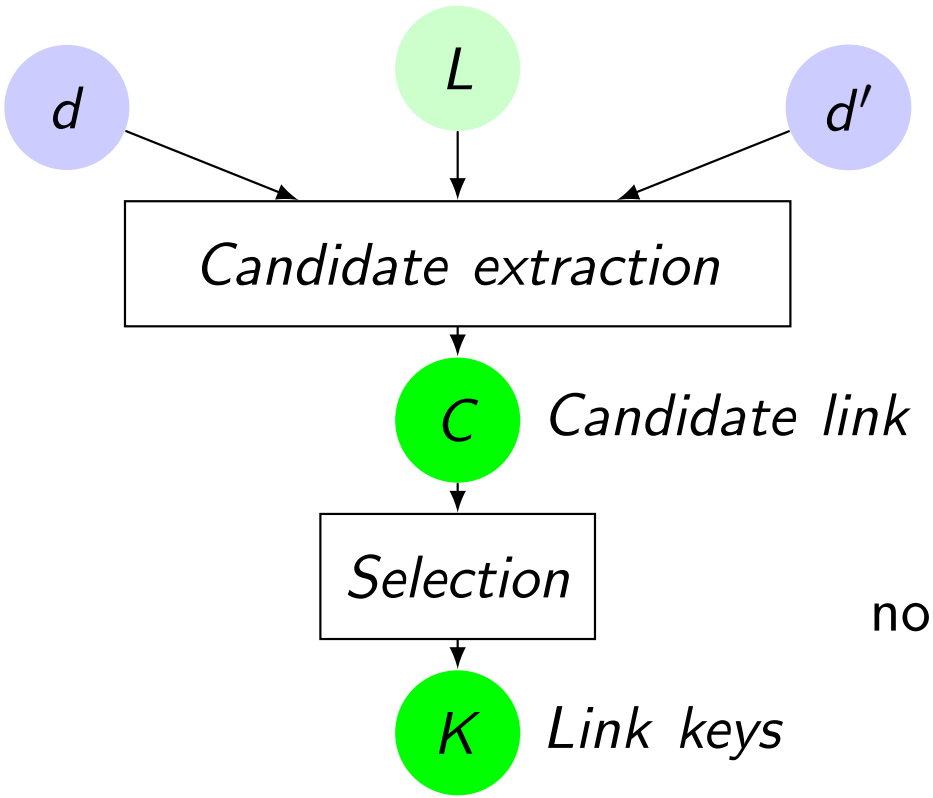
Data interlinking process



Candidate link keys:

- generate links, and
- are maximal

Data interlinking process



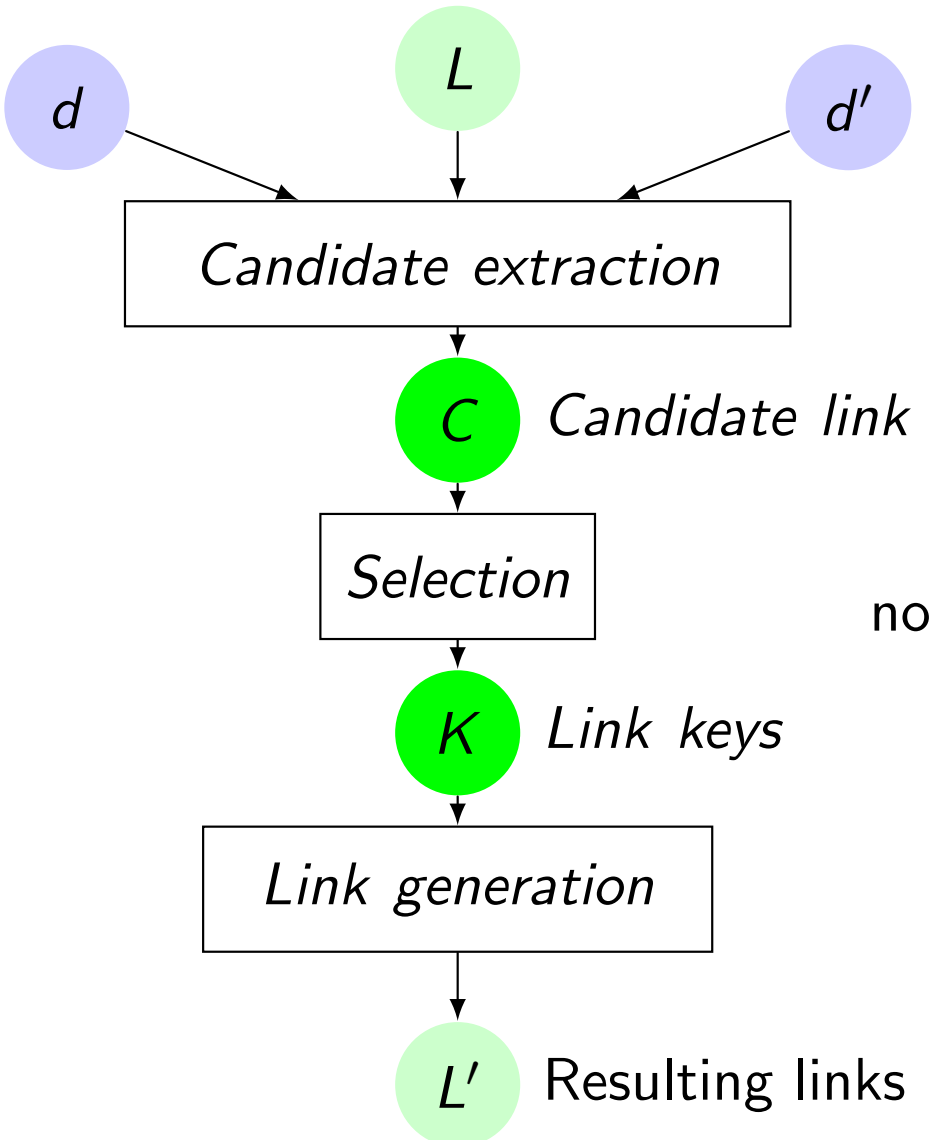
Candidate link keys:

- generate links, and
- are maximal

supervised: precision/recall

non supervised: discriminability/covera

Data interlinking process



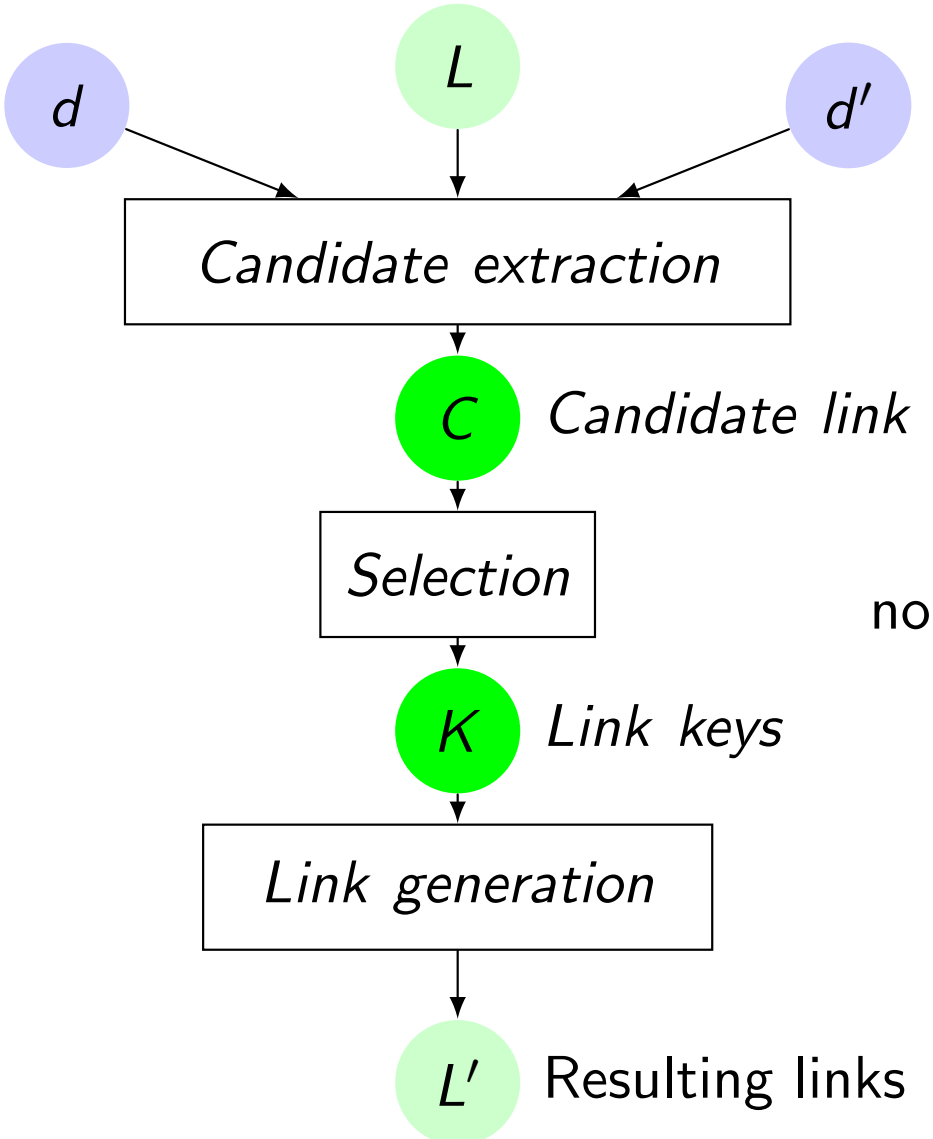
Candidate link keys:

- generate links, and
- are maximal

supervised: precision/recall

non supervised: discriminability/covera

Data interlinking process



Candidate link keys:
- generate links, and
- are maximal

supervised: precision/recall
non supervised: discriminability/covera

SPARQL queries

Datasets

Goal: find link keys between

- BnF and Abes (https://www.idref.fr : Identifiant et Référentiels pour l'enseignement supérieur et la recherche)
- Bnf and Onoma (référentiel d'acteurs intervenant dans le cycle de vie d'un bien culturel)

BnF and Abes sample data:

- 1000 most frequent homonyms
- all instances with name starting with 'a'

Onoma and BNF

ONOMA and BnF have an a priori low intersection.
We have build specific datasets between BnF and Onoma.

- Onoma: 6317 persons and 1344 groups
- 5900 are identified (Id MARQUE)
- retrieve BnF entities having same lastname and same firstname than instances from Onoma
- 962 instances from ONOMA have a correspondence in BNF
- 2604 instances from BNF have an correspondence in ONOMA

Adressed issues

First experiments outlined several issues that have been addressed:

- Heterogeneity between string literals
- Properties composition
  - in BnF Contributors are in relation with Works instances that are themselves connected to Manifestations
- Properties inversion
- Scaling with these two extensions...
- How to visualize and navigate between extracted candidate link keys

String normalization

String literals can be slightly different: "Pierre Mendès France" vs "Mendès France, Pierre".
String similarities are too costly.
We choose a basic normalization:

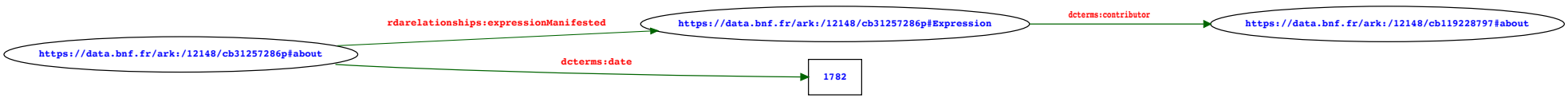
- lowercase
- remove diacritics
- tokenization based on any sequence of non alphabetical or numerical characters
- sort sequences of token

Both "Pierre Mendès France" and "Mendès France, Pierre" become ["france", "mendes", "pierre"]

Composition and Inverse

Example of compound property obtain by composition and inversion :

$\texttt{dcterms:contributor}^{-1}.\texttt{rdarelationships:expressionManifested}^{-1}.\texttt{dcterms:date}$



Some issue that we had to solve:

- It introduces a huge number of possibilities
- Many are meaningless

We have introduced:

- maximum length of compostion
- maximum expansion ratio: $soFactor_D(p) = \frac{|\{o|\langle x,p,o\rangle \in D\}|}{|\{s|\langle s,p,x\rangle \in D\}|}$

Visualization

A lot of candidate link keys are generated.

- BnF-Abes starting with 'a': 163 candidates
- BnF-Abes starting homonyms: 632 candidates
- BnF-Onoma starting homonyms: 209 candidates

A visualization prototype has been developed.

- allows to navigate from general candidates to specific ones

  (foaf:familyName, :NOM) $\rightarrow$ {(foaf:familyName, :NOM),(foaf:givenName,:PRENOM)}
- allows to sort them according to discriminability, coverage, and combination of the two
- displays generated links
- computes precision and recall evaluation if reference links are given

`http://exmo-web.inrialpes.fr/LinkexUI2`

Conclusion

- Link key extraction works :-)
- Did not discovered new rules unknown by domain experts
- Experts have been surprised by the low coverage of certain rules, for instance name, firstname, birthdate
- Evaluation also shows that it exists some duplicates or errors in data
- It can be used to discover preliminary link key without knowledge of the datasets

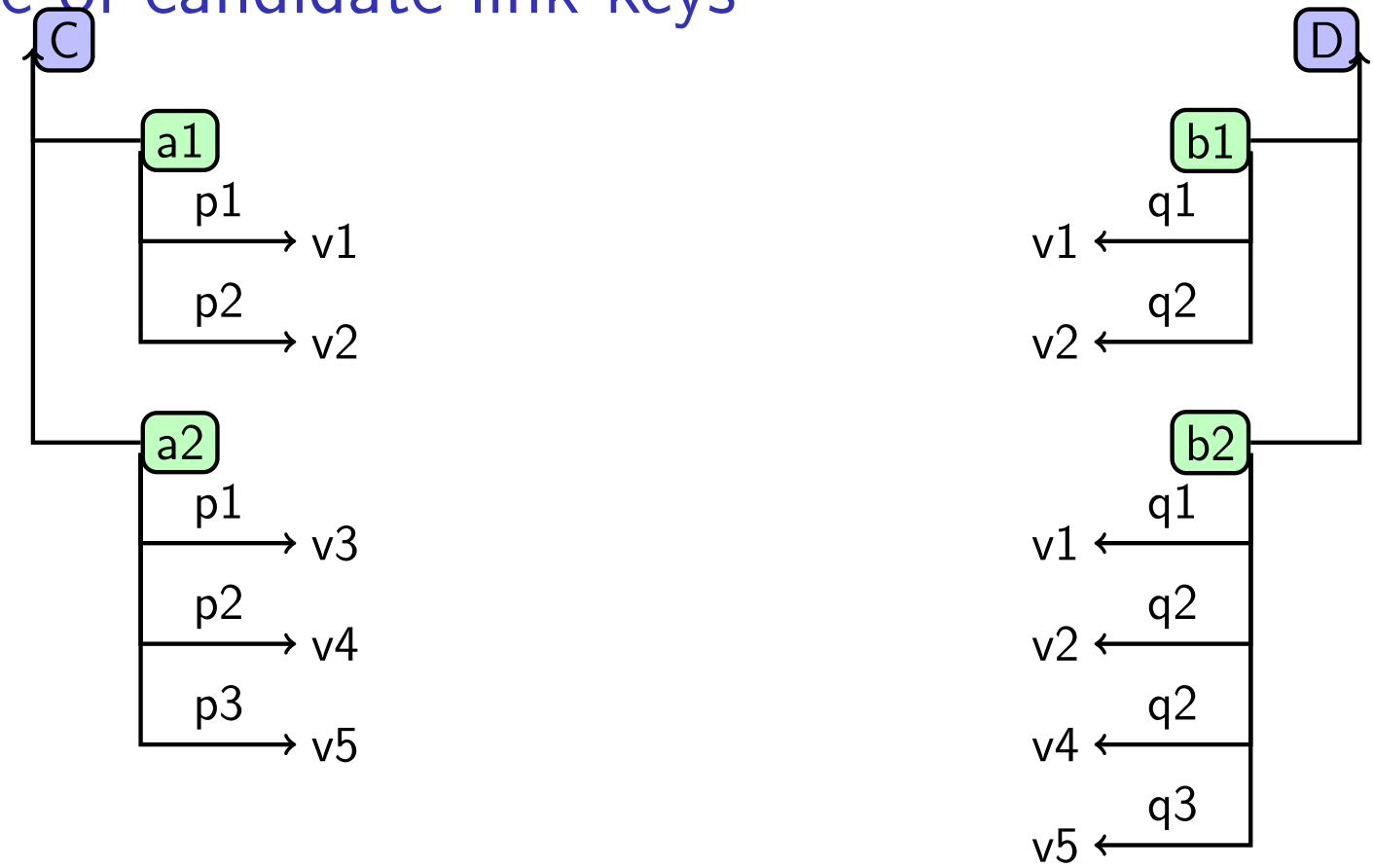
Perspectives / Open issues

- automatic sampling of large datasets
- more robust/ adaptable normalization (e.g. date normalization with different levels of granularity)
- automatic selection of subset of link keys

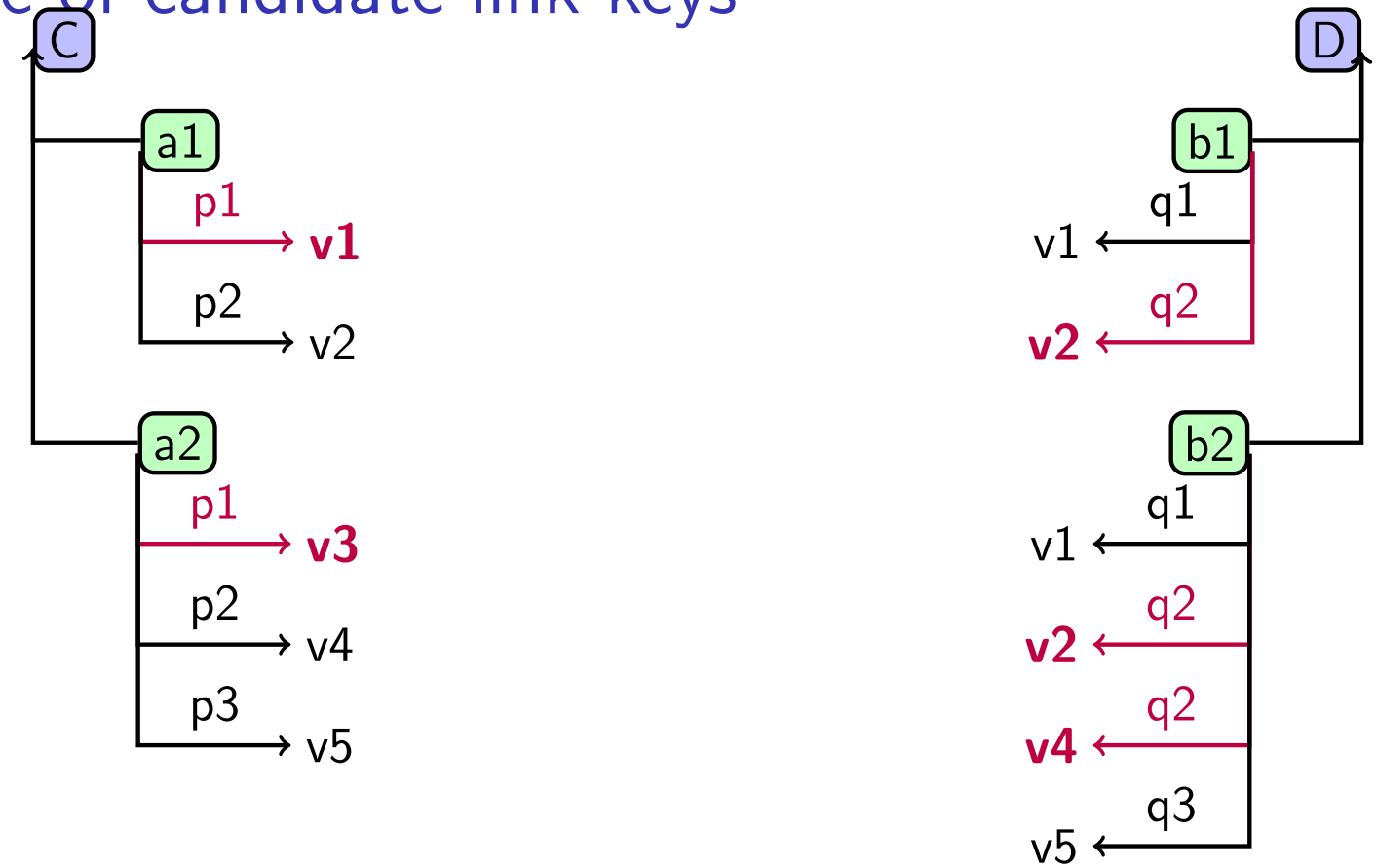
Questions?

Jerome . David @ inria . fr
Aude . Le-Moullec-Rieu @ bnf . fr

Example of candidate link keys

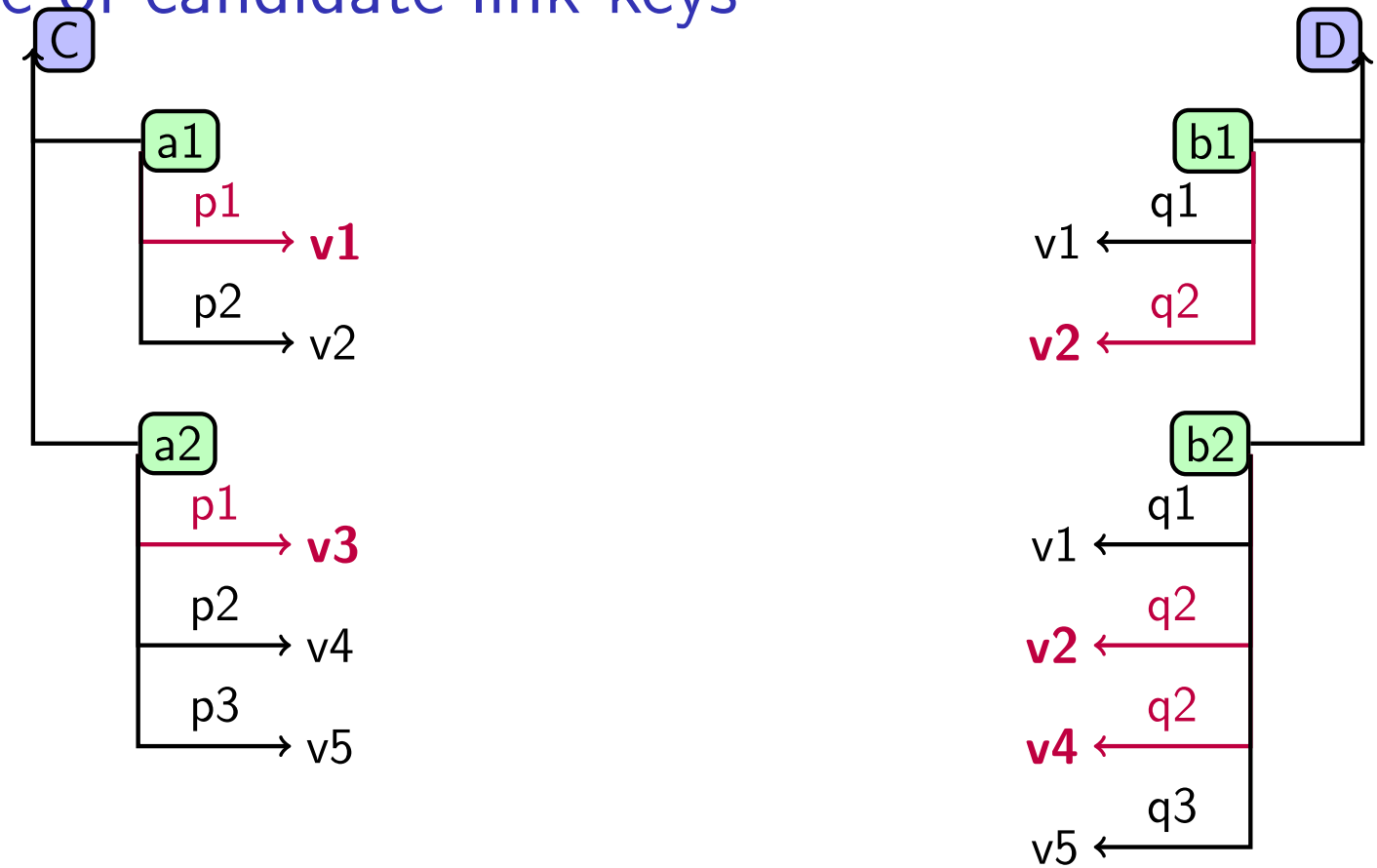


Example of candidate link keys



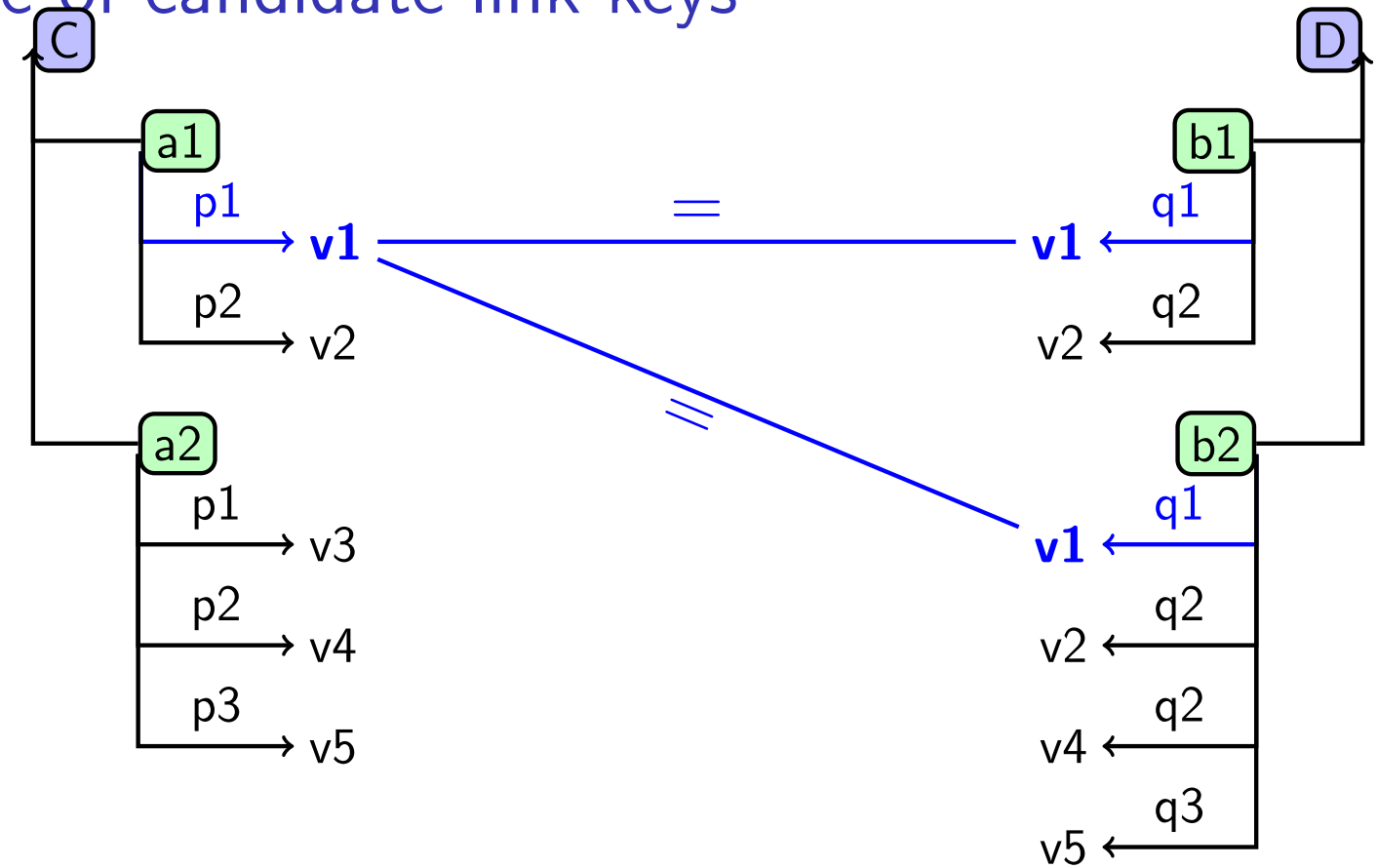
- $\{\langle p_1, q_2 \rangle\}$ a candidate?

Example of candidate link keys



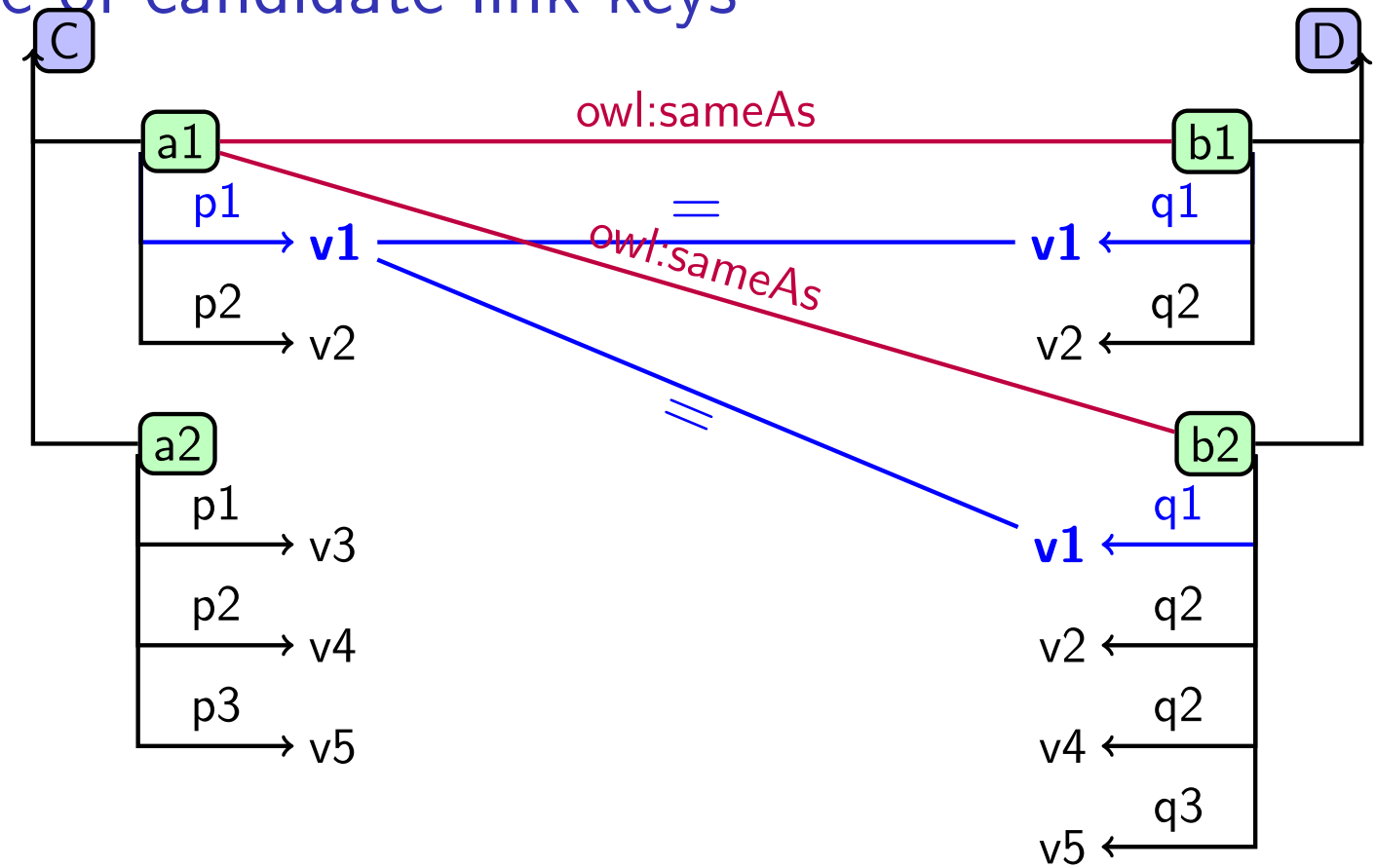
- $\{\langle p_1, q_2 \rangle\}$ a candidate? NO, it does not generate any link

Example of candidate link keys



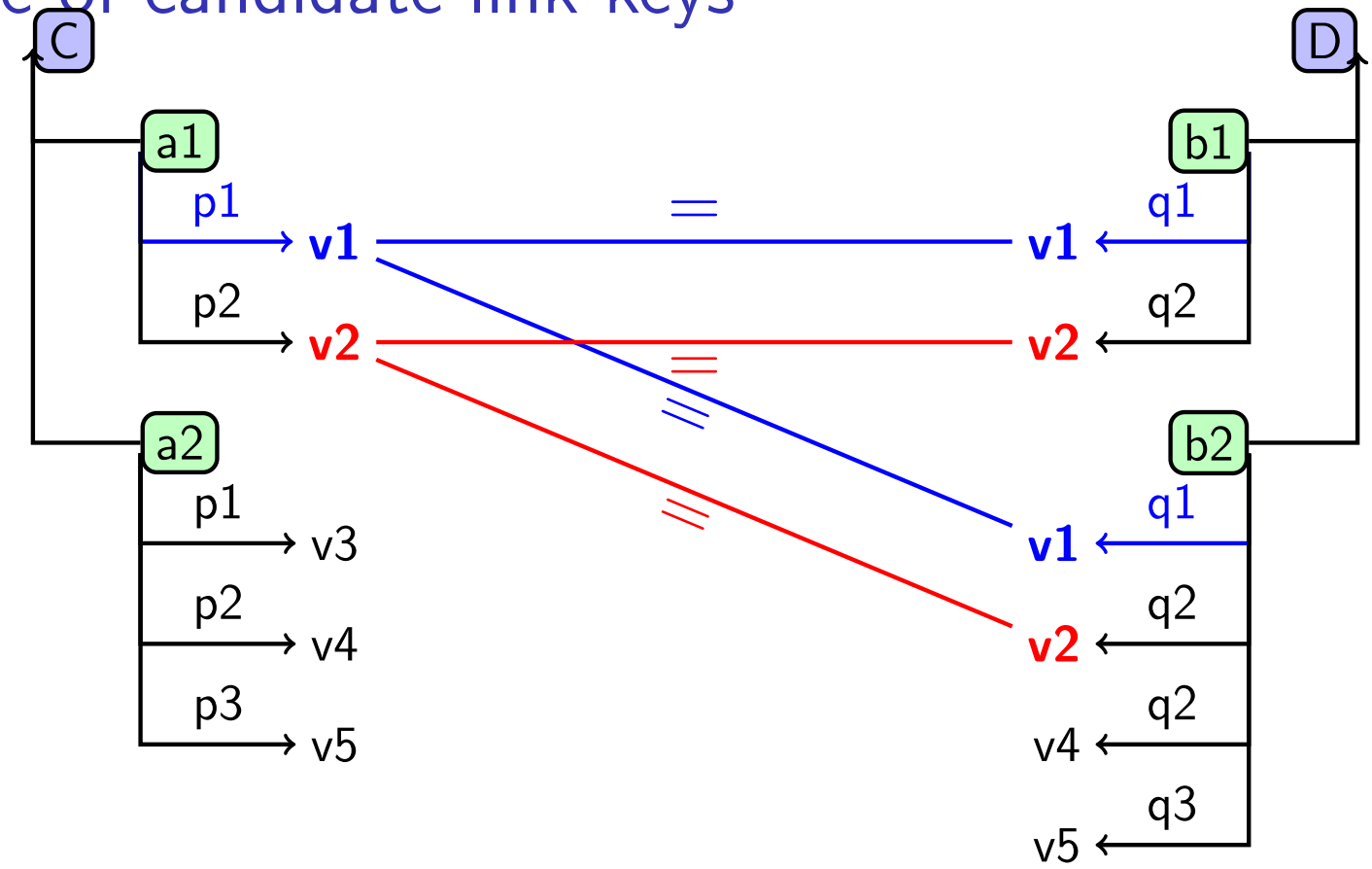
- $\{\langle p_1, q_2 \rangle\}$ a candidate? NO, it does not generate any link
- $\{\langle p_1, q_1 \rangle\}$ a candidate?

Example of candidate link keys



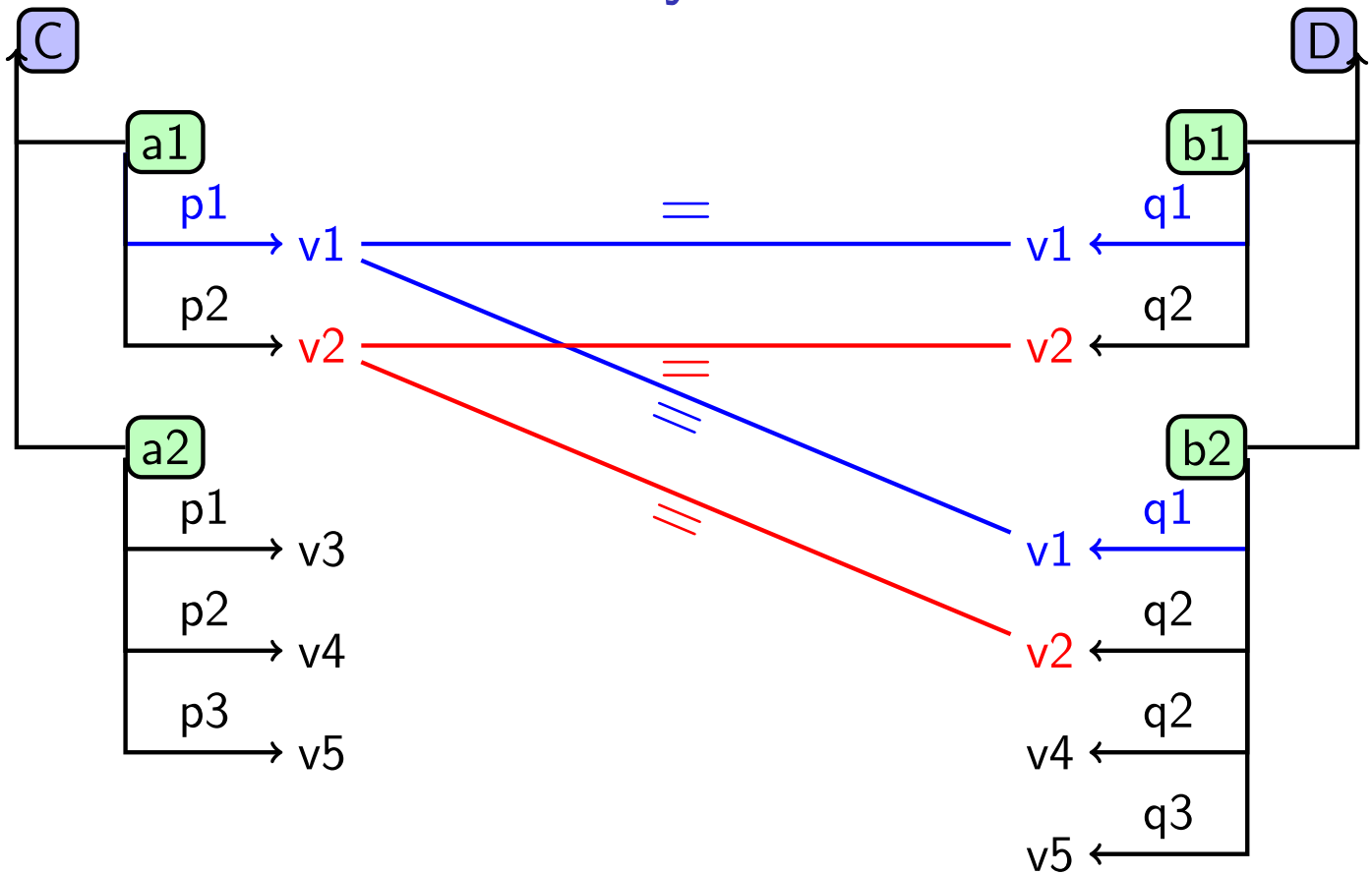
- $\{\langle p_1, q_2\rangle\}$ a candidate? NO, it does not generate any link
- $\{\langle p_1, q_1\rangle\}$ a candidate?
  - it could generate links: $\langle a_1, b_1\rangle$ and $\langle a_1, b_2\rangle$

Example of candidate link keys



- $\{\langle p_1, q_2\rangle\}$ a candidate? NO, it does not generate any link
- $\{\langle p_1, q_1\rangle\}$ a candidate? NO
  - it could generate links: $\langle a_1, b_1\rangle$ and $\langle a_1, b_2\rangle$
  - but it is not maximal: each link also shares $\{\langle p_2, q_2\rangle\}$

Example of candidate link keys



- $\{\langle p_1, q_2\rangle\}$ a candidate? NO, it does not generate any link
- $\{\langle p_1, q_1\rangle\}$ a candidate? NO
- Then $\{\langle p_1, q_1\rangle, \langle p_2, q_2\rangle\}$ is a candidate linkkey

Algorithm for candidate link key extraction

1. For each dataset, index each subject-property pair according to its values

indexDataset(D)	indexDataset(D')
$v_1 : \{\langle a_1, p_1\rangle\}$	$v_1 : \{\langle b_1, q_1\rangle, \langle b_2, q_1\rangle\}$
$v_2 : \{\langle a_1, p_2\rangle\}$	$v_2 : \{\langle b_1, q_2\rangle, \langle b_2, q_2\rangle\}$
$v_3 : \{\langle a_2, p_1\rangle\}$	
$v_4 : \{\langle a_2, p_2\rangle\}$	$v_4 : \{\langle b_2, q_2\rangle\}$
$v_5 : \{\langle a_2, p_3\rangle\}$	$v_5 : \{\langle b_2, q_3\rangle\}$

2. Iterate on index and compute for each pair of subjects the maximal set of pair of property on which they agree

Candidate links		Candidate link keys
$\langle a_1, b_1\rangle$	$\rightarrow$	$\{\langle p_1, q_1\rangle, \langle p_2, q_2\rangle\}$
$\langle a_1, b_2\rangle$	$\rightarrow$	$\{\langle p_1, q_1\rangle, \langle p_2, q_2\rangle\}$
$\langle a_2, b_1\rangle$	$\rightarrow$	$\emptyset$
$\langle a_2, b_2\rangle$	$\rightarrow$	$\{\langle p_2, q_2\rangle, \langle p_3, q_3\rangle\}$

3. Close by intersection

Resulting candidate link keys

$\boxed{\{\langle p_1, q_1\rangle, \langle p_2, q_2\rangle\}}$ $\boxed{\{\langle p_2, q_2\rangle, \langle p_3, q_3\rangle\}}$

$\boxed{\emptyset}$

Resulting candidate link keys

$\boxed{\{\langle p_1, q_1\rangle, \langle p_2, q_2\rangle\}}$ $\boxed{\{\langle p_2, q_2\rangle, \langle p_3, q_3\rangle\}}$

$\boxed{\{\langle p_2, q_2\rangle\}}$

$\boxed{\emptyset}$

Resulting candidate link keys