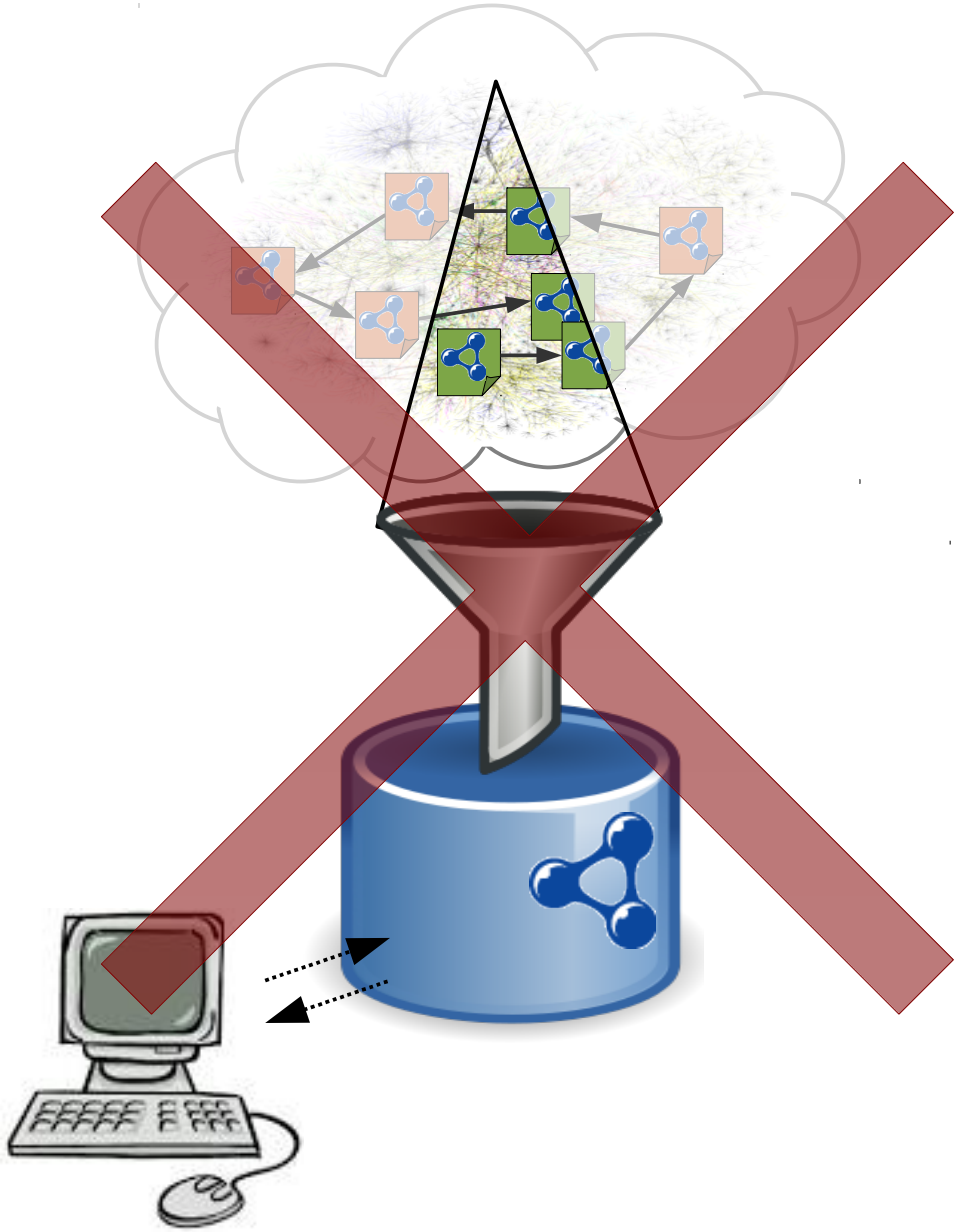
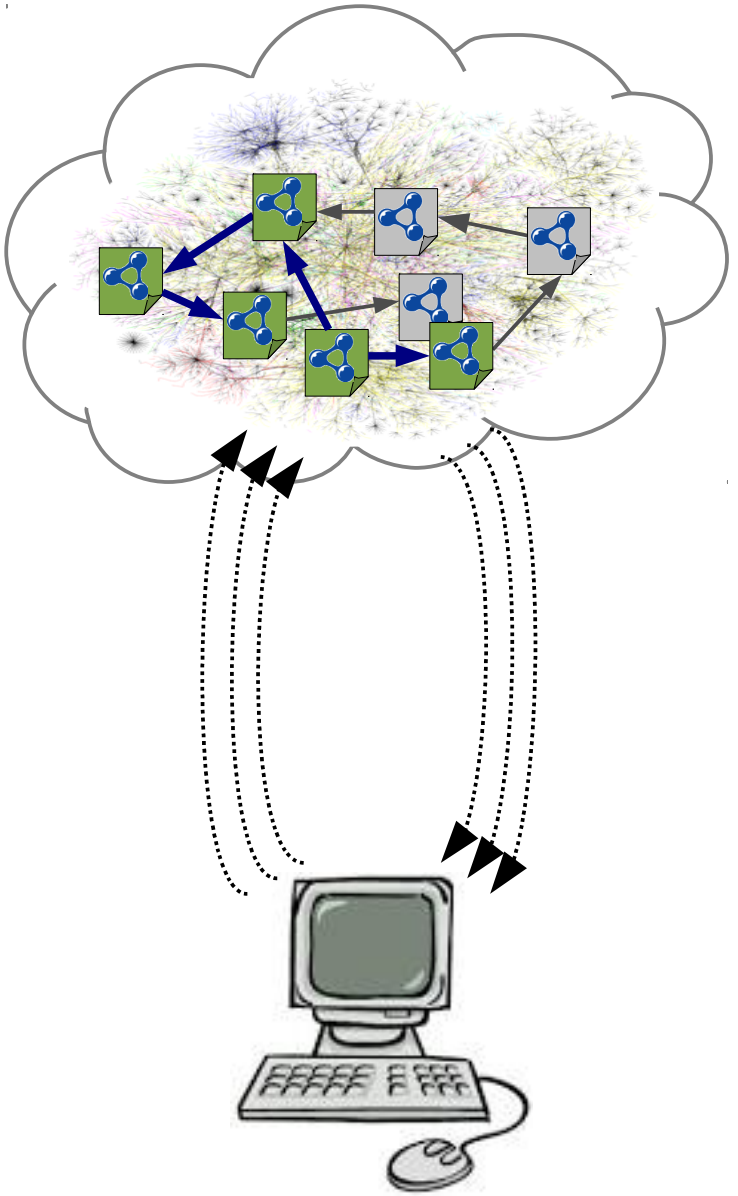
# LDQL: A Query Language for the Web of Linked Data

**Olaf Hartig**[1] and **Jorge Pérez**[2]

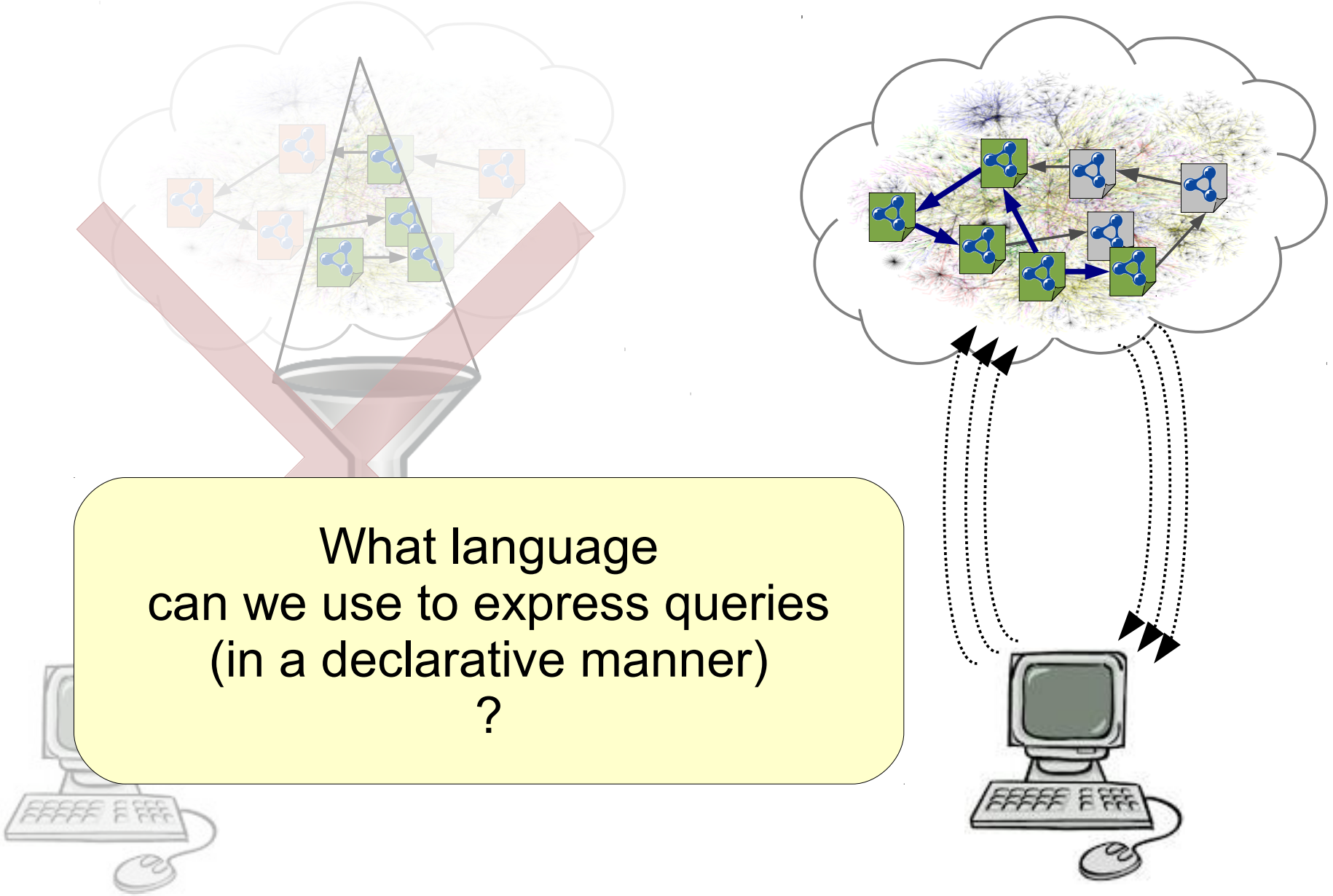
[1]Hasso Plattner Institute, University of Potsdam, Germany
[2]Department of Computer Science, Universidad de Chile

# Linked Data Query Processing





# Research Question



What language
can we use to express queries
(in a declarative manner)
?

# Existing Proposals 1/2

- Basic Graph Patterns
  - *Bounded* method [Bouquet, Ghidini, Serafini 2009]
  - *Navigational* method [Bouquet, Ghidini, Serafini 2009]
  - *Direct access* method [Bouquet, Ghidini, Serafini 2009]
  - *Matching-based* query semantics [Hartig 2011]
  - Family of *authoritativeness-based* query semantics [Harth and Speiser 2012]
  - Different *inference-based* query semantics [Umbrich et al. 2012]

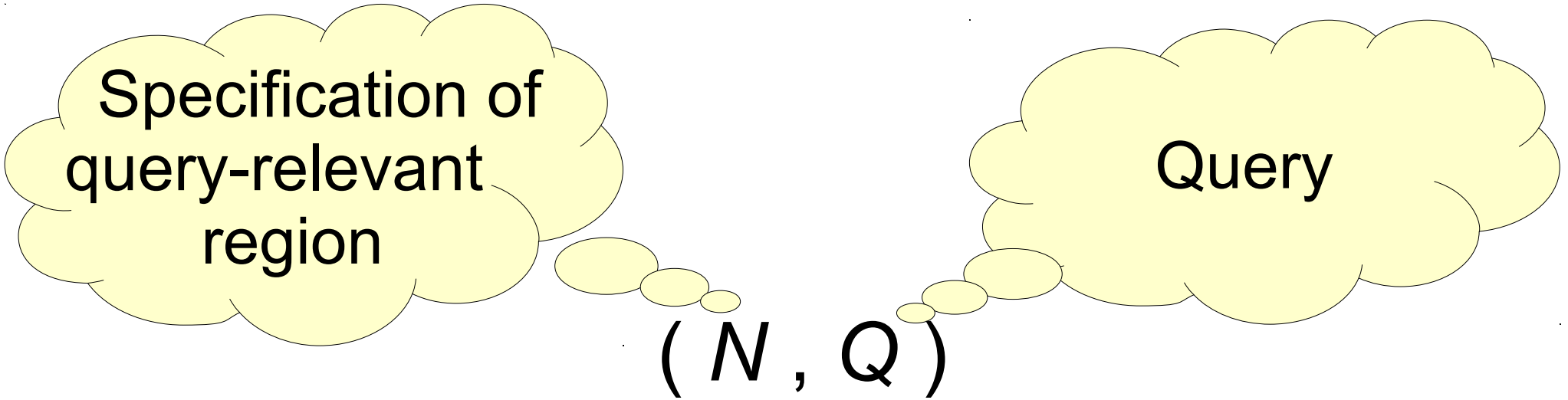
# Existing Proposals 2/2

- SPARQL 1.0
  - *Full-Web* query semantics [Hartig 2012]
  - Family of *reachability-based* semantics [Hartig 2012]
    - $c_{\text{ALL}}$-semantics, $c_{\text{NONE}}$-semantics, $c_{\text{MATCH}}$-semantics, …
- SPARQL 1.1 Property Paths Patterns
  - Context-based query semantics [Hartig and Pirrò 2015]
- NautiLOD [Fionda, Gutierrez, and Pirrò 2012]
- LDPath (no formal semantics) [Schaffert et al. 2012]

# Limitation of Existing Proposals

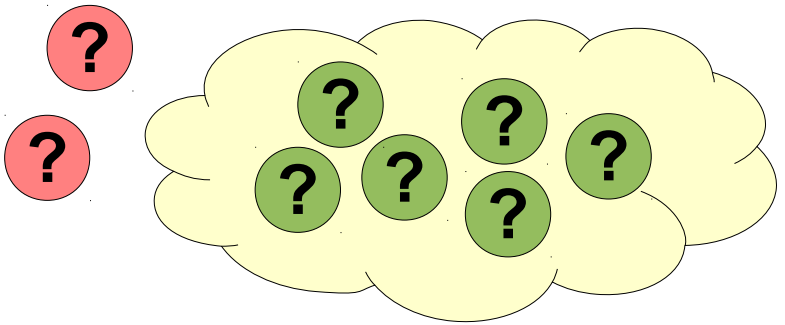
- No separation of the following two tasks:
  - Select query-relevant regions of the Web of Linked Data
  - Evaluate pattern over the data in the selected regions
- For instance, impossible to express queries such as:
  - Take all Linked Data reached by following two `foaf:knows` links starting from URI *u*, and evaluate a given SPARQL pattern over the union of this data.

# General Idea of LDQL



# Our Contributions

- Formal syntax and semantics of LDQL
  - Including some rewrite rules (i.e., equivalences)
- Study Web safeness of LDQL queries
  - Show a decidable syntactic property of LDQL queries for which a complete execution is feasible in practice
- Compare LDQL with existing proposals
  - Show that LDQL is strictly more expressive

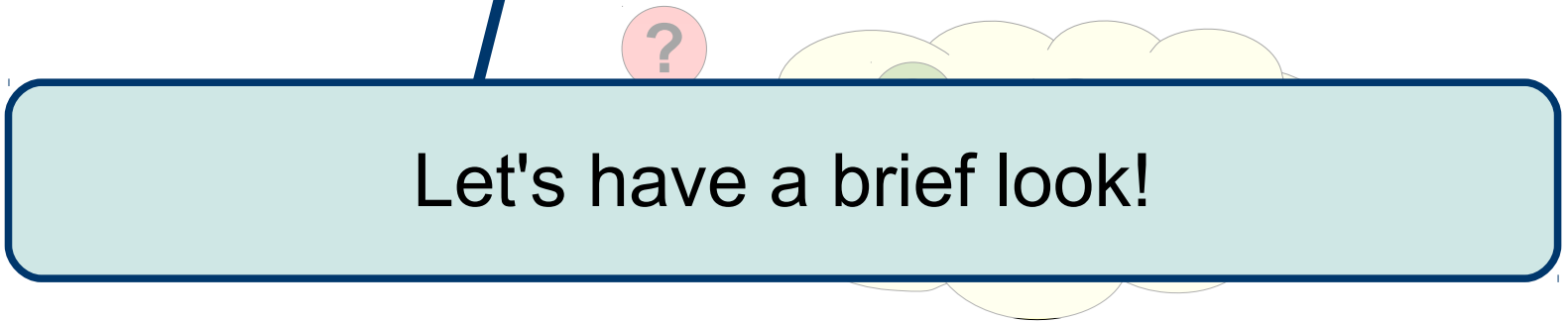


# Existing Proposals 2/2

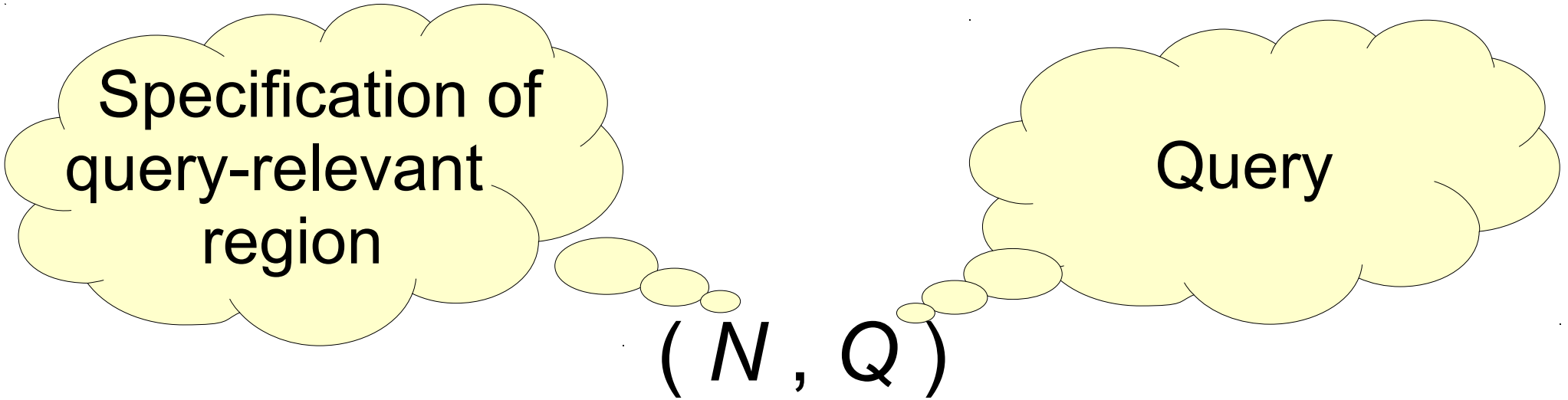
- SPARQL 1.0
  - *Full-Web* query semantics [Hartig 2012]
  - Family of *reachability-based* semantics [Hartig 2012]
    - $c_{ALL}$-semantics ✓ $c_{NONE}$-semantics ✓ $c_{MATCH}$-semantics ✓ …
- SPARQL 1.1 Property Paths Patterns
  - Context-based query semantics ✓ [Hartig and Pirrò 2015]
- NautiLOD ✓ [Fionda, Gutierrez, and Pirrò 2012]
- LDPath (no formal semantics) [Schaffert et al. 2012]

# Our Contributions

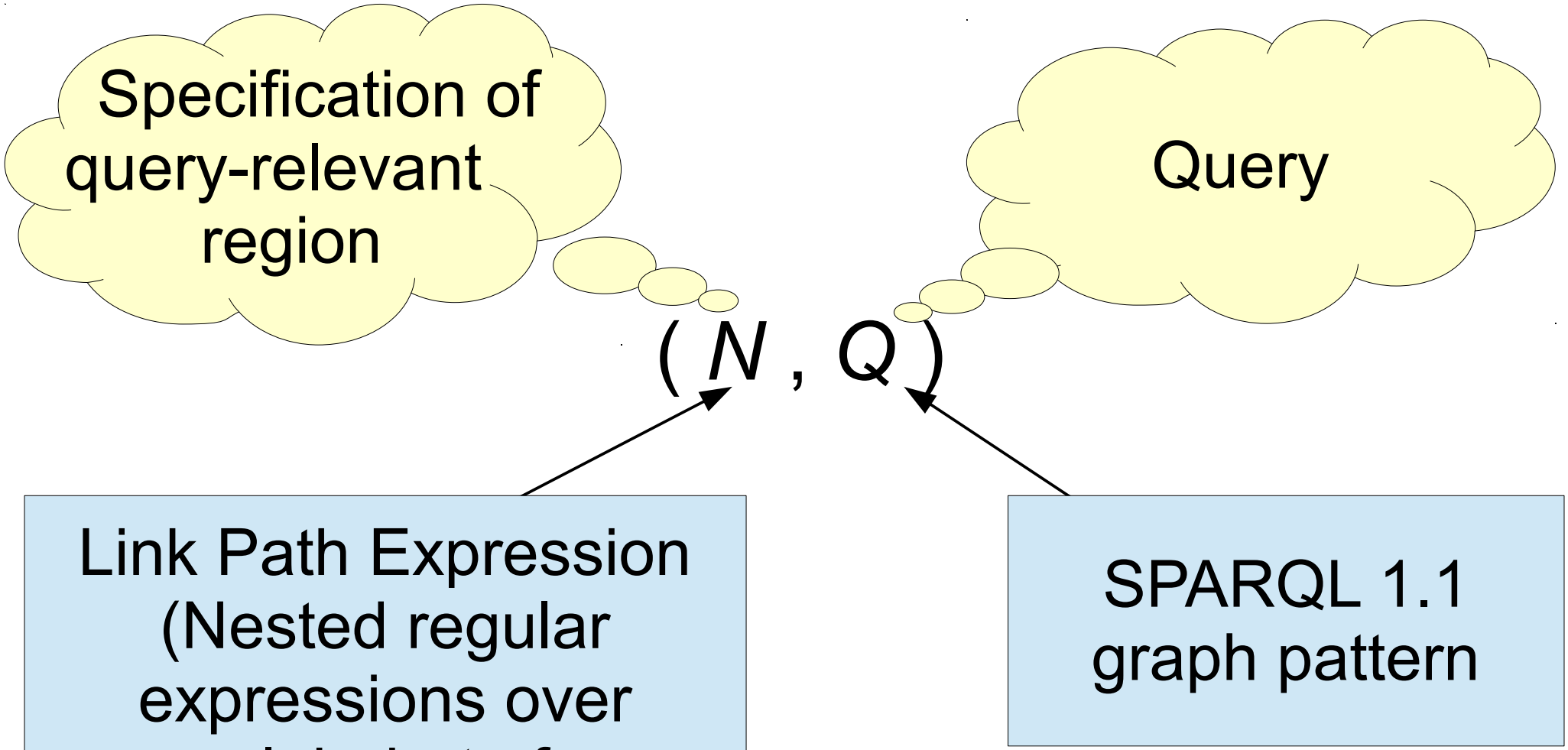
- Formal syntax and semantics of LDQL
  - Including some rewrite rules (i.e., equivalences)
- Study Web safeness of LDQL queries
  - Show a decidable syntactic property of LDQL queries for which a complete execution is feasible in practice
- Compare LDQL with existing proposals
  - Show that LDQL is strictly more expressive



# General Idea of LDQL



# General Idea of LDQL



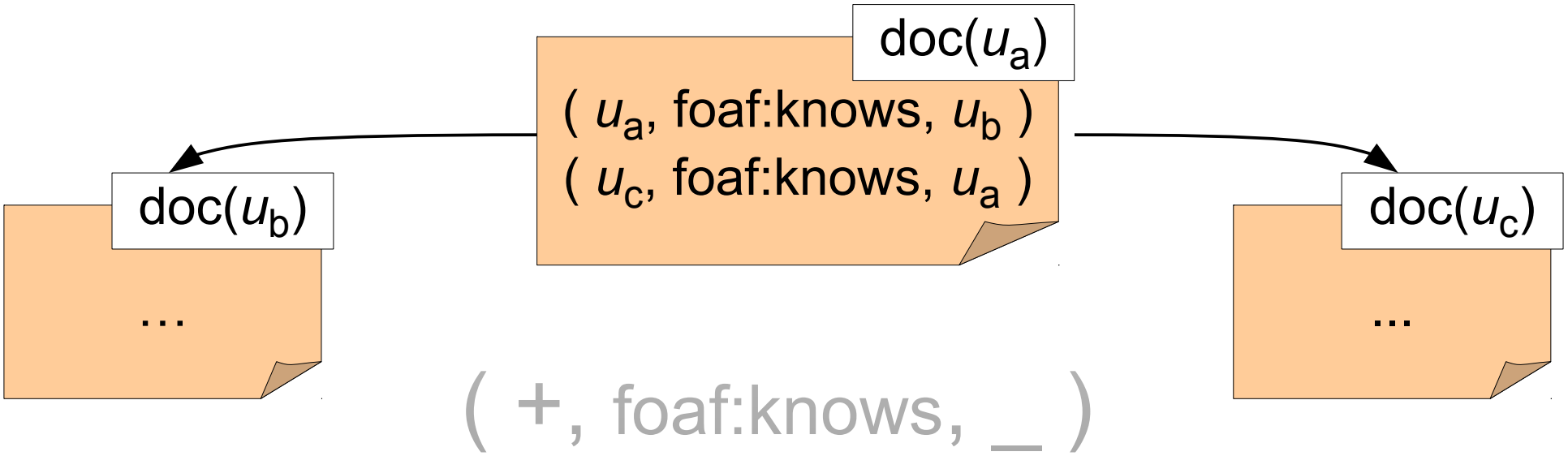
# Link Patterns

$$( \beta_1, \beta_2, \beta_3 )$$

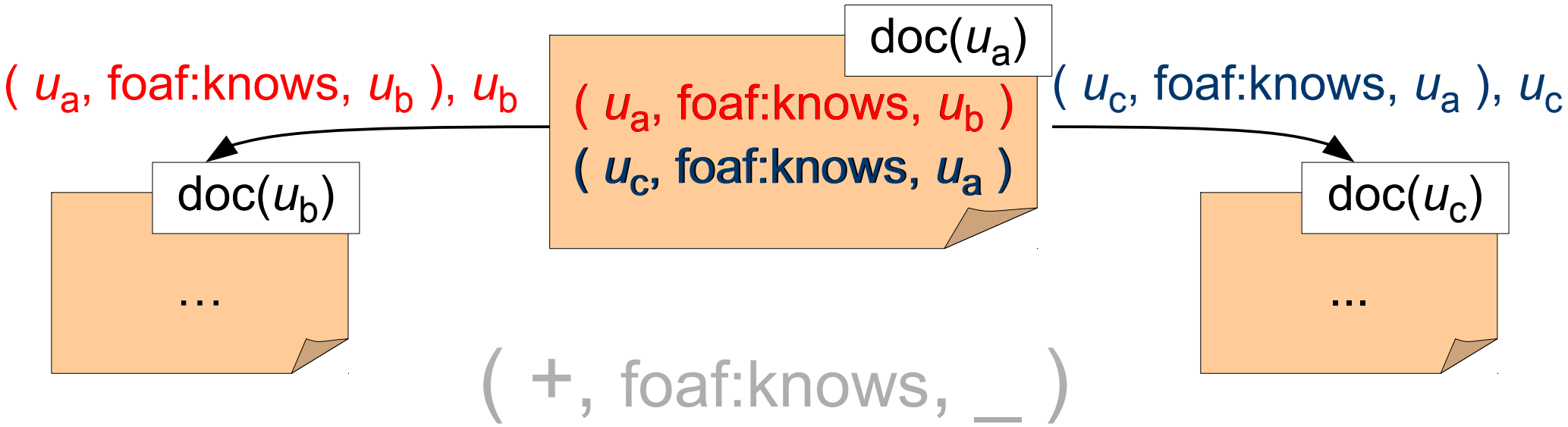
- Each $\beta_i$ is either:
  - a URI, or
  - a wildcard denoted by _, or
  - a placeholder for the context URI denoted by +
- $\beta_3$ may also be a literal
- Example:

$$( +, \text{foaf:knows}, \_ )$$

# Link Graph


doc($u_a$)
( $u_a$, foaf:knows, $u_b$ )
( $u_c$, foaf:knows, $u_a$ )
doc($u_b$)
...
doc($u_c$)
...
( +, foaf:knows, _ )

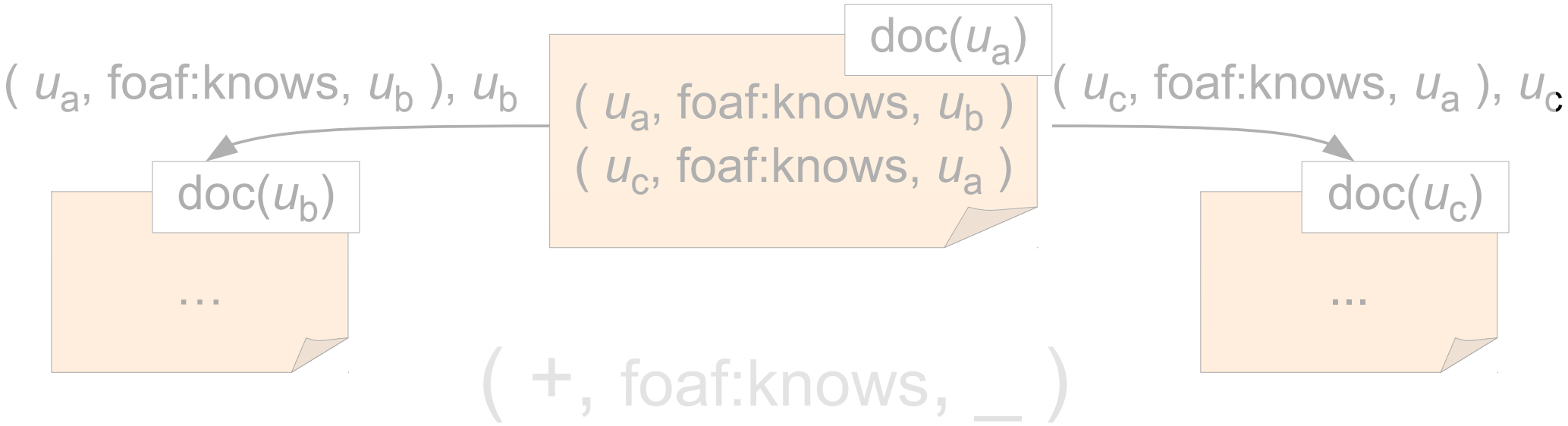
# Link Graph


( $u_a$, foaf:knows, $u_b$ ), $u_b$
doc($u_a$)
( $u_a$, foaf:knows, $u_b$ )
( $u_c$, foaf:knows, $u_a$ )
( $u_c$, foaf:knows, $u_a$ ), $u_c$
doc($u_b$)
...
doc($u_c$)
...
( +, foaf:knows, _ )

# Link Pattern Matching

A link graph edge labeled with RDF triple ($x_1$, $x_2$, $x_3$) and URI $u$ matches a link pattern ($\beta_1$, $\beta_2$, $\beta_3$) in the context of URI $u_{ctx}$ if:

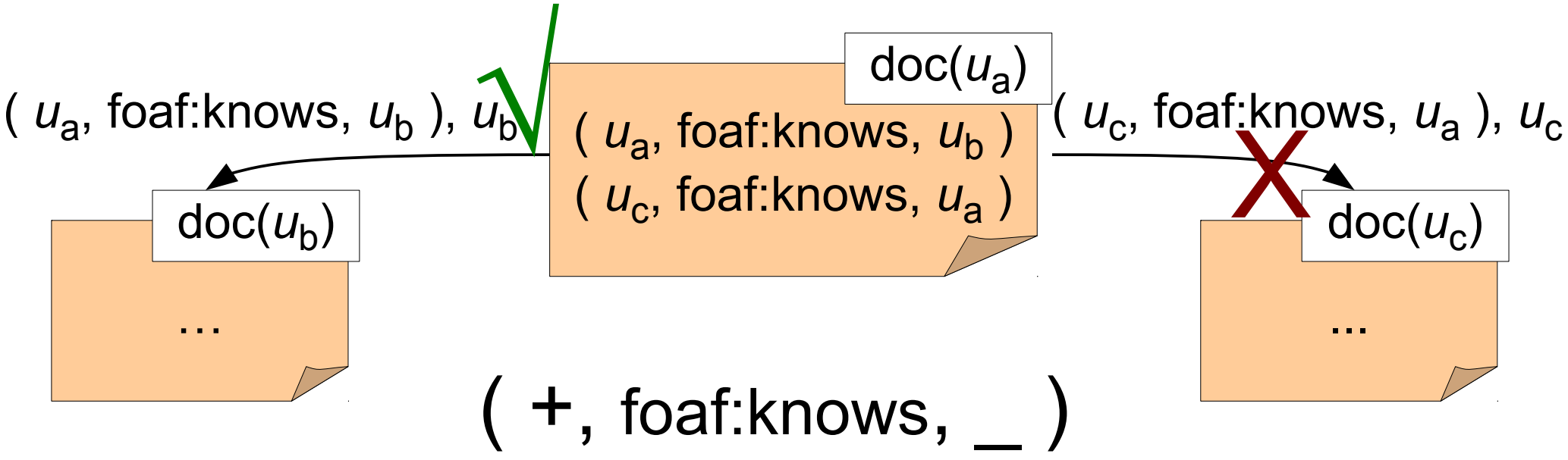
- $\exists\ i \in \{1,2,3\}$ such that $\beta_i$ = _ and $x_i = u$, and
- $\forall\ i \in \{1,2,3\}$ (i) $\beta_i = x_i$ , (ii) $\beta_i$ = _ , or (iii) $\beta_i$ = + and $x_i = u_{ctx}$



# Link Pattern Matching

A link graph edge labeled with RDF triple ($x_1$, $x_2$, $x_3$) and URI $u$ matches a link pattern ($\beta_1$, $\beta_2$, $\beta_3$) in the context of URI $u_{ctx}$ if:

- $\exists\ i \in \{1,2,3\}$ such that $\beta_i$ = _ and $x_i = u$, and
- $\forall\ i \in \{1,2,3\}$ (i) $\beta_i = x_i$ , (ii) $\beta_i$ = _ , or (iii) $\beta_i$ = + and $x_i = u_{ctx}$



# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

$(q \text{ AND } q)$ |

$(q \text{ UNION } q)$ |

$\pi_V q$ | ($V$ a set of variables)

(SEED $U$ $q$) | ($U$ a set of URIs)

(SEED $?v$ $q$) ($?v$ a variable)

$lpe := \varepsilon \mid lp \mid lpe/lpe \mid lpe|lpe \mid lpe^* \mid [lpe] \mid (?v, q)$

# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

$(q$ AND $q)$ |

Example: Take all Linked Data reached by following two foaf:knows links starting from URI $u$, and evaluate SPARQL pattern $P$ over the union of this data.

( (+,foaf:knows, _) /(+,foaf:knows, _) , $P$ )

with seeds $S = \{u\}$

(SEED $?v$ $q$) ($?v$ a variable)

$lpe$ := $\varepsilon$ | $lp$ | $lpe/lpe$ | $lpe|lpe$ | $lpe^*$ | $[lpe]$ | $(?v, q)$

# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

$(q$ AND $q)$ |

Example: Take all Linked Data reached by following two foaf:knows links starting from URI $u$, and evaluate SPARQL pattern $P$ over the union of this data.

( (+,foaf:knows, _) /(+,foaf:knows, _) , $P$ )

with seeds $S = \{u\}$

(SEED $?v$ $q$) ($?v$ a variable)

$lpe$ := $\varepsilon$ | $lp$ | $lpe/lpe$ | $lpe|lpe$ | $lpe^*$ | $[lpe]$ | $(?v, q)$

# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

$(q$ AND $q)$ |

Example: Take all Linked Data reached by following two foaf:knows links starting from URI $u$, and evaluate SPARQL pattern $P$ over the union of this data.

( (+,foaf:knows, _) /(+,foaf:knows, _) , $P$ )

with seeds $S = \{u\}$

(SEED $?v$ $q$) ($?v$ a variable)

$lpe$ := $\varepsilon$ | $lp$ | $lpe/lpe$ | $lpe|lpe$ | $lpe^*$ | $[lpe]$ | $(?v, q)$

# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

For every LDQL query $q'$,
there exists a semantically equivalent LDQL query $q''$
such that
every LPE in $q''$ consists only of $\varepsilon$ and $lpe^*$ and $(?v, q)$.

$lpe$ := $\varepsilon$ | $lp$ | $lpe/lpe$ | $lpe|lpe$ | $lpe^*$ | $[lpe]$ | $(?v, q)$

# (Algebraic) Syntax of LDQL

$q$ := $(lpe, P)$ | ($lpe$ an LPE, $P$ a SPARQL pattern)

$(q$ AND $q)$ |

$(q$ UNION $q)$ |

$\pi_V q$ | ($V$ a set of variables)

(SEED $U$ $q$) | ($U$ a set of URIs)

(SEED $?v$ $q$) ($?v$ a variable)

$lpe$ := $\varepsilon$ | $lp$ | $lpe/lpe$ | $lpe|lpe$ | $lpe^*$ | $[lpe]$ | $(?v, q)$

Example: ( ($lpe_1$ , $P_1$) AND (SEED $?x$ $q_2$) )

where $P_1$ is: ( ($?x$,p,$?y$) FILTER ($?y$ > 3))

# Summary

- Query language for the Web of Linked Data
- Main feature: separation of ...

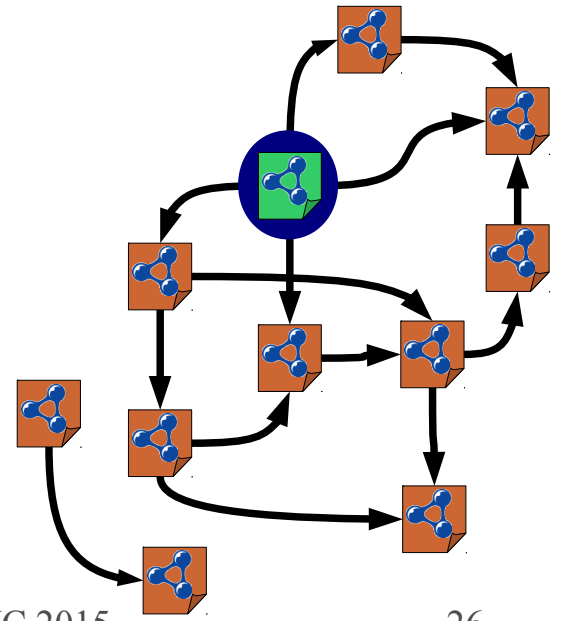
  … navigational components to select regions of the Web of Linked Data, and

  … query patterns to be matched in the data of the selected regions
- In the paper we study LDQL
  - Web-safeness property
  - LDQL is strictly more expressive than the major existing proposals

# Backup Slides

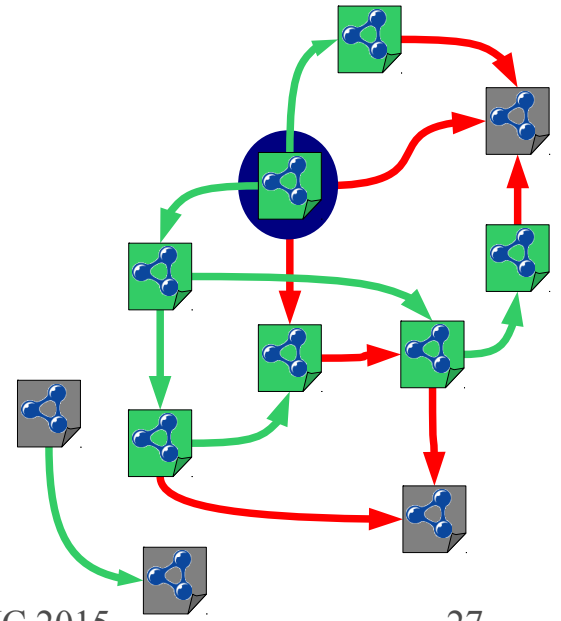
# Reachability-Based Semantics

- Restrict scope of evaluating a SPARQL pattern ***P*** over a Web of Linked Data ***W*** to a ***reachable subweb*** of ***W***
  - Seed documents ***S***,



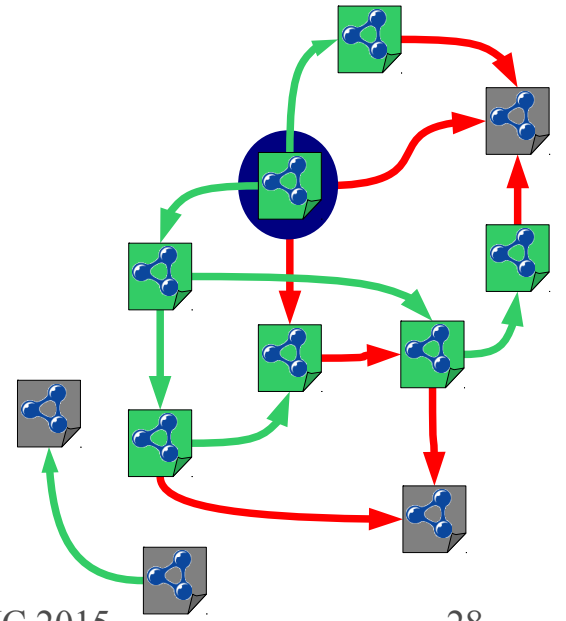
# Reachability-Based Semantics

- Restrict scope of evaluating a SPARQL pattern ***P*** over a Web of Linked Data ***W*** to a ***reachable subweb*** of ***W***
  - Seed documents ***S***, and
  - Documents along paths of data links that satisfy a reachability criterion ***c*** (e.g., $c_{all}$, $c_{none}$, $c_{match}$)



# Reachability-Based Semantics

- Restrict scope of evaluating a SPARQL pattern ***P*** over a Web of Linked Data ***W*** to a ***reachable subweb*** of ***W***
  - Seed documents ***S***, and
  - Documents along paths of data links that satisfy a reachability criterion ***c*** (e.g., $c_{all}$, $c_{none}$, $c_{match}$)
- $eval_c(\ P, S, W\ ) = eval(\ P, G'\ )$
  - ***G'*** is the union of the RDF triples in all docs in the **reachable subweb** of ***W***



# NautiLOD

- Regular expressions over URIs that are predicates
- Each step has to match in the current context document
  - Base case: eval($p$,$u_{cxt}$,$\boldsymbol{W}$) = { $u'$ | ($u_{cxt}$, $p$, $u'$) in $D_{\boldsymbol{W}}(u_{cxt})$ }
- Example: $p_1$ / $p_2$
- Additional features:
  - Test based on SPARQL ASK queries
  - Actions

# Web Safeness

An LDQL query $q$ is Web-safe if there exists an algorithm that, for any Web of Linked Data $W$ = ($D$, $adoc$) and any finite set $S$ of URIs, computes eval($q$, $S$, $W$) by looking up a finite number of URIs without assuming an a priori availability of any information about $D$ and dom($adoc$).