

Ontology-Mediated Query Answering

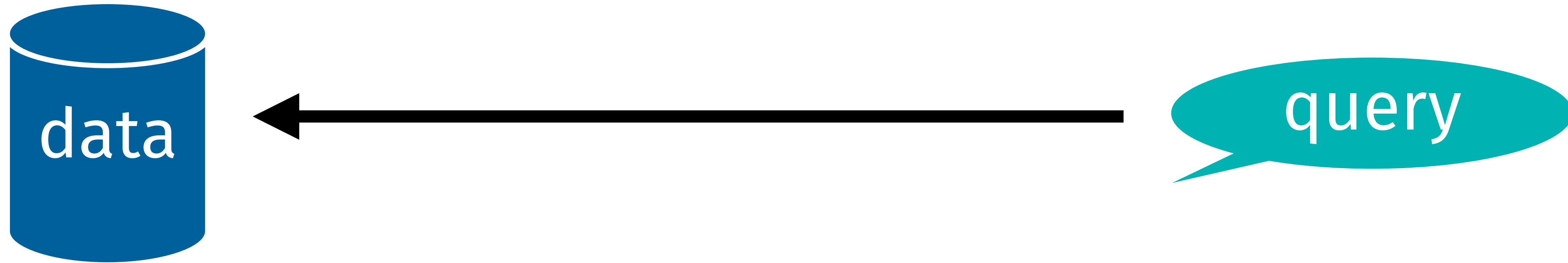
Using Rules and Reasoning to Get More from Data

Meghyn Bienvenu

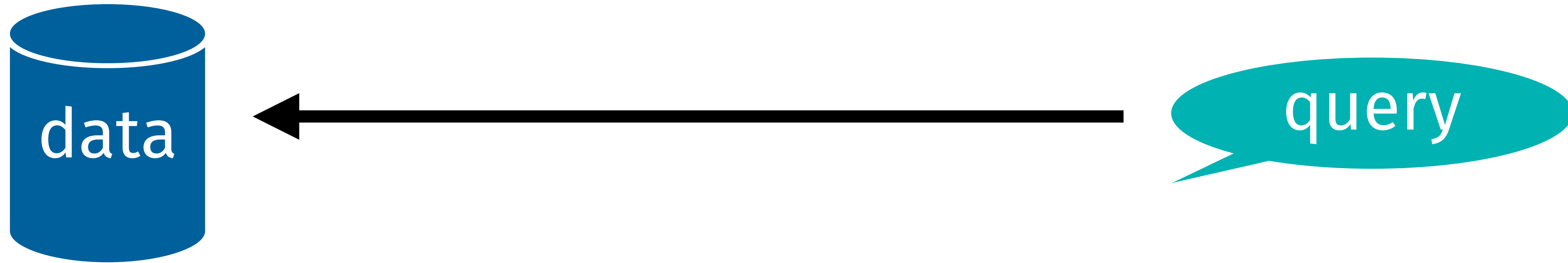
*Laboratoire Bordelais de Recherche
en Informatique (LaBRI)*

Introduction

Challenges of querying data

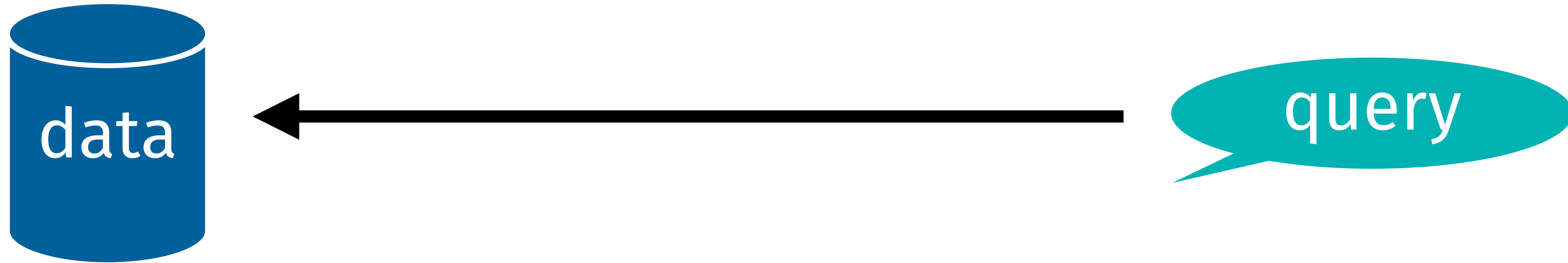


Challenges of querying data



Why is it **hard** to query data?

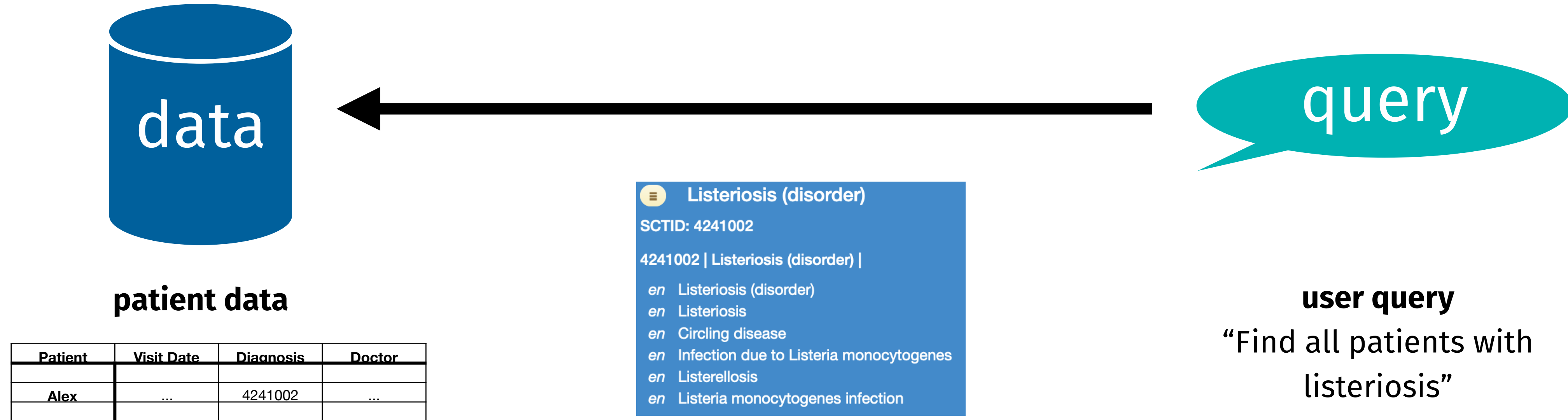
Challenges of querying data



Why is it hard to query data?

- Need to know **how database(s) are structured**
- Can be **many ways** to **express the same thing**

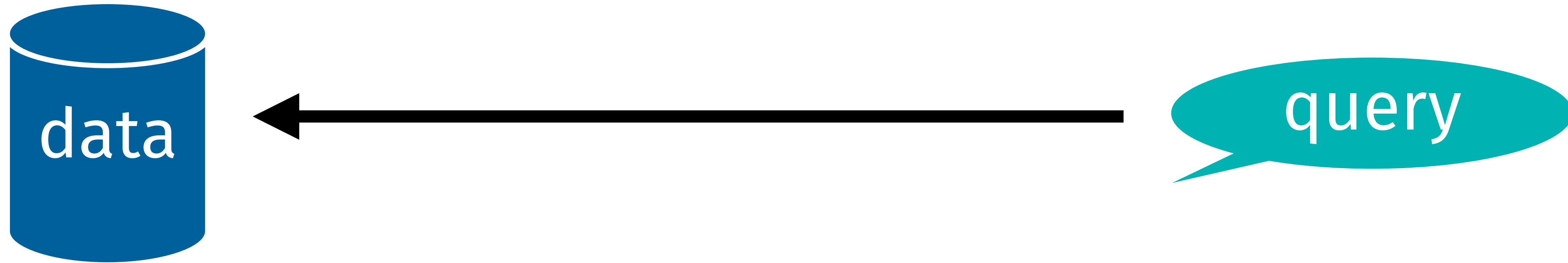
Challenges of querying data



Why is it hard to query data?

- Need to know **how database(s) are structured**
- Can be **many ways** to **express the same thing**

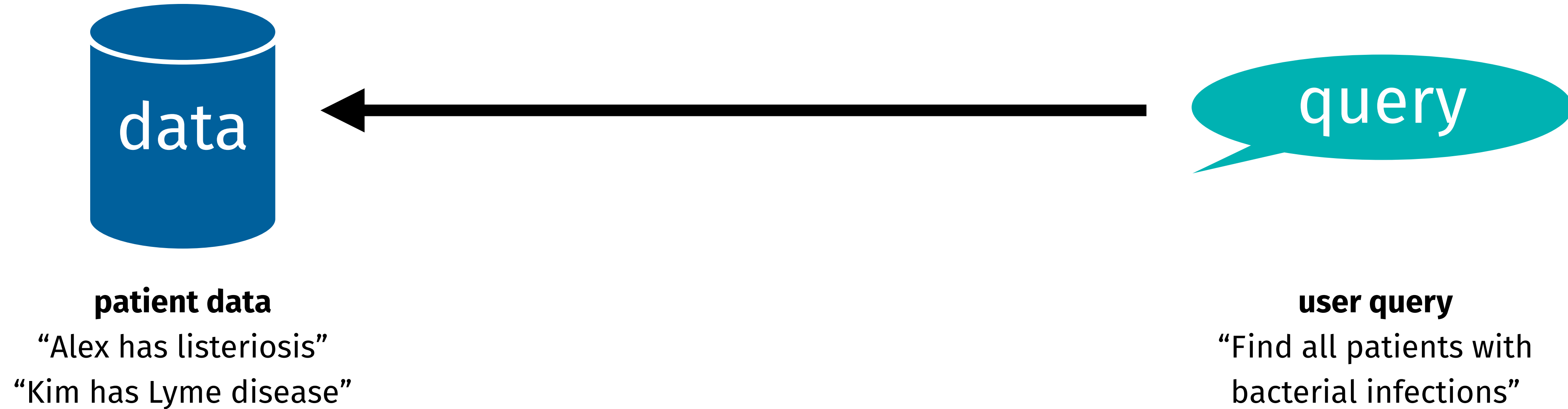
Challenges of querying data



Why is it hard to query data?

- Need to know **how database(s) are structured**
- Can be **many ways** to **express the same thing**
- **Query terms not present** nor easily mapped to **database** (e.g. different granularity)
- Need to exploit **implicit information, not in data**

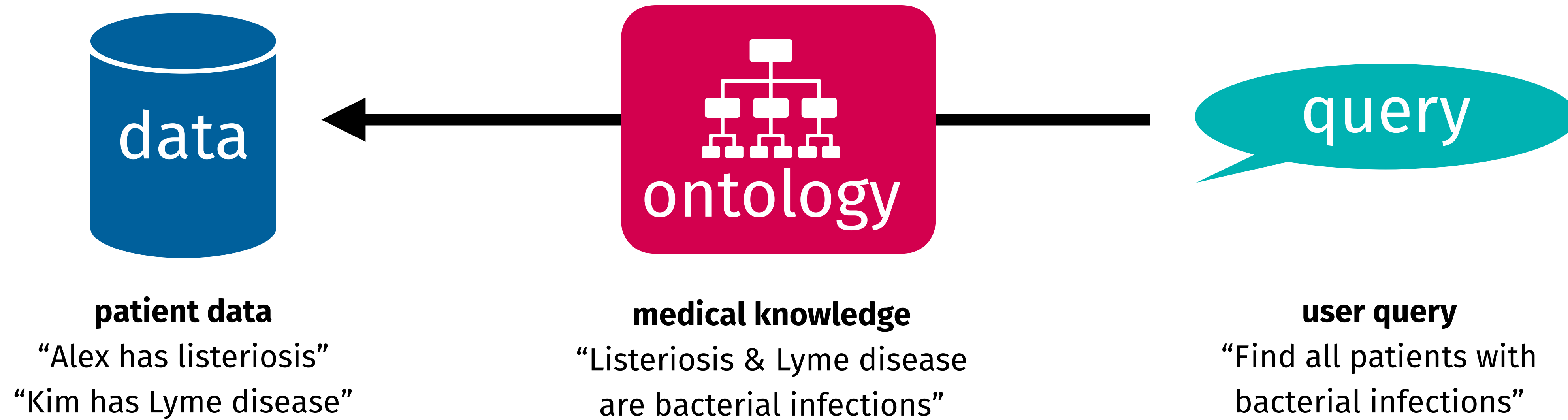
Challenges of querying data



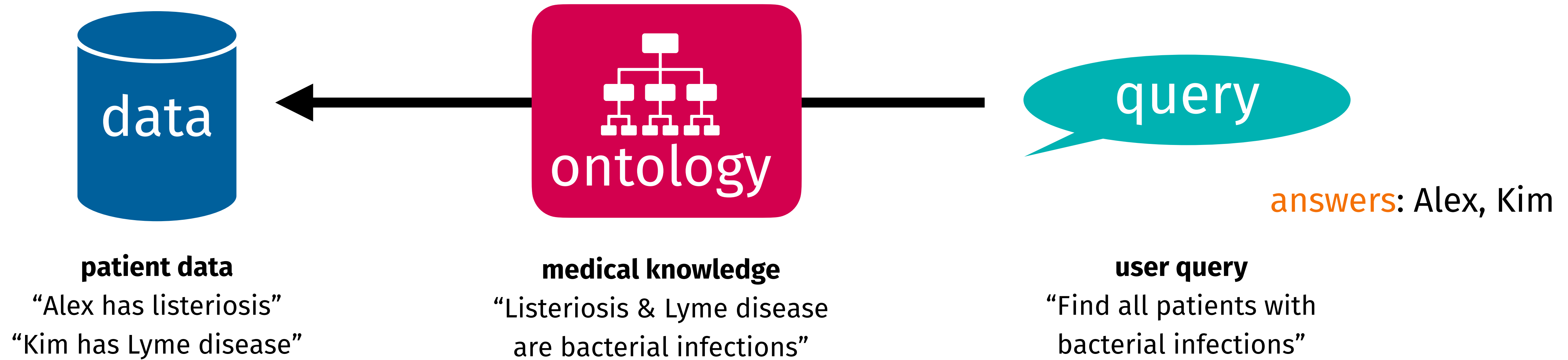
Why is it hard to query data?

- Need to know **how database(s) are structured**
- Can be **many ways** to **express the same thing**
- **Query terms not present** nor easily mapped to **database** (e.g. different granularity)
- Need to exploit **implicit information, not in data**

Ontologies to the rescue!



Ontologies to the rescue!



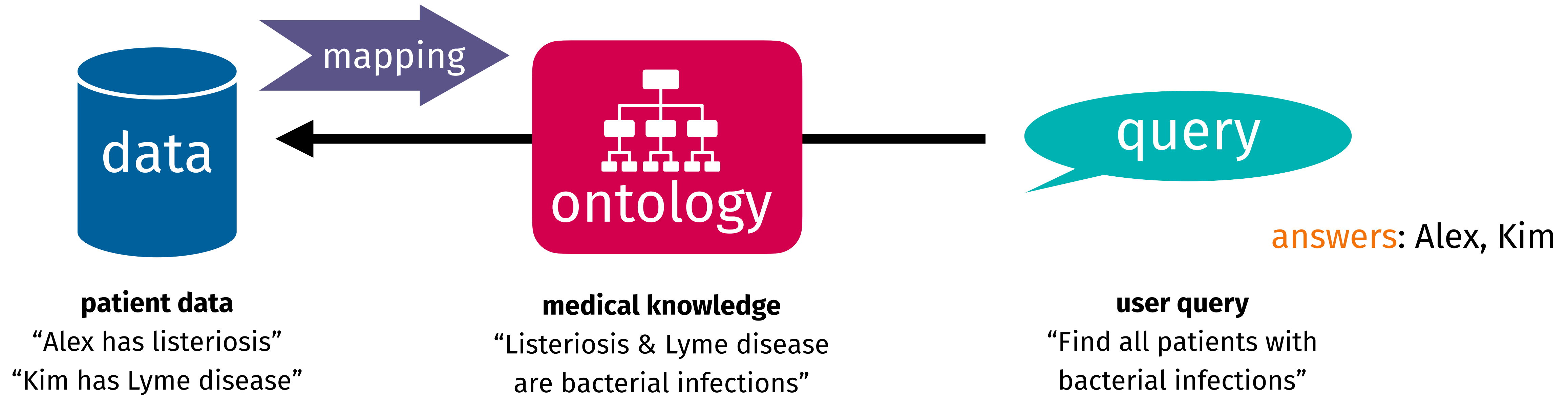
Ontologies to the rescue!



Ontology provides:

- convenient vocabulary for users to formulate queries
- domain knowledge that can be exploited during query answering

Ontologies to the rescue!

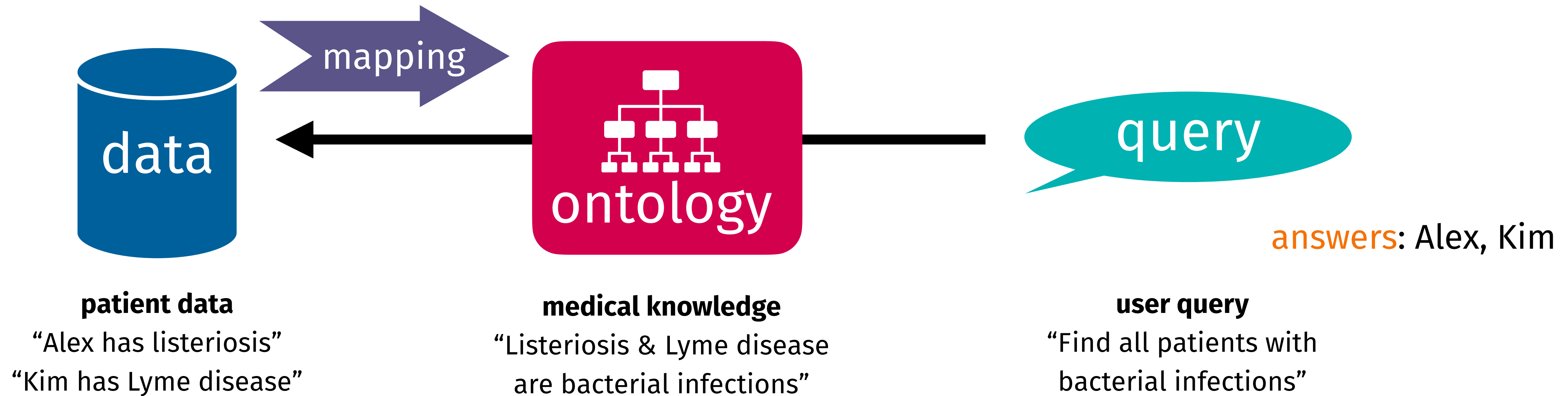


Ontology provides:

- **convenient vocabulary** for users to **formulate queries**
- **domain knowledge** that can be **exploited during query answering**

Mapping: can be used to bridge ontology + database vocabularies

Ontologies to the rescue!



Ontology provides:

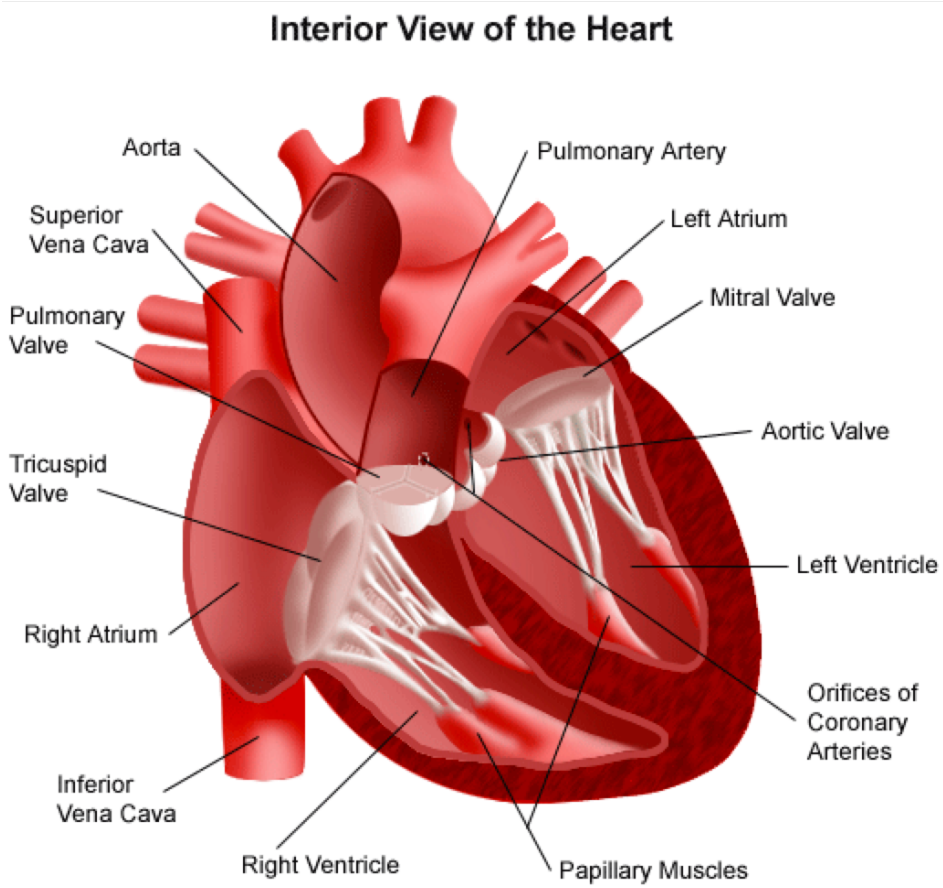
- **convenient vocabulary** for users to **formulate queries**
- **domain knowledge** that can be **exploited during query answering**

Today's talk:
focus on ontologies,
mostly ignore mappings

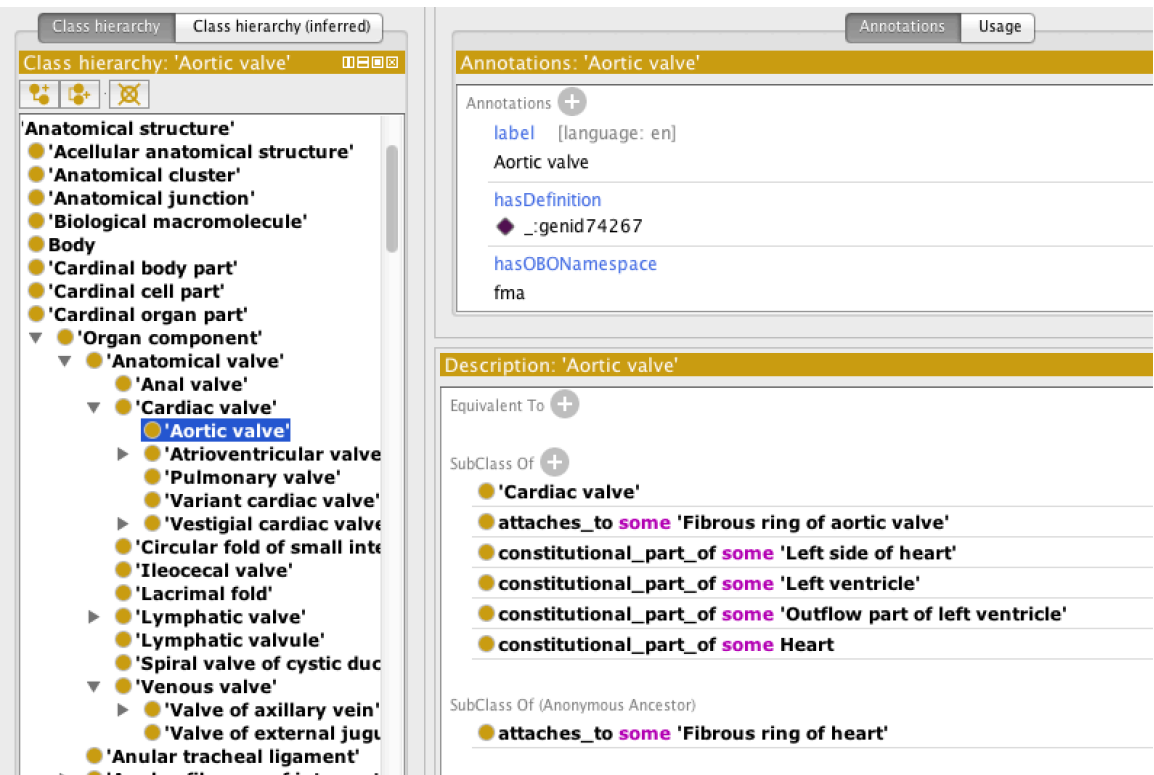
Mapping: can be used to bridge ontology + database vocabularies

Application areas

*Querying & exchanging
medical records*



Medecine



*Support analysis of health
data for medical research*



Life sciences

*Help scientists share, query, &
visualize experimental data*



Entreprise information systems

Easy and flexible access to data to support decision-making



Ontologies in healthcare: SNOMED

Large-scale comprehensive medical ontology

- more than 350,000 terms covering all aspects of clinical healthcare
 - symptoms, diagnoses, medical procedures, body structures, laboratory tests, organisms, pharmaceutical products



Ontologies in healthcare: SNOMED

Large-scale comprehensive medical ontology

- more than **350,000 terms** covering all aspects of clinical healthcare
 - symptoms, diagnoses, medical procedures, body structures, laboratory tests, organisms, pharmaceutical products



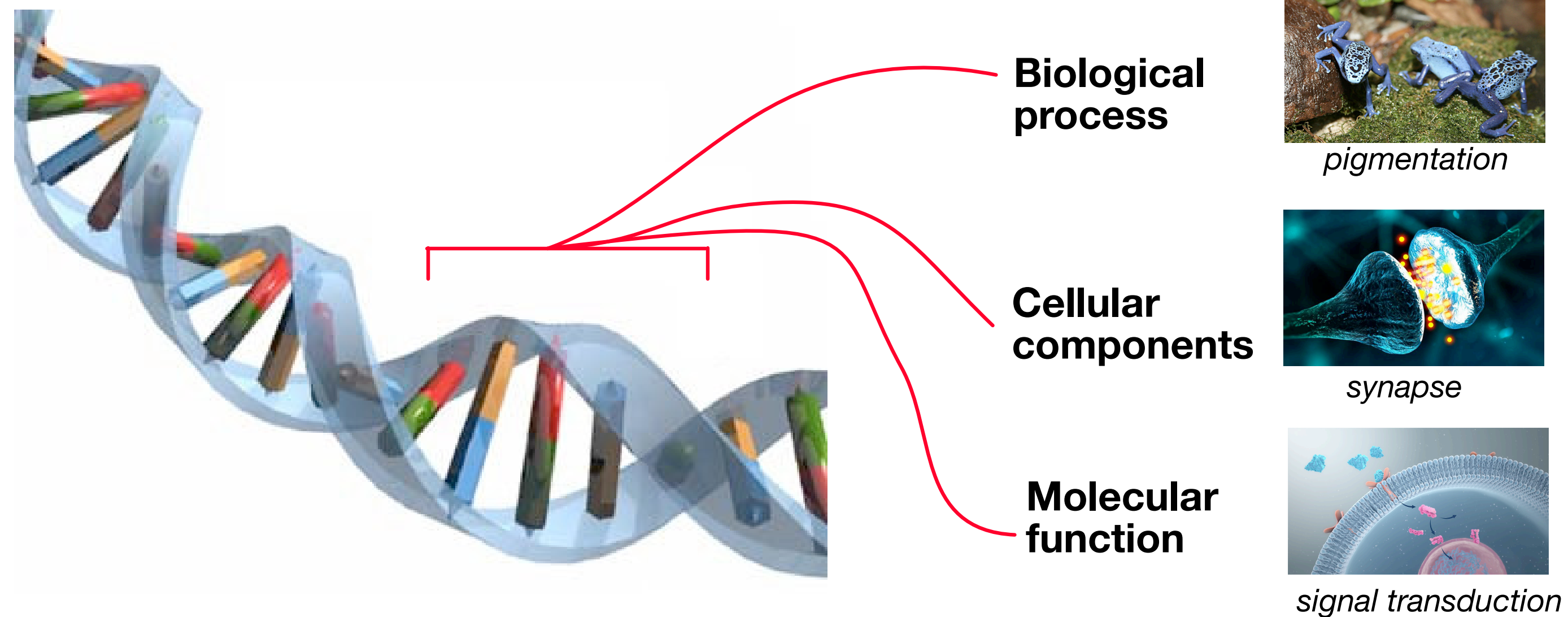
Widely adopted standard for healthcare terminology

- in use in **> 80 countries** (including U.S., Canada, UK)
- used in **health IT** systems (e.g. IBM Watson Health, Babylon Health)

Ontologies in life sciences: Gene Ontology

Aim: **rigorous shared vocabulary** to describe the **role of genes across organisms**

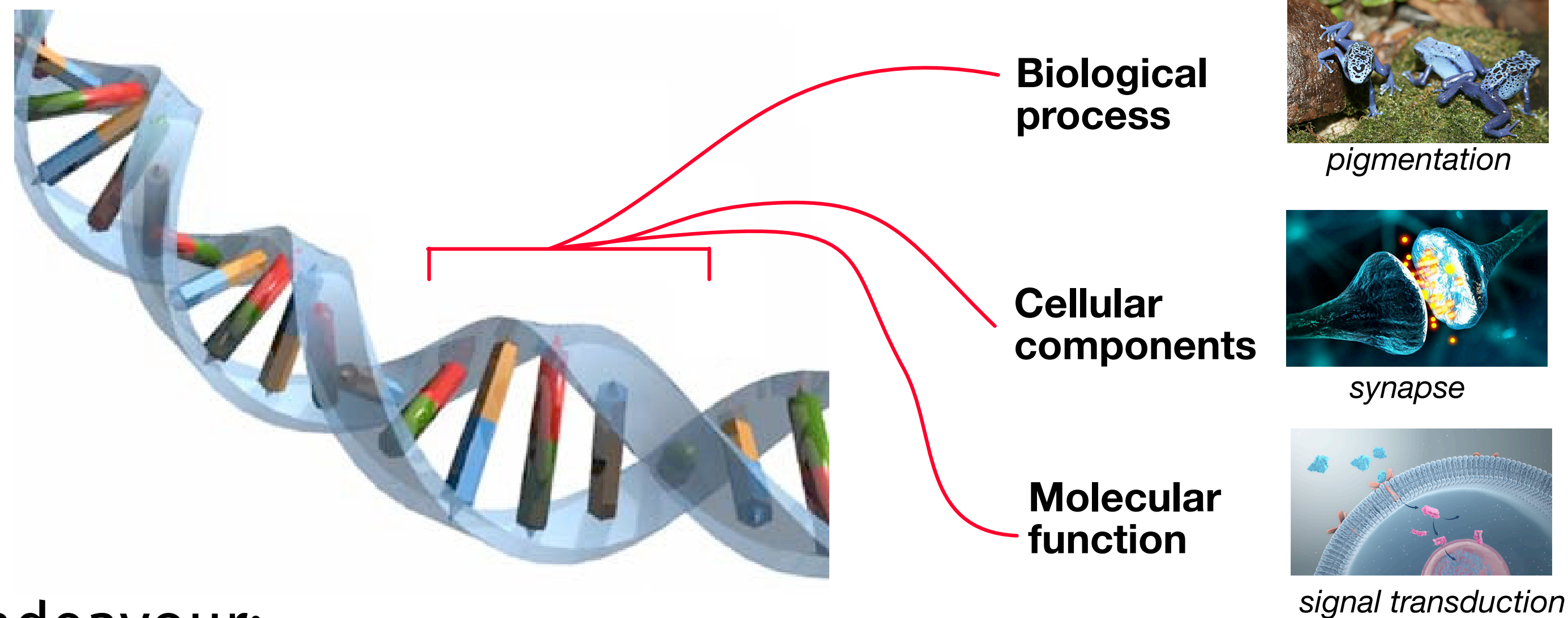
Annotations: evidence-based statements relating a particular gene product to ontology terms



Ontologies in life sciences: Gene Ontology

Aim: **rigorous shared vocabulary** to describe the **role of genes across organisms**

Annotations: evidence-based statements relating a particular gene product to ontology terms

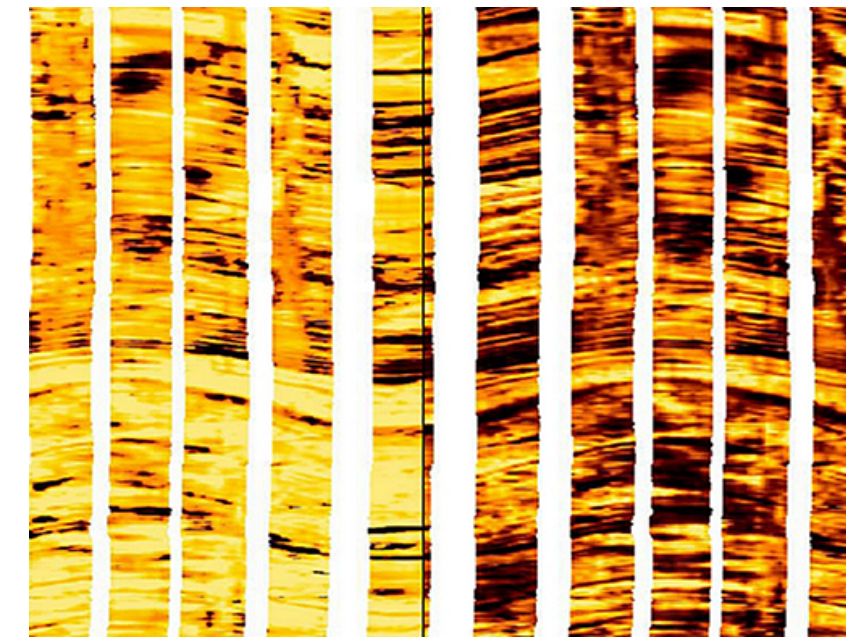
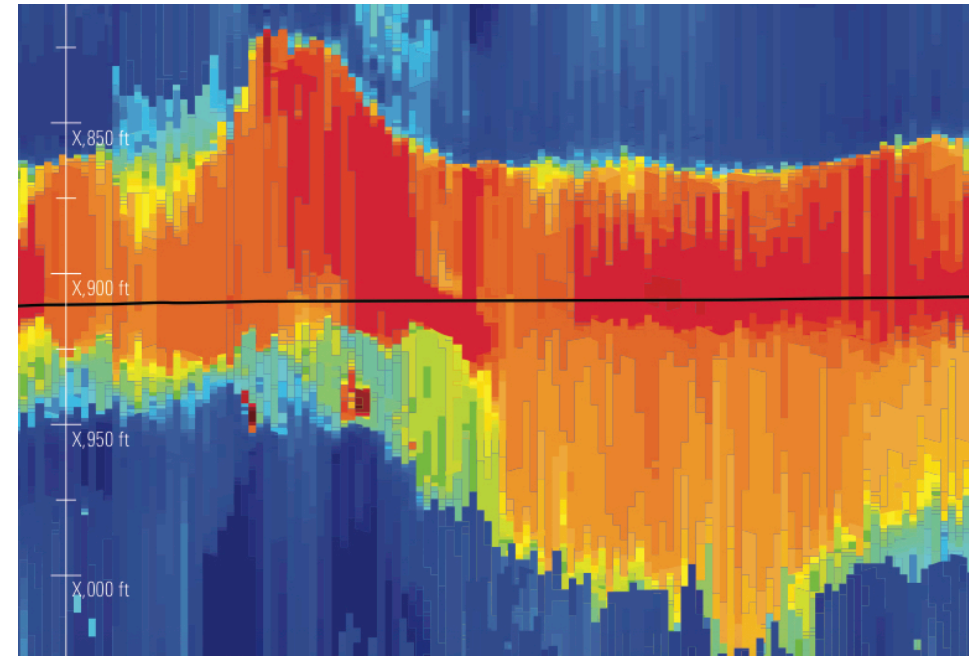


Very successful endeavour:

- > 100K published scientific articles with keyword “Gene Ontology”
- > 700K experimentally-supported annotations

Ontologies in industry: Statoil usecase

Goal: aid geologists
in gathering information
for analysis

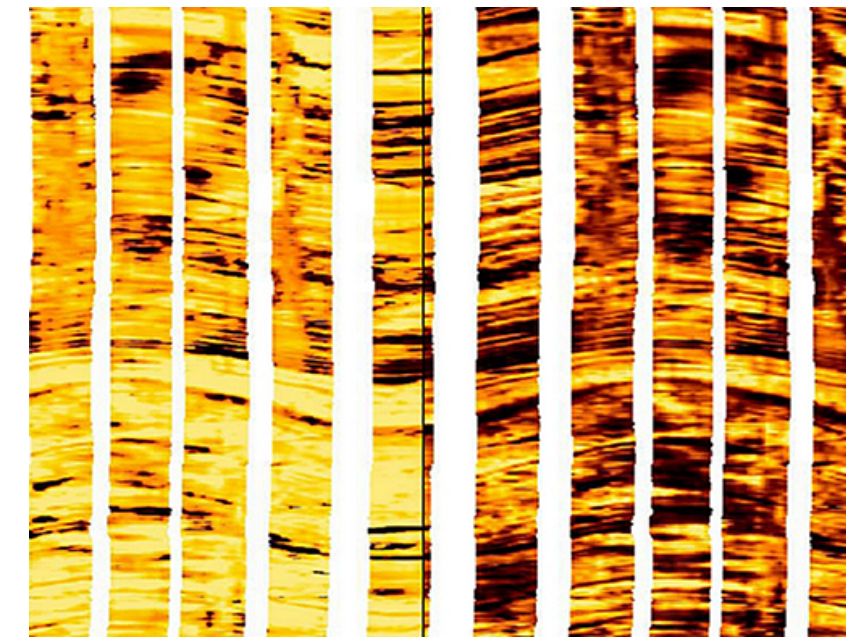
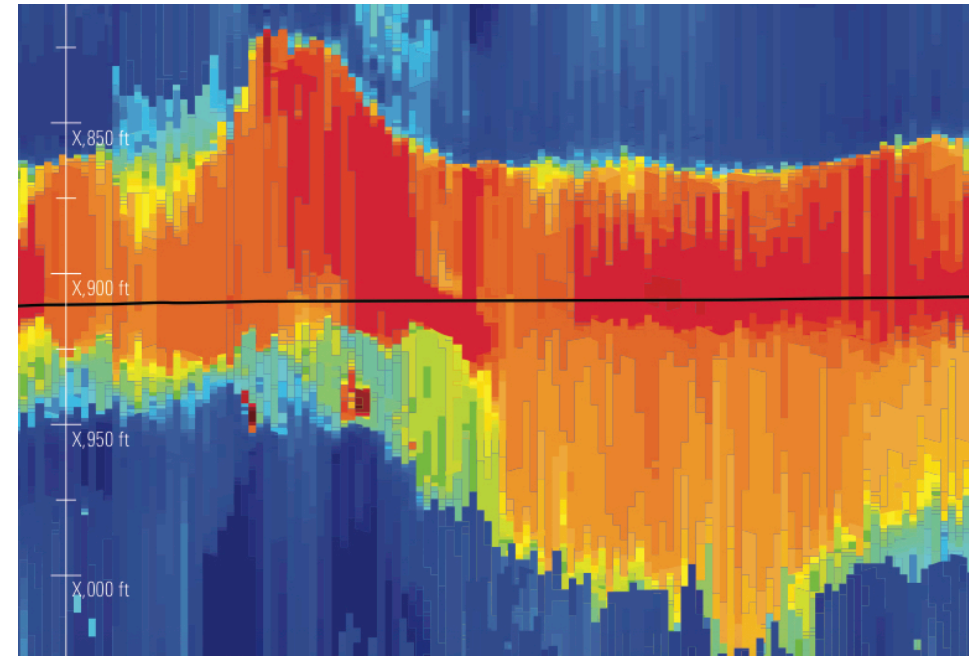


Relevant data stored across > 3000 database tables

- geologist question = complex query (thousands of terms, 50-200 joins)

Ontologies in industry: Statoil usecase

Goal: aid geologists
in gathering information
for analysis



Relevant data stored across > 3000 database tables

- geologist question = complex query (thousands of terms, 50-200 joins)

OMQA approach:

- ontology provides familiar vocabulary for query formulation
- mappings used to connect database tables to ontology terms
- use reasoning to automatically transform ontology query into DB query

Overview of today's talk

Motivation

Ontology-Mediated Query Answering

- ontologies, queries, certain answers

Complexity & Algorithms

- query rewriting, materialization approaches

Research Directions

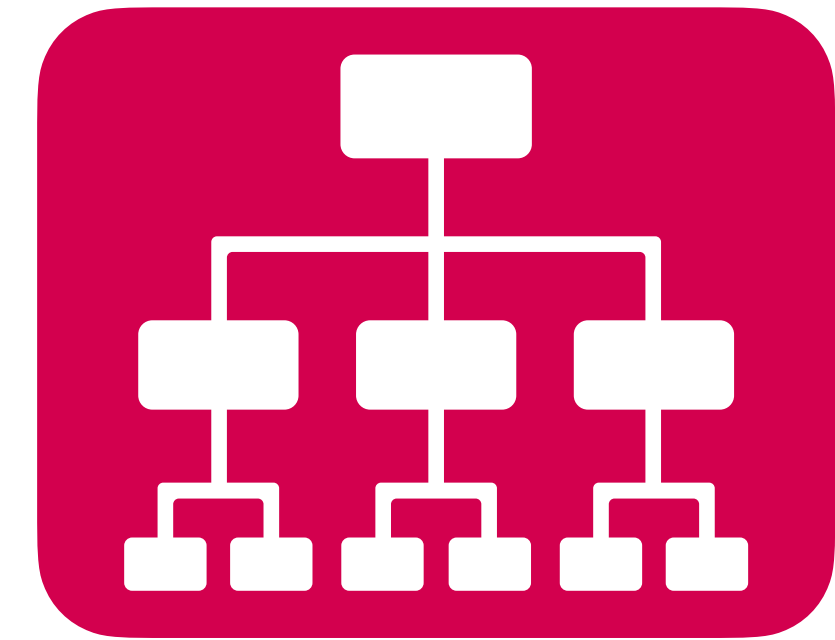
Ontology-Mediated Query Answering

Ontologies

Ontology a formal specification of the knowledge of a particular domain, thereby making it amenable to machine processing

Such a specification consists of:

- terminology (or vocabulary) of the domain
- semantic relationships between terms
 - relations of specificity or generality, equivalence, disjointness, ...

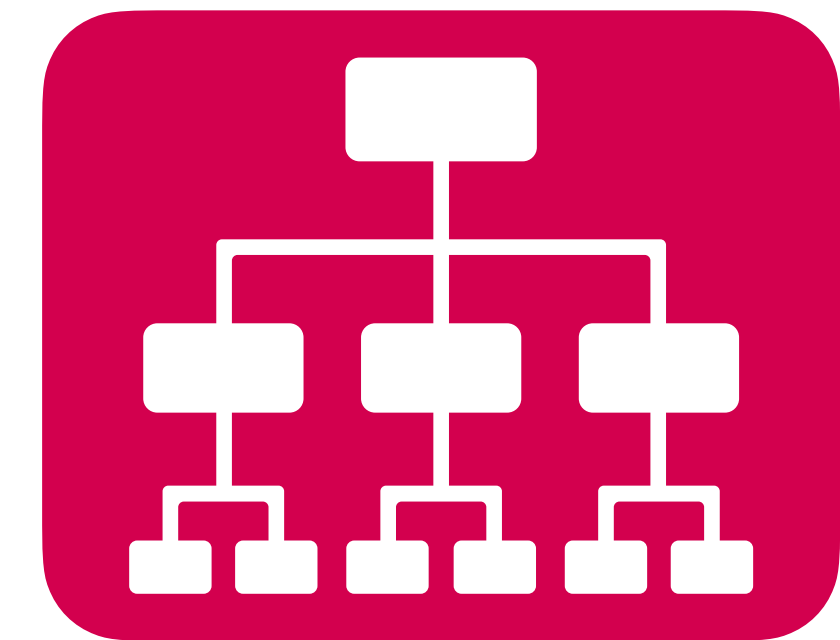


Ontologies

Ontology a formal specification of the knowledge of a particular domain, thereby making it amenable to machine processing

Such a specification consists of:

- terminology (or vocabulary) of the domain
- semantic relationships between terms
 - relations of specificity or generality, equivalence, disjointness, ...



Desiderata for ontology language:

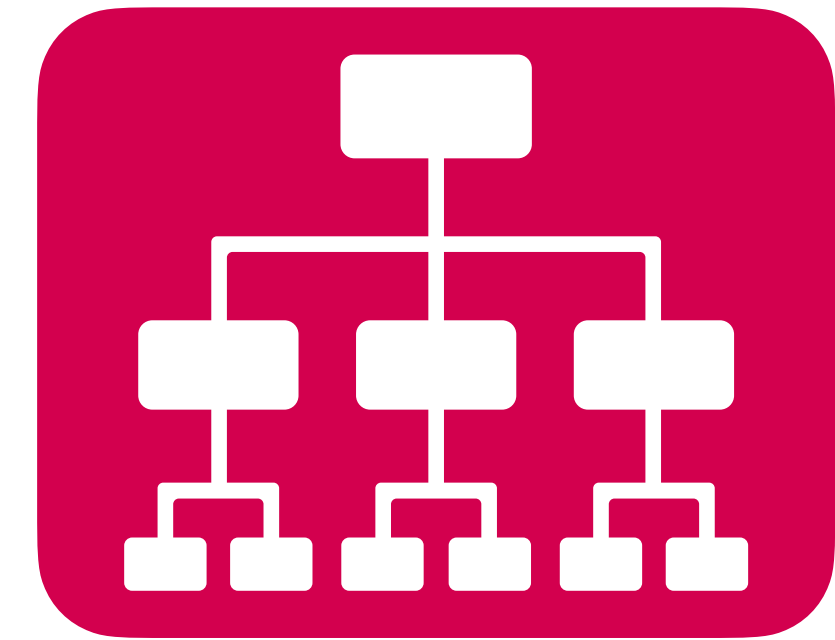
- useful expressivity
- well-defined semantics
- decidable / efficient reasoning

Ontologies

Ontology a formal specification of the knowledge of a particular domain, thereby making it amenable to machine processing

Such a specification consists of:

- terminology (or vocabulary) of the domain
- semantic relationships between terms
 - relations of specificity or generality, equivalence, disjointness, ...



Desiderata for ontology language:

- useful expressivity
- well-defined semantics
- decidable / efficient reasoning

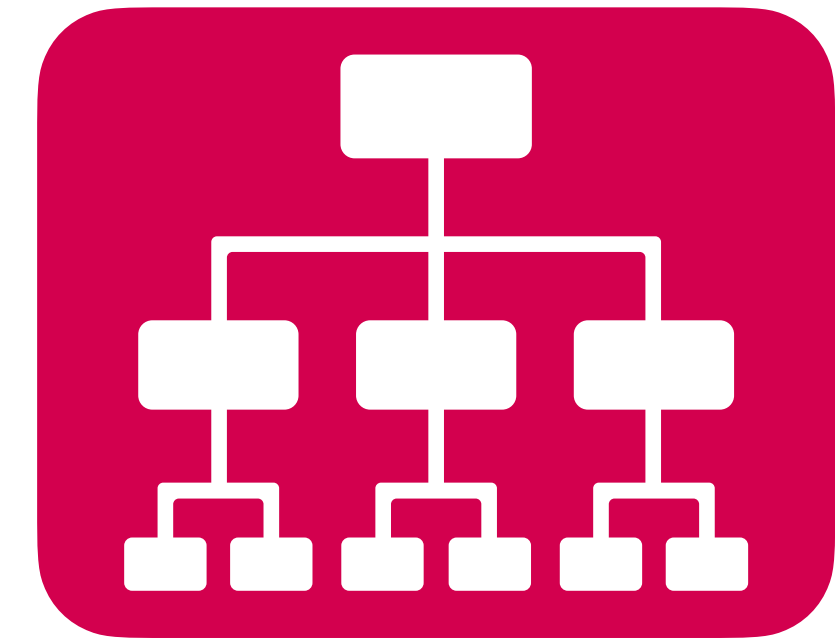
First-order logic?

Ontologies

Ontology a formal specification of the knowledge of a particular domain, thereby making it amenable to machine processing

Such a specification consists of:

- terminology (or vocabulary) of the domain
- semantic relationships between terms
 - relations of specificity or generality, equivalence, disjointness, ...



Desiderata for ontology language:

- useful expressivity
- well-defined semantics
- decidable / efficient reasoning

First-order logic?

- highly expressive ✓
- clear semantics ✓
- undecidable ✗

LOTS of ontology languages!

Description Logics (DLs)

$SR\mathcal{OIQ}$ \mathcal{EL} $DL\text{-}Lite_{\mathcal{A}}$ $DL\text{-}Lite_{\mathcal{R}}$ RSA \mathcal{DLFD}
 $ALCH\mathcal{OI}$ \mathcal{ELI} $Horn\text{-}ALC$
 \mathcal{ELH}_{\perp} $ALCF$ $Horn\text{-}SHIQ$ $\mathcal{EL}++$ ALC $DL\text{-}Lite_{bool}$

Rule-Based Languages

$agrd$ $Datalog$ $sticky$ MFA $Datalog^{+/-}$ $weakly\ acyclic$
 $existential\ rules$
 $greedy\ bts$ $linear$ $frontier\ one$ $tgds$ $guarded$ $disjunctive$

OWL / RDFS

OWL 2 QL OWL 2 EL OWL 2 RL

OWL 2 Web Ontology Language
Document Overview (Second Edition)
W3C Recommendation 11 December 2012



RDF Schema (RDFS)

OWL DL OWL Full
OWL Lite

LOTS of ontology languages!

Description Logics (DLs)

Family of decidable fragments of first-order logic
Only unary and binary relations (classes, properties)
Each DL defined by modelling constructs it offers

SROIQ *ALCHOI* *ELH_U* *DLFD* *Horn-ALC* *DL-Lite_{bool}*

Rule-Based Languages

agrd **Datalog** sticky MFA **Datalog^{+/-}** weakly acyclic
greedy bts **existential rules** tgds guarded disjunctive
linear frontier-one

OWL / RDFS

OWL 2 QL OWL 2 EL OWL 2 RL

OWL 2 Web Ontology Language
Document Overview (Second Edition)
W3C Recommendation 11 December 2012



RDF Schema (RDFS)

OWL DL OWL Full
OWL Lite

LOTS of ontology languages!

Description Logics (DLs)

- Family of decidable fragments of first-order logic
- Only unary and binary relations (classes, properties)
- Each DL defined by modelling constructs it offers

Rule-Based Languages

- Classes of logical rules, expressible in first-order logic
- Admit relations of arbitrary arity
- Defined by shape of rules or properties of ruleset

OWL / RDFS

OWL 2 QL OWL 2 EL OWL 2 RL

RDF Schema (RDFS)

OWL 2 Web Ontology Language
Document Overview (Second Edition)
W3C Recommendation 11 December 2012



OWL DL

OWL Lite

OWL Full

LOTS of ontology languages!

Description Logics (DLs)

Family of decidable fragments of first-order logic
Only unary and binary relations (classes, properties)
Each DL defined by modelling constructs it offers

Rule-Based Languages

Classes of logical rules, expressible in first-order logic
Admit relations of arbitrary arity
Defined by shape of rules or properties of ruleset

OWL / RDFS

W3C standards for ontologies & terminologies
OWL based upon DLs, but designed for use in practice
Additional features, e.g. annotations, meta-modelling

LOTS of ontology languages!

Description Logics (DLs)

$\text{Course} \sqsubseteq \exists \text{taughtBy}.\top$

$\text{GradCourse} \sqsubseteq \text{Course} \sqcap \forall \text{taughtBy}.\text{Professor}$

Rule-Based Languages

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y)$

$\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Professor}(y)$

OWL / RDFS

```
<SubClassOf>
  <Class IRI="Course"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="taughtBy"/>
    <Class abbreviatedIRI="owl:Thing"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
```

```
Class: GradCourse
SubClassOf:
  Course,
  taughtBy only Professor
```

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> → <head>

if <body>, then <head>

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> \rightarrow <head>

if <body>, then <head>

Simple (Datalog) rules head vars \subseteq body vars & all vars are universally quantified

$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> \rightarrow <head>

if <body>, then <head>

Simple (Datalog) rules head vars \subseteq body vars & all vars are universally quantified

$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

Existential rules may have existentially quantified vars in head (reason on unknown individuals)

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$ $\text{Course}(x) \rightarrow \exists y \text{ taughtBy}(x, y)$

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> \rightarrow <head>

if <body>, then <head>

Simple (Datalog) rules head vars \subseteq body vars & all vars are universally quantified

$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

Existential rules may have existentially quantified vars in head (reason on unknown individuals)

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$ $\text{Course}(x) \rightarrow \exists y \text{ taughtBy}(x, y)$

Disjunctive rules may have disjunction in ruleheads (reasoning by cases)

$\text{Course}(x) \rightarrow \text{UndergradCourse}(x) \vee \text{GradCourse}$

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> \rightarrow <head>

if <body>, then <head>

Simple (Datalog) rules head vars \subseteq body vars & all vars are universally quantified

$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

Existential rules may have existentially quantified vars in head (reason on unknown individuals)

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$ $\text{Course}(x) \rightarrow \exists y \text{ taughtBy}(x, y)$

Disjunctive rules may have disjunction in ruleheads (reasoning by cases)

$\text{Course}(x) \rightarrow \text{UndergradCourse}(x) \vee \text{GradCourse}(x)$

Negative constraints forbid contradictory situations

$\text{UndergradCourse}(x) \wedge \text{GradCourse}(x) \rightarrow \perp$

Ontologies as sets of rules

Most ontology axioms can be expressed using some forms of **rules**

<body> \rightarrow <head>

if <body>, then <head>

Simple (Datalog) rules head vars \subseteq body vars & all vars are universally quantified

$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

Existential rules may have existentially quantified vars in head (reason on unknown individuals)

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$ $\text{Course}(x) \rightarrow \exists y \text{ taughtBy}(x, y)$

Disjunctive rules may have disjunction in ruleheads (reasoning by cases)

$\text{Course}(x) \rightarrow \text{UndergradCourse}(x) \vee \text{GradCourse}$

Negative constraints forbid contradictory situations

$\text{UndergradCourse}(x) \wedge \text{GradCourse}(x) \rightarrow \perp$

Rules with equality functionality / cardinality

$\text{hasMainSupervisor}(x, y) \wedge \text{hasMainSupervisor}(x, z) \rightarrow y = z$

Queries

Atomic Queries

Consist of a **single atom**, used to find all tuples in relation

find all PhD holders

$$q_1(x) = \text{PhDHolder}(x)$$

find all pairs of a course and who teaches it

$$q_2(x, y) = \text{taughtBy}(x, y)$$

Queries

Atomic Queries

Consist of a **single atom**, used to find all tuples in relation

find all PhD holders *find all pairs of a course and who teaches it*

$$q_1(x) = \text{PhDHolder}(x)$$

$$q_2(x, y) = \text{taughtBy}(x, y)$$

Conjunctive Queries

Consist of a **conjunctive of atoms**, some of the **variables may be existentially quantified** (projected away)

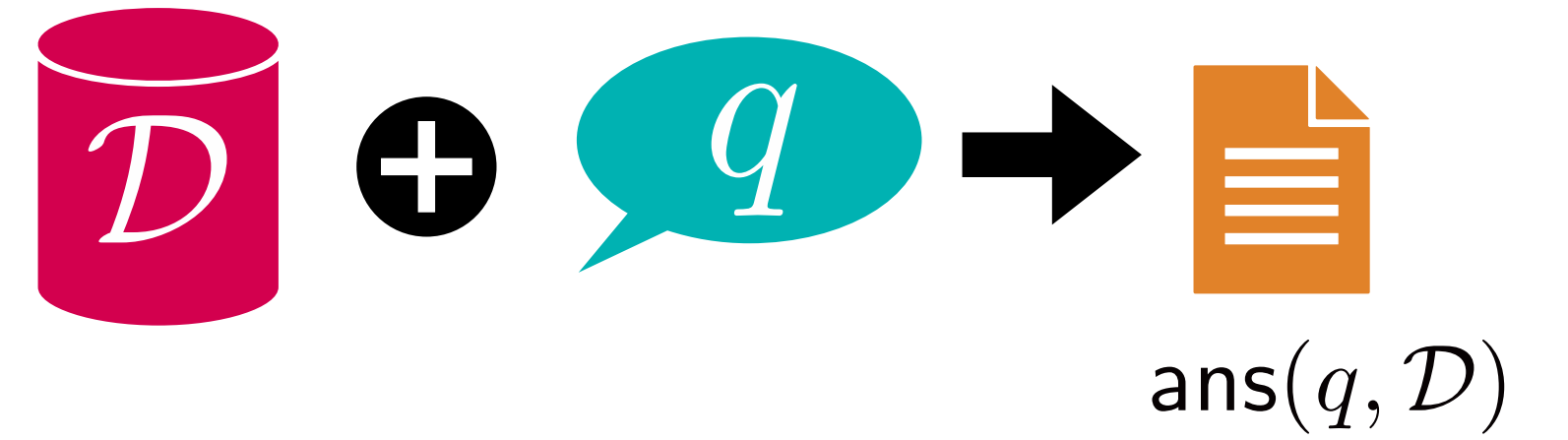
find all courses that are taught by a PhD holder

$$q_3(x) = \exists y \text{ taughtBy}(x, y) \wedge \text{PhDHolder}(y)$$


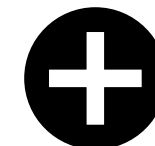

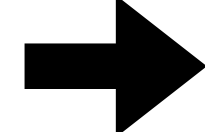

Related to: **select-project-join** in SQL, **basic graph patterns** in SPARQL

Certain answers

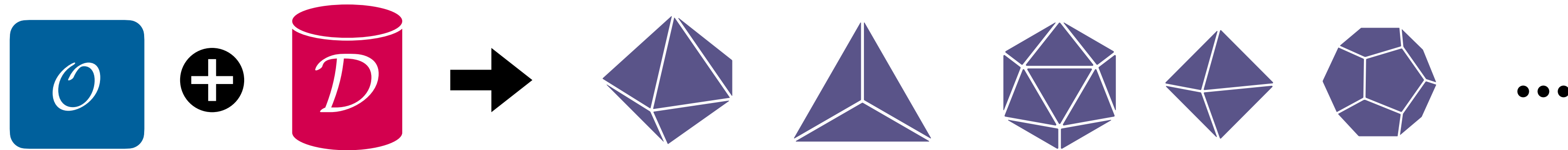
Know how to define answers to queries on (plain) datasets:
(finite set of facts)



Certain answers


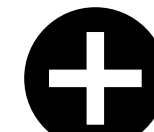

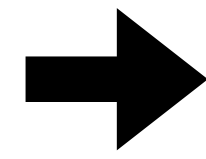

Know how to define answers to queries on **(plain) datasets**:
(finite set of facts)     
 $\text{ans}(q, \mathcal{D})$

To define answers in OMQA setting, we consider the **models of ontology + dataset**:

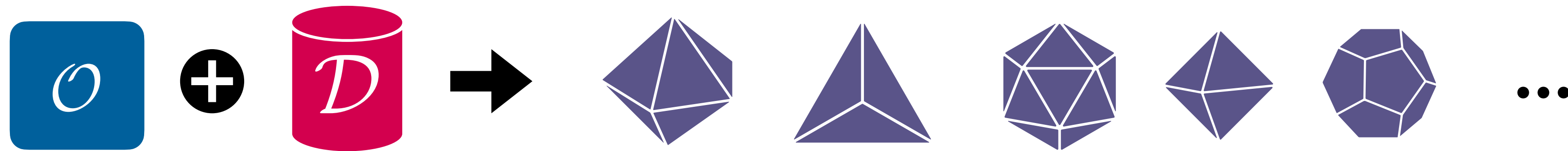


$\text{Mods}(\mathcal{O}, \mathcal{D})$: interpretations satisfying $(\mathcal{O}, \mathcal{D})$ / states of the world compatible with $(\mathcal{O}, \mathcal{D})$

Certain answers

Know how to define answers to queries on **(plain) datasets**:
(finite set of facts)     
 $\text{ans}(q, \mathcal{D})$


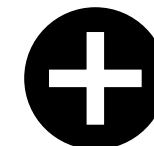

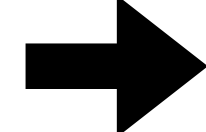

To define answers in OMQA setting, we consider the **models of ontology + dataset**:



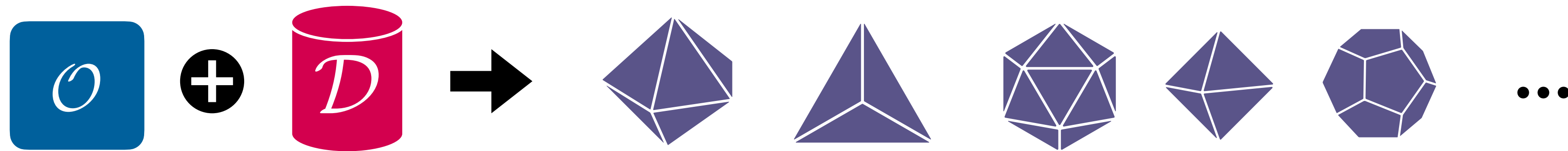
$\text{Mods}(\mathcal{O}, \mathcal{D})$: interpretations satisfying $(\mathcal{O}, \mathcal{D})$ / states of the world compatible with $(\mathcal{O}, \mathcal{D})$

In each model, can define answers like for datasets:     
 $\text{ans}(q, \mathcal{M})$

Certain answers

Know how to define answers to queries on **(plain) datasets**:
(finite set of facts)     
 $\text{ans}(q, \mathcal{D})$

To define answers in OMQA setting, we consider the **models of ontology + dataset**:



$\text{Mods}(\mathcal{O}, \mathcal{D})$: interpretations satisfying $(\mathcal{O}, \mathcal{D})$ / states of the world compatible with $(\mathcal{O}, \mathcal{D})$

In each model, can define answers like for datasets:     
 $\text{ans}(q, \mathcal{M})$

Certain answers: tuples that are **answers in every model** (i.e. **entailed / derivable** from $(\mathcal{O}, \mathcal{D})$)

$$(\mathcal{O}, \mathcal{D}) \models q(\vec{a}) \iff \vec{a} \in \text{ans}(q, \mathcal{M}) \text{ for every } \mathcal{M} \in \text{Mods}(\mathcal{O}, \mathcal{D})$$

Example: Certain answers

Consider the following ontology and dataset:

$$\begin{aligned} \mathcal{O} = & \text{Prof}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \rightarrow \text{Course}(x) & \text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \\ & \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \end{aligned}$$

$$\mathcal{D} = \text{taughtBy}(\text{cs50}, \text{paolo}) \quad \text{Postdoc}(\text{paolo}) \quad \text{GradCourse}(\text{cs87})$$

and reconsider the query $q_3(x) = \exists y \text{taughtBy}(x, y) \wedge \text{PhDHolder}(y)$

Example: Certain answers

Consider the following ontology and dataset:

$$\begin{aligned}\mathcal{O} = & \text{Prof}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \rightarrow \text{Course}(x) & \text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \\ & \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)\end{aligned}$$

$$\mathcal{D} = \text{taughtBy}(\text{cs50}, \text{paolo}) \quad \text{Postdoc}(\text{paolo}) \quad \text{GradCourse}(\text{cs87})$$

and reconsider the query $q_3(x) = \exists y \text{taughtBy}(x, y) \wedge \text{PhDHolder}(y)$

Two certain answers to $q_3(x)$ w.r.t. $(\mathcal{O}, \mathcal{D})$: cs50 and cs87

$\mathcal{O}, \mathcal{D} \models q_3(\text{cs50})$ due to $\text{taughtBy}(\text{cs50}, \text{paolo}) \quad \text{Postdoc}(\text{paolo}) \quad \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x)$

Example: Certain answers

Consider the following ontology and dataset:

$$\begin{aligned} \mathcal{O} = & \text{Prof}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \rightarrow \text{Course}(x) & \text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \\ & \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) & \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \end{aligned}$$

$$\mathcal{D} = \text{taughtBy}(\text{cs50}, \text{paolo}) \quad \text{Postdoc}(\text{paolo}) \quad \text{GradCourse}(\text{cs87})$$

and reconsider the query $q_3(x) = \exists y \text{taughtBy}(x, y) \wedge \text{PhDHolder}(y)$

Two certain answers to $q_3(x)$ w.r.t. $(\mathcal{O}, \mathcal{D})$: cs50 and cs87

$$\mathcal{O}, \mathcal{D} \models q_3(\text{cs50}) \text{ due to } \text{taughtBy}(\text{cs50}, \text{paolo}) \quad \text{Postdoc}(\text{paolo}) \quad \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x)$$

$$\begin{aligned} \mathcal{O}, \mathcal{D} \models q_3(\text{cs87}) \text{ due to } & \text{GradCourse}(\text{cs87}) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \\ & \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \quad \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \end{aligned}$$

Pointers to OMQA literature

Early paper

Poggi et al. **Linking Data to Ontologies**. Journal of Data Semantics, 2008.

Tutorial chapters

Bienvenu & Ortiz. **Ontology-Mediated Query Answering with Data-Tractable Description Logics**. Reasoning Web, 2015.

Mugnier & Thomazo. **An Introduction to Ontology-Based Query Answering with Existential Rules**. Reasoning Web, 2014.

Short survey

Xiao et al. **Ontology-Based Data Access: A Survey**. IJCAI 2018.

Complexity & Algorithms

Complexity of OMQA

OMQA decision problem: **identify certain answers**, i.e. decide whether $(\mathcal{O}, \mathcal{D}) \models q(\vec{a})$

Extensive complexity analysis covering wide range of ontology & query languages, restrictions

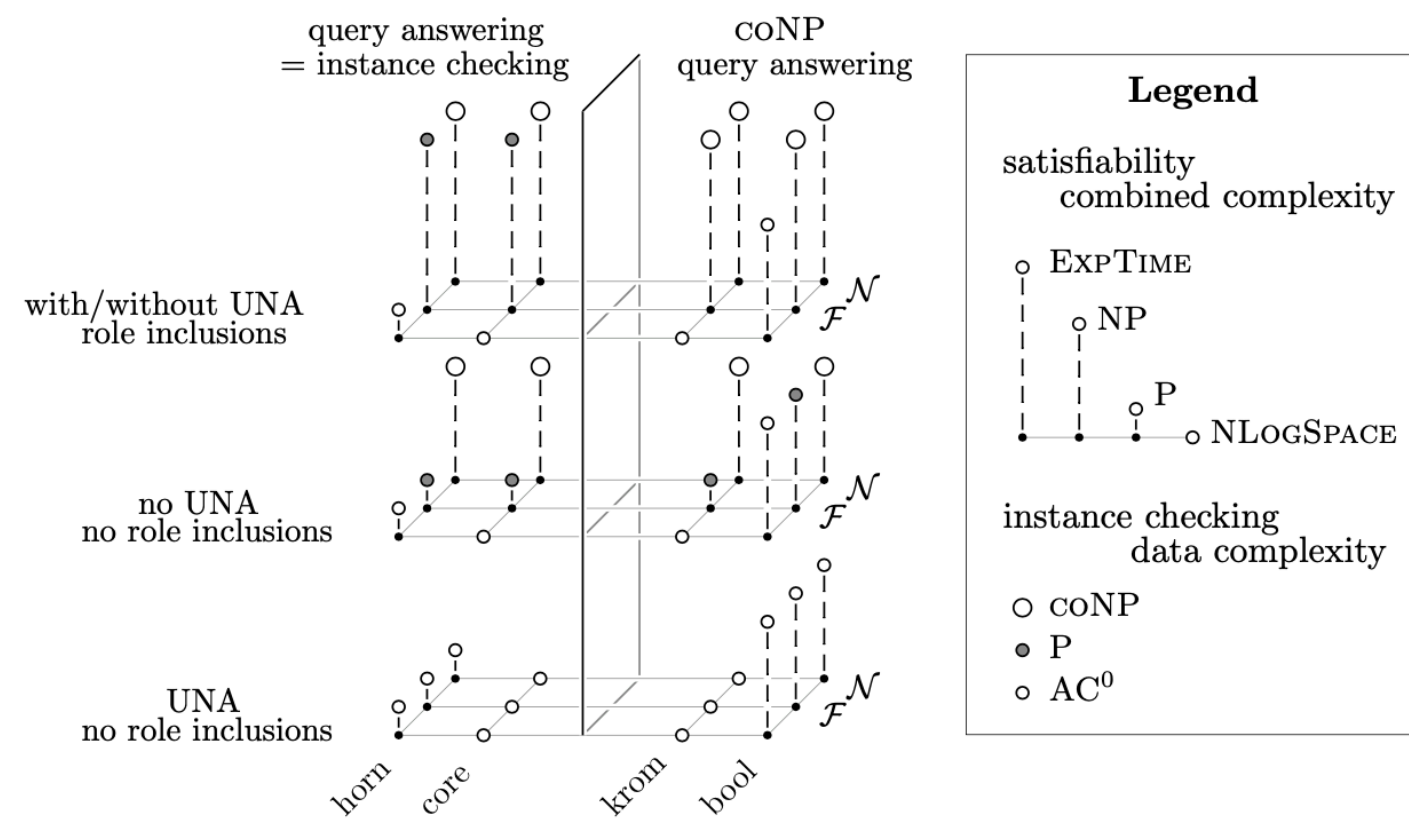
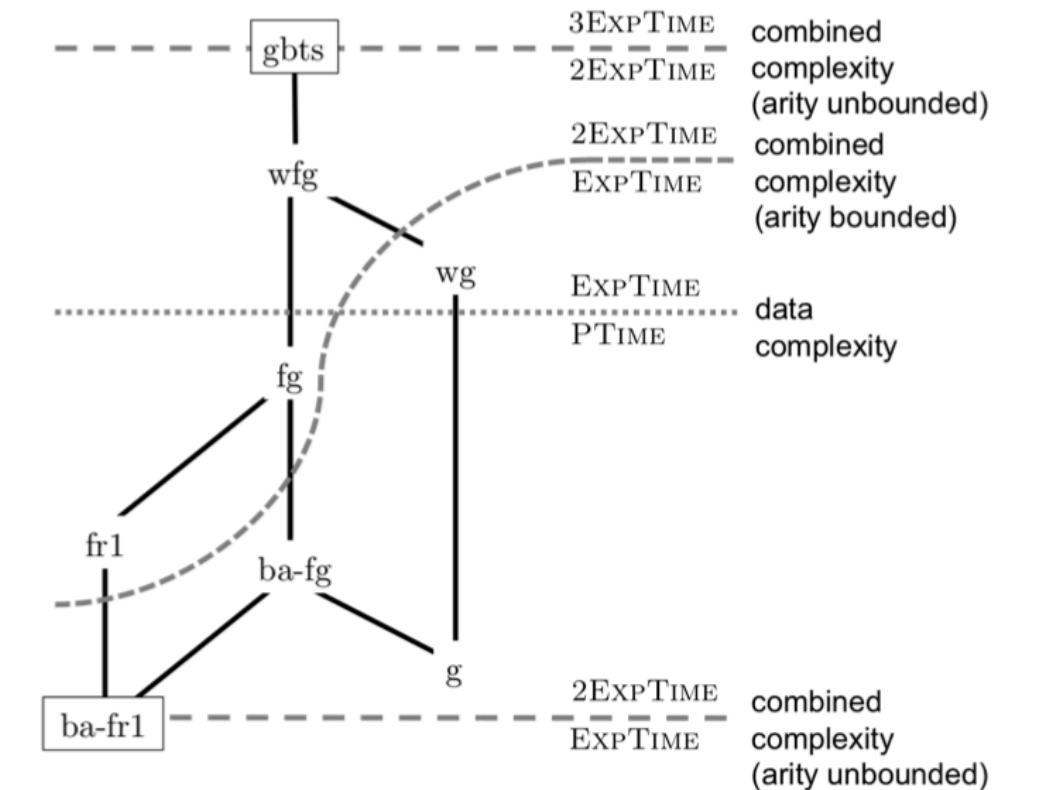
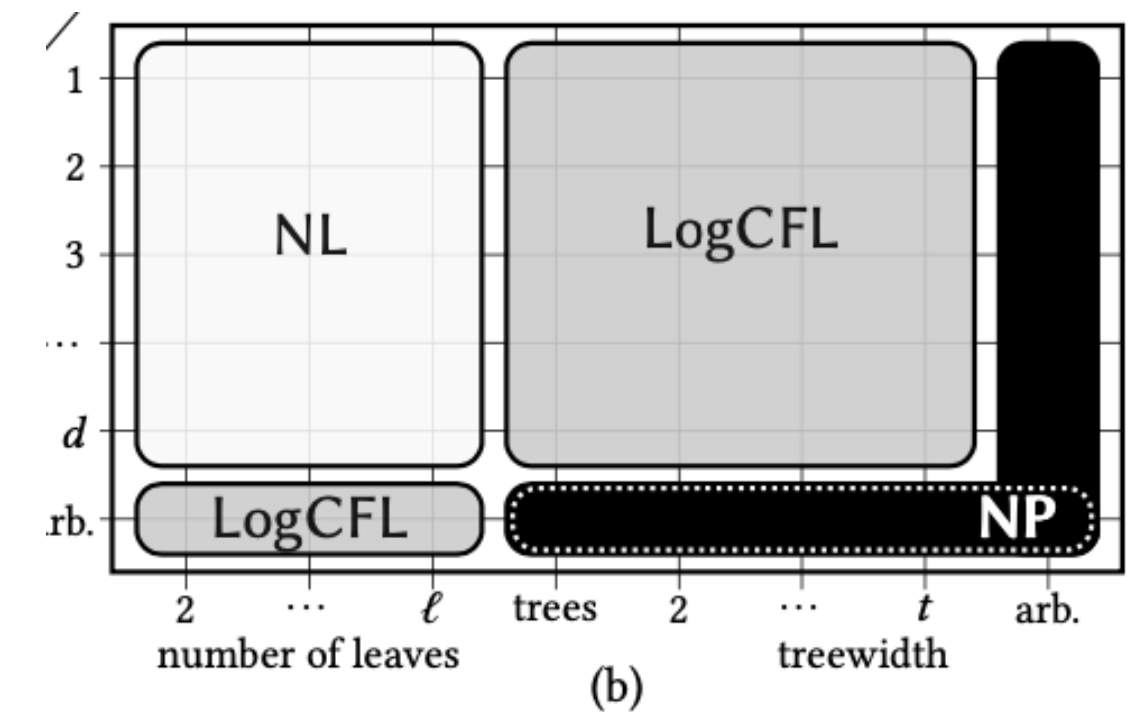


Figure 6: Complexity of basic *DL-Lite* logics.

	CQ		C(2)RPQ	
	data	combined	data	combined
DL-Lite _{RDFS}	in AC ⁰	NP-c ≥ (B)	NL-c ≤ Thm 6.8	NP-c ≤ Thm 6.8
DL-Lite _(R)	in AC ⁰ ≤ (D)	NP-c ≤ (D)	NL-c ≤ Thm 6.8	PSpace-c ≥ Prop 4.5, ≤ Thm 6.8
EL(H)	P-c ≤ (F)	NP-c ≤ (F)	P-c ≤ Thm 6.8	PSpace-c ≥ Prop 4.5, ≤ Thm 6.8

Class	arity unbounded	arity bounded	Data Complexity
gbts	in 3EXP ^{TIME} *	2EXP ^{TIME} -c *	EXP ^{TIME} -c *
wfg	2EXP ^{TIME} -c *	2EXP ^{TIME} -c *	EXP ^{TIME} -c *
fg	2EXP ^{TIME} -c *	2EXP ^{TIME} -c *	P ^{TIME} -c *
fr1	2EXP ^{TIME} -c *	2EXP ^{TIME} -c *	P ^{TIME} -c *
wg	2EXP ^{TIME} -c	EXP ^{TIME} -c	EXP ^{TIME} -c
guarded	2EXP ^{TIME} -c	EXP ^{TIME} -c	P ^{TIME} -c
ba-fg	2EXP ^{TIME} -c *	EXP ^{TIME} -c *	P ^{TIME} -c *
ba-fr1	EXP ^{TIME} -hard ⁽¹⁾	EXP ^{TIME} -c *	P ^{TIME} -c *

	IQs		CQs	
	data complexity	combined complexity	data complexity	combined complexity
DL-Lite	in AC ₀	NLOGSPACE	in AC ₀	NP
DL-Lite _R	in AC ₀	NLOGSPACE	in AC ₀	NP
EL, ELH	P	P	P	NP
ELI, ELHI _⊥ , Horn-SHOIQ	P	EXP	P	EXP
ALC, ALCHQ	coNP	EXP	coNP	EXP
ALCI, SH, SHIQ	coNP	EXP	coNP	2EXP
SHOIQ	coNP	coNEXP	coNP-hard ¹	coN2EXP-hard ¹



Complexity landscape: Takeaways

Small differences in expressivity can lead to significant differences in complexity

Complexity landscape: Takeaways

Small differences in expressivity can lead to significant differences in complexity

Useful to distinguish between combined complexity and data complexity

- combined complexity: measured in terms of whole input
- data complexity: measured in terms of size of dataset (i.e. treat ontology, query as fixed)
- allows to identify more tractable fragments (e.g. can be EXPTIME combined, but PTIME for data complexity)

Complexity landscape: Takeaways

Small differences in expressivity can lead to significant differences in complexity

Useful to distinguish between combined complexity and data complexity

- combined complexity: measured in terms of whole input
- data complexity: measured in terms of size of dataset (i.e. treat ontology, query as fixed)
- allows to identify more tractable fragments (e.g. can be EXPTIME combined, but PTIME for data complexity)

PTIME data complexity only possible for languages without any form of disjunction

- reasoning by cases leads to coNP data complexity (or worse)

Complexity landscape: Takeaways

Small differences in expressivity can lead to significant differences in complexity

Useful to distinguish between combined complexity and data complexity

- combined complexity: measured in terms of whole input
- data complexity: measured in terms of size of dataset (i.e. treat ontology, query as fixed)
- allows to identify more tractable fragments (e.g. can be EXPTIME combined, but PTIME for data complexity)

PTIME data complexity only possible for languages without any form of disjunction

- reasoning by cases leads to coNP data complexity (or worse)

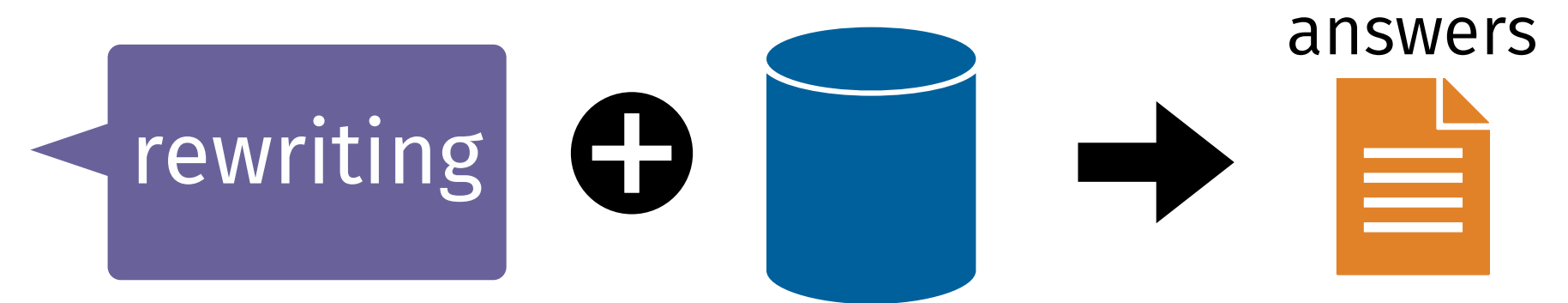
Most work on OMQA algorithms focuses on settings with PTIME data complexity

Query rewriting

Query Rewriting *Reduce OMQA to standard database query evaluation*



Rewriting step



Evaluation step (database system)

Query rewriting

Query Rewriting *Reduce OMQA to standard database query evaluation*



Formally: **rewriting** of (\mathcal{O}, q) is a **query** q^* such that **for every dataset**,

$$(\mathcal{O}, \mathcal{D}) \models q(\vec{a}) \iff \vec{a} \in \text{ans}(q^*, \mathcal{D})$$

certain answers to q on $(\mathcal{O}, \mathcal{D})$

answers to q^* on \mathcal{D}

Query rewriting

Query Rewriting *Reduce OMQA to standard database query evaluation*



Formally: **rewriting** of (\mathcal{O}, q) is a **query** q^* such that **for every dataset**,

$$(\mathcal{O}, \mathcal{D}) \models q(\vec{a}) \iff \vec{a} \in \text{ans}(q^*, \mathcal{D})$$

certain answers to q on $(\mathcal{O}, \mathcal{D})$ answers to q^* on \mathcal{D}

Usually focus on **first-order rewritings**: q^* is a **first-order (~SQL) query** (can evaluate using relational DB)

Query rewriting

Query Rewriting *Reduce OMQA to standard database query evaluation*



Formally: **rewriting** of (\mathcal{O}, q) is a **query** q^* such that **for every dataset**,

$$(\mathcal{O}, \mathcal{D}) \models q(\vec{a}) \iff \vec{a} \in \text{ans}(q^*, \mathcal{D})$$

certain answers to q on $(\mathcal{O}, \mathcal{D})$ answers to q^* on \mathcal{D}

Usually focus on **first-order rewritings**: q^* is a **first-order (~SQL) query** (can evaluate using relational DB)

Many **implemented rewriting algorithms**, e.g. **Ontop** (DL-Lite, OWL 2 QL), **Graal** (existential rules)

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

First-order rewriting of (\mathcal{O}, q_1) : "find those who are a PhD holder, or a prof, or a postdoc, or teach some grad course"

$$q^*(x) = \text{PhDHolder}(x) \vee \text{Prof}(x) \vee \text{Postdoc}(x) \vee (\exists z \text{ taughtBy}(z, x) \wedge \text{GradCourse}(z))$$

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

First-order rewriting of (\mathcal{O}, q_1) : "find those who are a PhD holder, or a prof, or a postdoc, or teach some grad course"

$$q^*(x) = \text{PhDHolder}(x) \vee \text{Prof}(x) \vee \text{Postdoc}(x) \vee (\exists z \text{ taughtBy}(z, x) \wedge \text{GradCourse}(z))$$

Evaluating the rewriting over the dataset

Prof(kim)

Postdoc(paolo)

taughtBy(cs87, ann)

GradCourse(cs87)

gives us the **three certain answers**:

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

First-order rewriting of (\mathcal{O}, q_1) : "find those who are a PhD holder, or a prof, or a postdoc, or teach some grad course"

$$q^*(x) = \text{PhDHolder}(x) \vee \text{Prof}(x) \vee \text{Postdoc}(x) \vee (\exists z \text{ taughtBy}(z, x) \wedge \text{GradCourse}(z))$$

Evaluating the rewriting over the dataset

Prof(kim)

Postdoc(paolo)

taughtBy(cs87, ann)

GradCourse(cs87)

gives us the three certain answers: kim

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

First-order rewriting of (\mathcal{O}, q_1) : "find those who are a PhD holder, or a prof, or a postdoc, or teach some grad course"

$$q^*(x) = \text{PhDHolder}(x) \vee \text{Prof}(x) \vee \text{Postdoc}(x) \vee (\exists z \text{ taughtBy}(z, x) \wedge \text{GradCourse}(z))$$

Evaluating the rewriting over the dataset

Prof(kim)

Postdoc(paolo)

taughtBy(cs87, ann)

GradCourse(cs87)

gives us the three certain answers: kim paolo

Example: Query rewriting

Consider the following ontology:

$$\mathcal{O} = \begin{array}{l} \text{Prof}(x) \rightarrow \text{PhDHolder}(x) \qquad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y) \\ \text{Postdoc}(x) \rightarrow \text{PhDHolder}(x) \end{array}$$

and reconsider the query $q_1(x) = \text{PhDHolder}(x)$

First-order rewriting of (\mathcal{O}, q_1) : "find those who are a PhD holder, or a prof, or a postdoc, or teach some grad course"

$$q^*(x) = \text{PhDHolder}(x) \vee \text{Prof}(x) \vee \text{Postdoc}(x) \vee (\exists z \text{ taughtBy}(z, x) \wedge \text{GradCourse}(z))$$

Evaluating the rewriting over the dataset

Prof(kim)

Postdoc(paolo)

taughtBy(cs87, ann)

GradCourse(cs87)

gives us the three certain answers: kim paolo ann

Query Rewriting: Limits and Possibilities

Cost of query rewriting

Rewriting algorithms can produce *huge rewritings* - is this *unavoidable*?

Study *succinctness* of rewritings

Setting: OWL 2 QL (DL-Lite) ontology + conjunctive query

Dimensions: query + ont. structure, form of rewriting

- Exponential size *blowup unavoidable in general*
- *Tractable classes: guaranteed short rewritings*, efficient evaluation

Novel, non-trivial connections to *circuit complexity*

Bienvenu et al. Ontology-Mediated Queries: Combined Complexity and Succinctness of Rewritings via Circuit Complexity. JACM 2018

Query Rewriting: Limits and Possibilities

Cost of query rewriting

Rewriting algorithms can produce *huge rewritings* - is this *unavoidable*?

Study *succinctness* of rewritings

Setting: OWL 2 QL (DL-Lite) ontology + conjunctive query

Dimensions: query + ont. structure, form of rewriting

- Exponential size *blowup unavoidable in general*
- *Tractable classes: guaranteed short rewritings*, efficient evaluation

Novel, non-trivial connections to *circuit complexity*

Bienvenu et al. Ontology-Mediated Queries: Combined Complexity and Succinctness of Rewritings via Circuit Complexity. JACM 2018

(In)existence of rewritings

In many ontology languages, FO-rewritings need not exist. End of story?

Task: decide if rewriting exists for a particular ontology & query

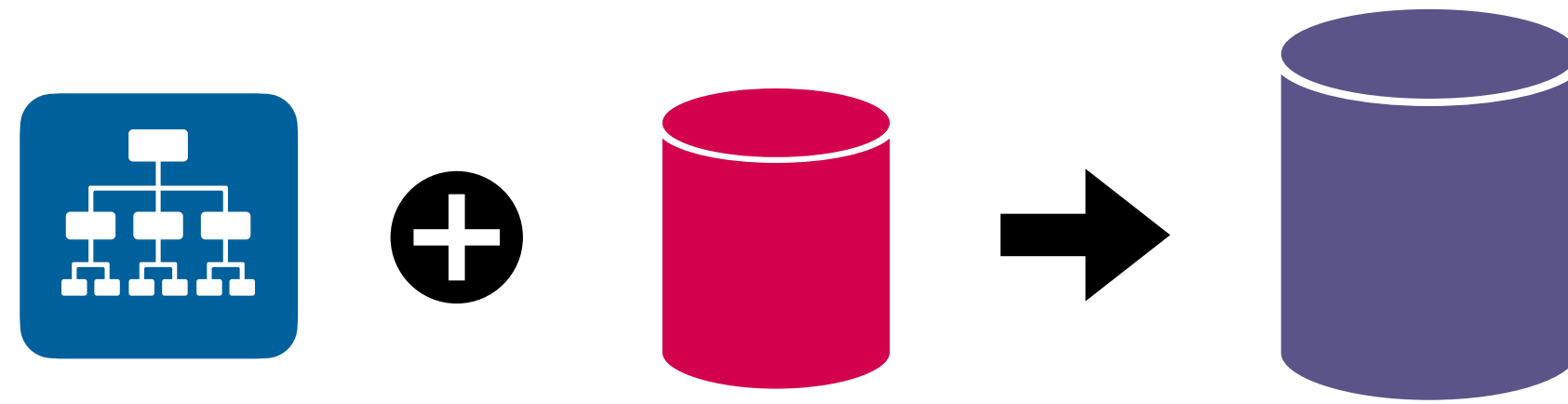
- *Characterizations, precise complexity, algorithms* for various DLs
- Basis for *extending the reach of query rewriting* approach

Bienvenu et al. First Order-Rewritability and Containment of Conjunctive Queries in Horn Description Logics. IJCAI 2016

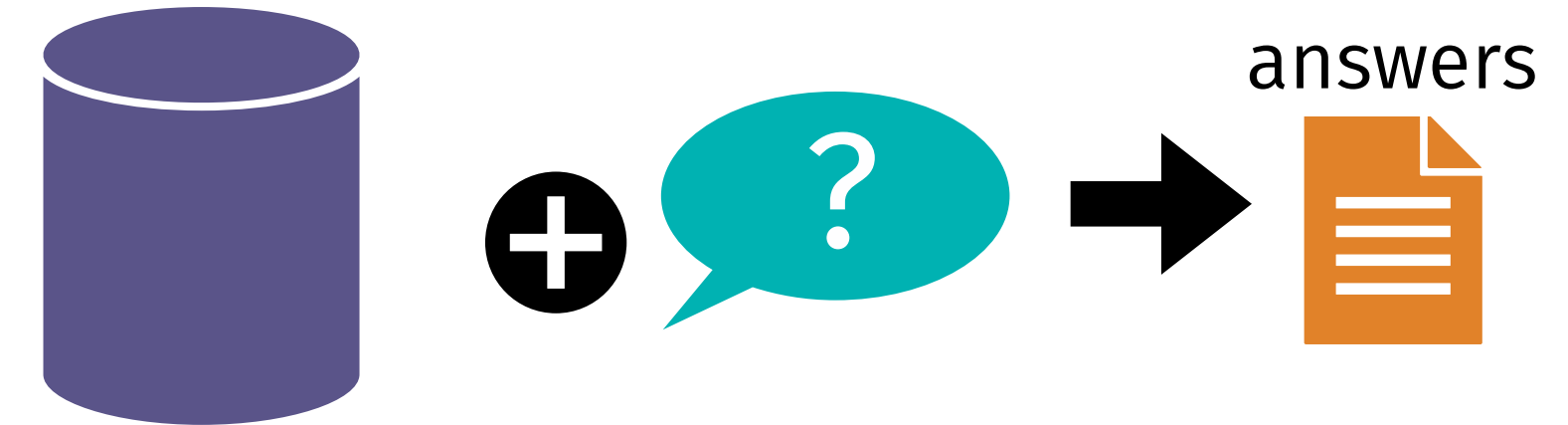
Materialization-based approaches

Materialization

Complete the data with implicit information from the ontology



Materialization step

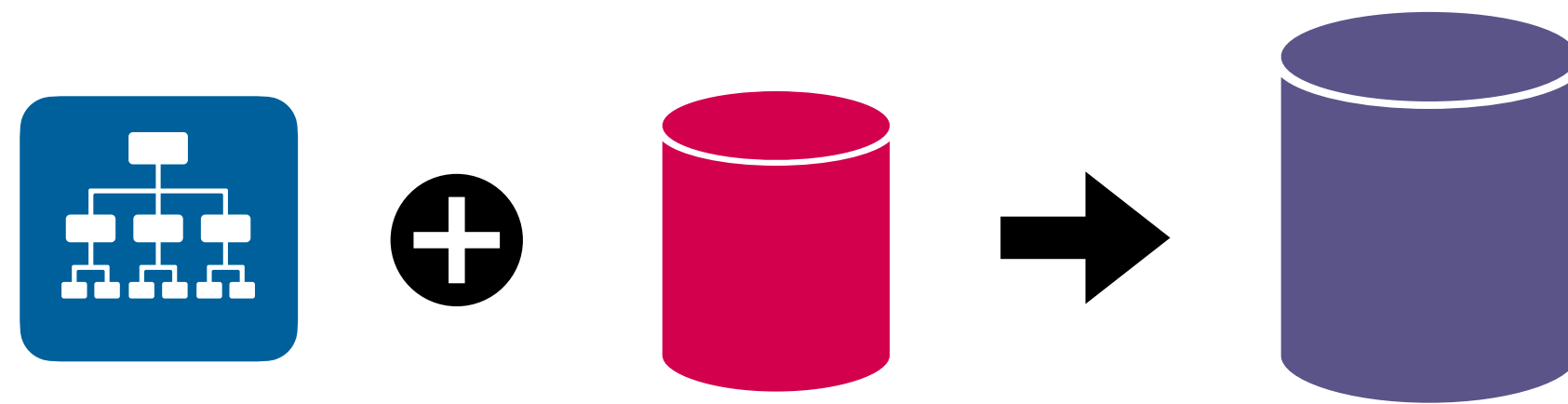


Evaluation step
(e.g. using database system)

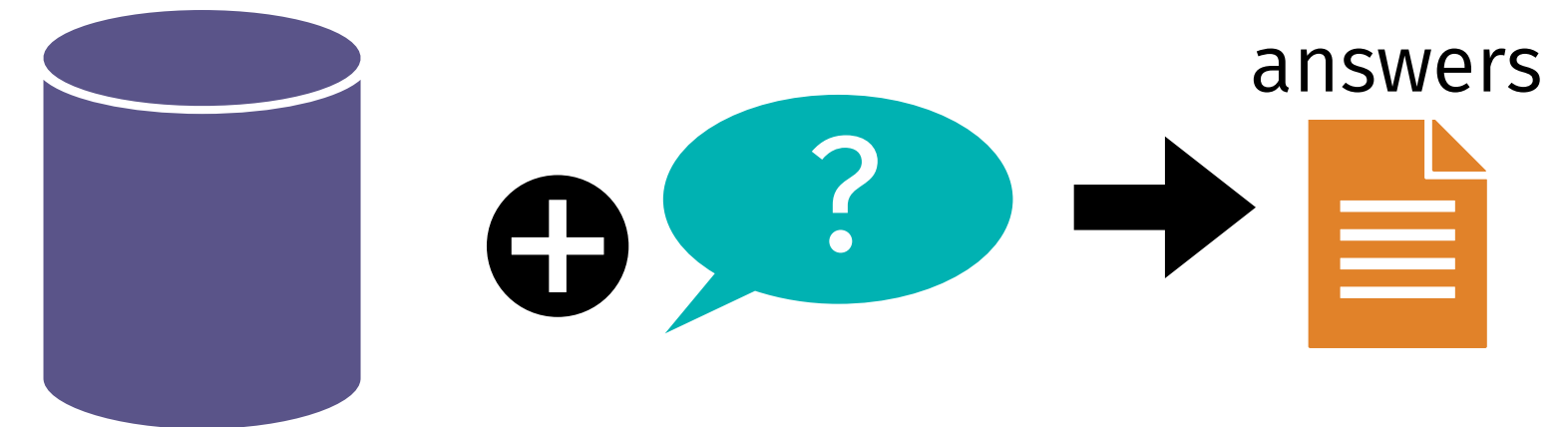
Materialization-based approaches

Materialization

Complete the data with implicit information from the ontology

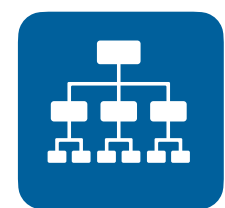


Materialization step



Evaluation step
(e.g. using database system)

Illustration with running example:



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{Postdoc}(x) \rightarrow \text{PhDHolder}(x)$
 $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



$\text{Prof}(\text{kim})$ $\text{Postdoc}(\text{paolo})$ $\text{taughtBy}(\text{cs87}, \text{ann})$ $\text{GradCourse}(\text{cs87})$

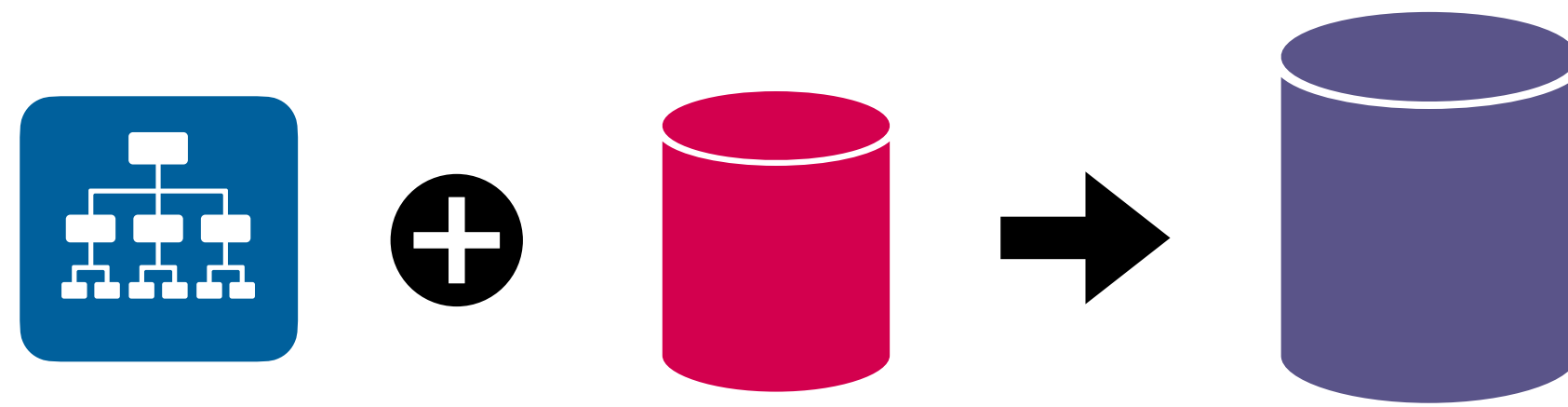
$q_1(x) = \text{PhDHolder}(x)$

kim

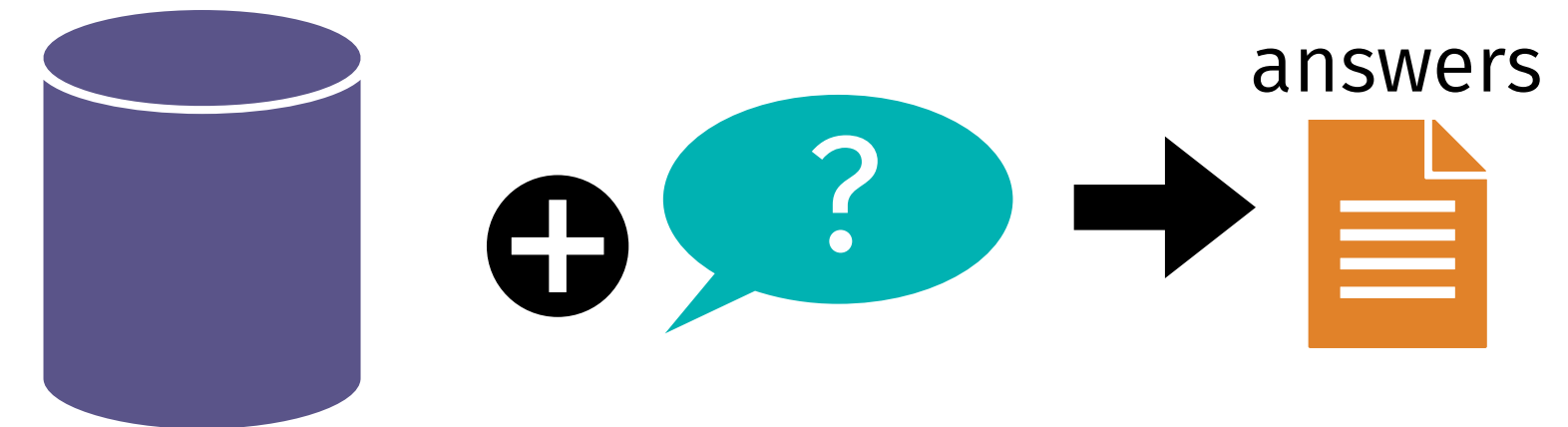
Materialization-based approaches

Materialization

Complete the data with implicit information from the ontology

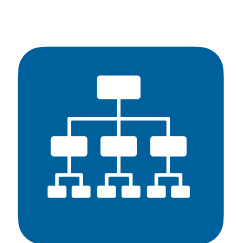


Materialization step



Evaluation step
(e.g. using database system)

Illustration with running example:

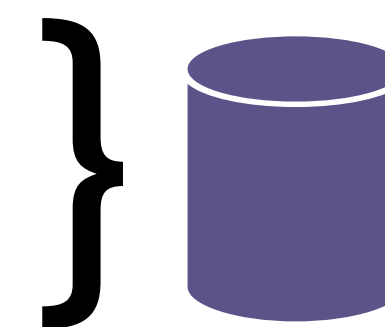


$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{Postdoc}(x) \rightarrow \text{PhDHolder}(x)$
 $\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



$\text{Prof}(\text{kim})$ $\text{Postdoc}(\text{paolo})$ $\text{taughtBy}(\text{cs87}, \text{ann})$ $\text{GradCourse}(\text{cs87})$

$\text{PhDHolder}(\text{kim})$ $\text{PhDHolder}(\text{paolo})$ $\text{Prof}(\text{ann})$ $\text{PhDHolder}(\text{ann})$



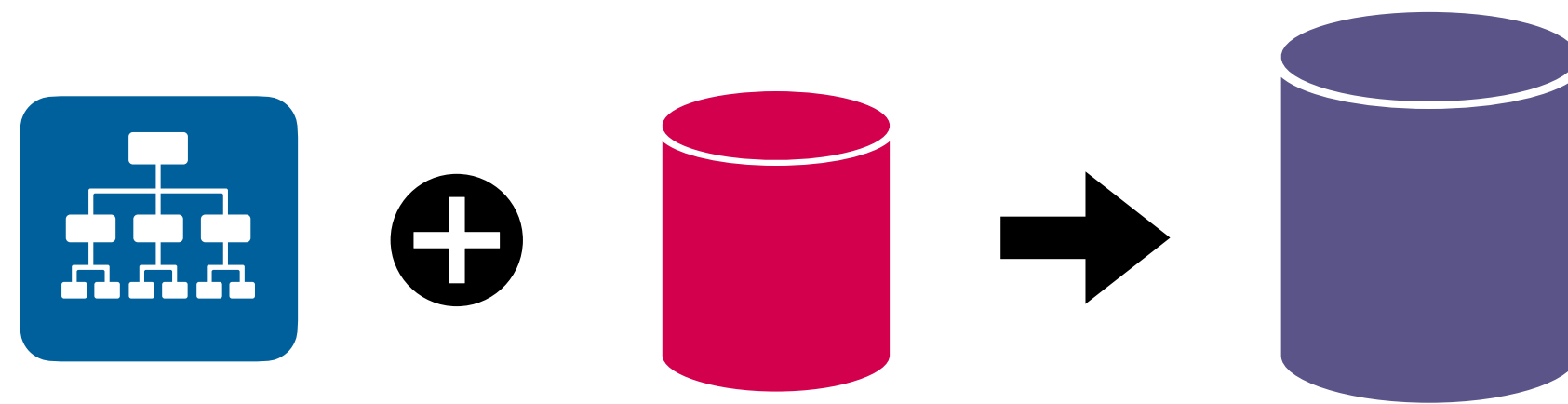
$q_1(x) = \text{PhDHolder}(x)$

kim

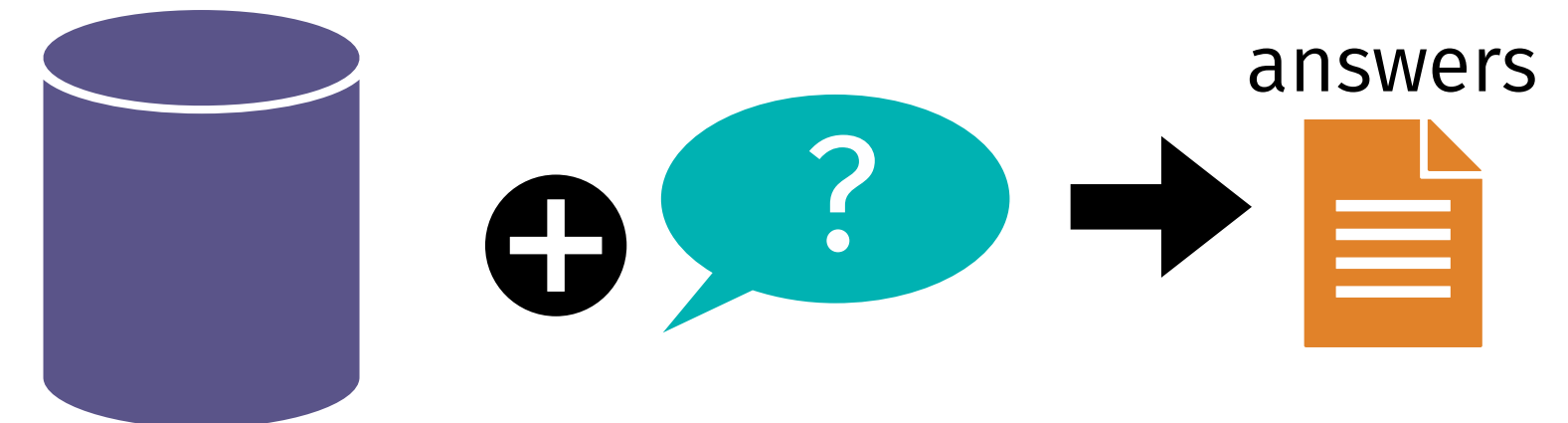
Materialization-based approaches

Materialization

Complete the data with implicit information from the ontology

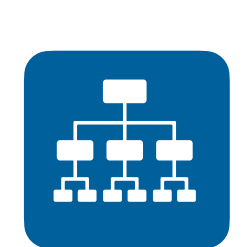


Materialization step



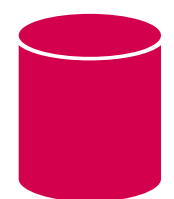
Evaluation step
(e.g. using database system)

Illustration with running example:



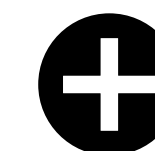
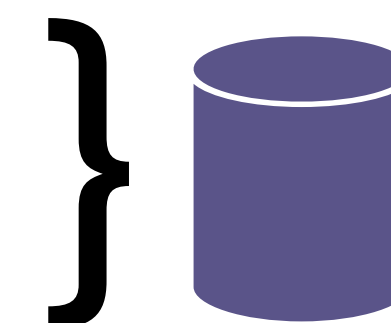
$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$ $\text{Postdoc}(x) \rightarrow \text{PhDHolder}(x)$

$\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

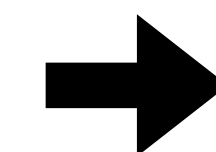


$\text{Prof}(\text{kim})$ $\text{Postdoc}(\text{paolo})$ $\text{taughtBy}(\text{cs87}, \text{ann})$ $\text{GradCourse}(\text{cs87})$

$\text{PhDHolder}(\text{kim})$ $\text{PhDHolder}(\text{paolo})$ $\text{Prof}(\text{ann})$ $\text{PhDHolder}(\text{ann})$



$q_1(x) = \text{PhDHolder}(x)$



kim
paolo
ann

Chase: Handling Unnamed Individuals

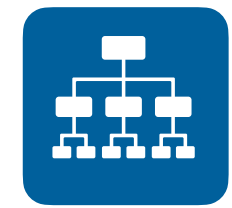
Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**

Chase: Handling Unnamed Individuals

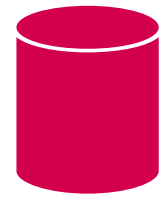
Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$

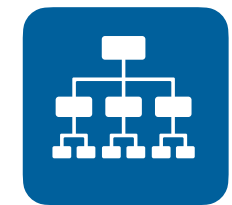


$\text{GradCourse}(\text{cs87})$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$

$\text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y)$

$\text{GradCourse}(x) \rightarrow \text{Course}(x)$

$\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



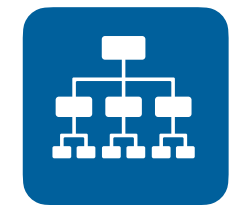
$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87})$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$

$\text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y)$

$\text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



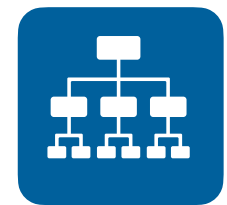
$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87}) \text{ taughtBy}(\text{cs87}, \alpha)$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \boxed{\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)}$



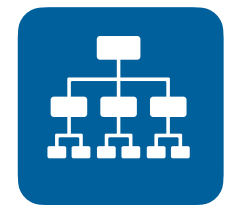
$\boxed{\text{GradCourse}(\text{cs87})}$

$\text{Course}(\text{cs87}) \quad \boxed{\text{taughtBy}(\text{cs87}, \alpha)} \quad \text{Prof}(\alpha)$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$

$\text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y)$

$\text{GradCourse}(x) \rightarrow \text{Course}(x)$

$\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87}) \quad \text{taughtBy}(\text{cs87}, \alpha)$

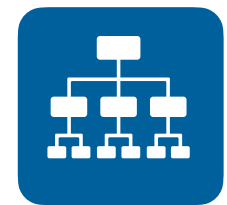
$\text{Prof}(\alpha)$

$\text{PhDHolder}(\alpha)$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x)$

$\text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y)$

$\text{GradCourse}(x) \rightarrow \text{Course}(x)$

$\text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87})$

$\text{taughtBy}(\text{cs87}, \alpha)$

$\text{Prof}(\alpha)$

$\text{PhDHolder}(\alpha)$

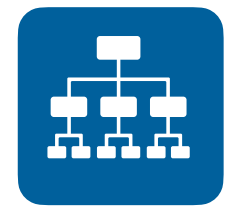
$\text{hasDegree}(\alpha, \beta)$

$\text{PhD}(\beta)$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



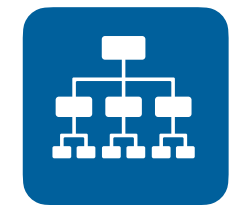
GradCourse(cs87)

$\text{Course}(\text{cs87}) \quad \text{taughtBy}(\text{cs87}, \alpha) \quad \text{Prof}(\alpha) \quad \text{PhDHolder}(\alpha) \quad \text{hasDegree}(\alpha, \beta) \quad \text{PhD}(\beta)$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



$\text{GradCourse}(\text{cs87})$

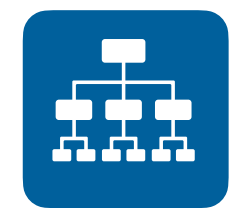
$\underbrace{\text{Course}(\text{cs87}) \text{ taughtBy}(\text{cs87}, \alpha) \text{ Prof}(\alpha) \text{ PhDHolder}(\alpha) \text{ hasDegree}(\alpha, \beta) \text{ PhD}(\beta)}$

$\text{Chase}_{\mathcal{O}}(\mathcal{D})$

Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

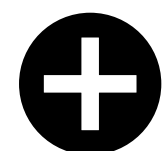
$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



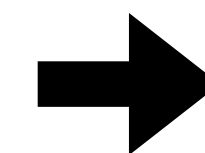
$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87}) \quad \text{taughtBy}(\text{cs87}, \alpha) \quad \text{Prof}(\alpha) \quad \text{PhDHolder}(\alpha) \quad \text{hasDegree}(\alpha, \beta) \quad \text{PhD}(\beta)$

$\text{Chase}_{\mathcal{O}}(\mathcal{D})$



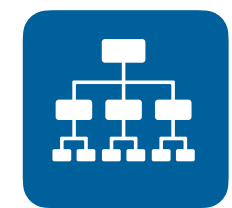
$q_3(x) = \exists y \text{taughtBy}(x, y) \wedge \text{PhDHolder}(y)$



Chase: Handling Unnamed Individuals

Materialization can also be employed for **existential rules**, using **chase procedure**:

- **apply rules exhaustively**, introduce **fresh elements** to serve as witnesses for **existential vars**



$\text{Prof}(x) \rightarrow \text{PhDHolder}(x) \quad \text{PhDHolder}(x) \rightarrow \exists z \text{HasDegree}(x, z) \wedge \text{PhD}(z)$

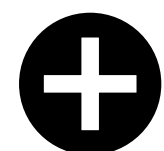
$\text{Course}(x) \rightarrow \exists y \text{taughtBy}(x, y) \quad \text{GradCourse}(x) \rightarrow \text{Course}(x) \quad \text{GradCourse}(x) \wedge \text{taughtBy}(x, y) \rightarrow \text{Prof}(y)$



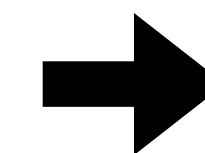
$\text{GradCourse}(\text{cs87})$

$\text{Course}(\text{cs87}) \quad \text{taughtBy}(\text{cs87}, \alpha) \quad \text{Prof}(\alpha) \quad \text{PhDHolder}(\alpha) \quad \text{hasDegree}(\alpha, \beta) \quad \text{PhD}(\beta)$

$\text{Chase}_{\mathcal{O}}(\mathcal{D})$



$q_3(x) = \exists y \text{taughtBy}(x, y) \wedge \text{PhDHolder}(y)$



Key property:

$(\mathcal{O}, \mathcal{D}) \models q(\vec{a}) \iff \vec{a} \in \text{ans}(q, \text{Chase}_{\mathcal{O}}(\mathcal{D}))$

certain answer

answer on chased dataset

*applies to conjunctive queries,
ontologies given by existential rules*

Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

When the **chase is finite**, it provides a **method for computing certain answers**

Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?

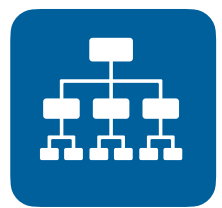
Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?



$\text{PhDHolder}(x) \rightarrow \exists z \text{ hasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$



$\text{PhDHolder}(\text{ming})$

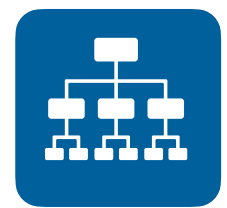
Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

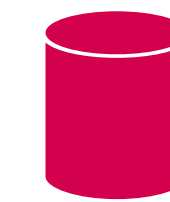
When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?



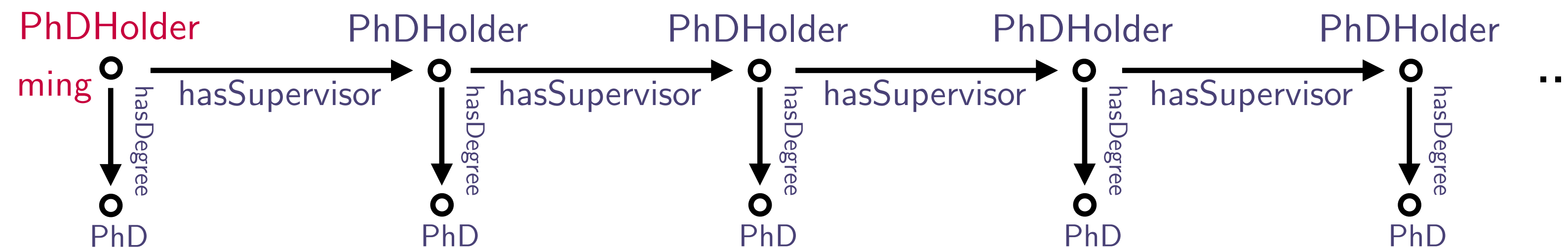
$\text{PhDHolder}(x) \rightarrow \exists z \text{ hasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$



$\text{PhDHolder}(\text{ming})$

$\text{Chase}_{\mathcal{O}}(\mathcal{D})$



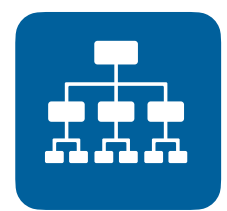
Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

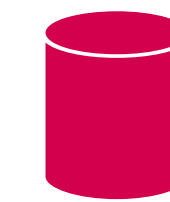
When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?

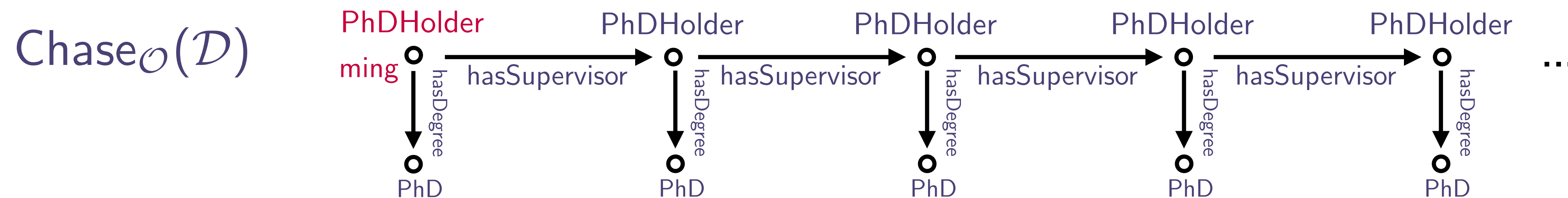


$\text{PhDHolder}(x) \rightarrow \exists z \text{ hasDegree}(x, z) \wedge \text{PhD}(z)$

$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$



$\text{PhDHolder}(\text{ming})$



Sometimes possible to use **compact representation of chase** (called **combined approach**)

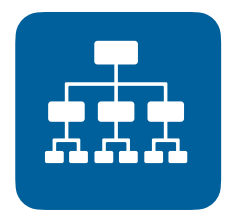
Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

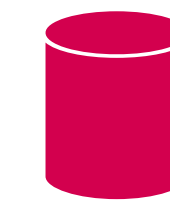
When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?

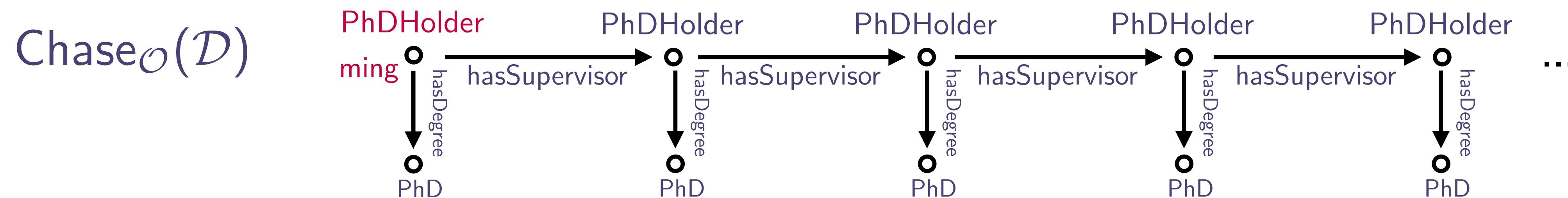


$\text{PhDHolder}(x) \rightarrow \exists z \text{ hasDegree}(x, z) \wedge \text{PhD}(z)$

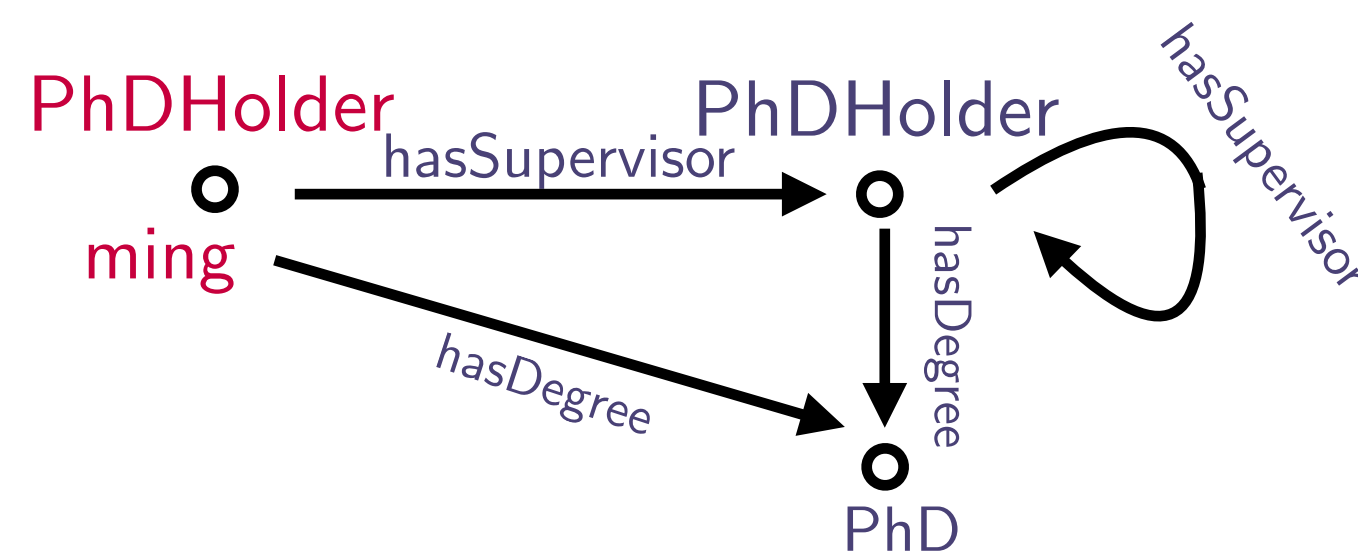
$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$



$\text{PhDHolder}(\text{ming})$



Sometimes possible to use **compact representation of chase** (called **combined approach**)



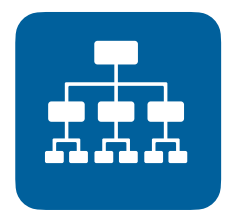
Combined Approach

Lutz et al. CQ Answering in the Description Logic
EL Using a Relational Database System. IJCAI'09

Carral et al. The Combined Approach to
Query Answering in Horn-ALCHOIQ. KR'18

When the **chase is finite**, it provides a **method for computing certain answers**

What about cases where the **chase is infinite**?

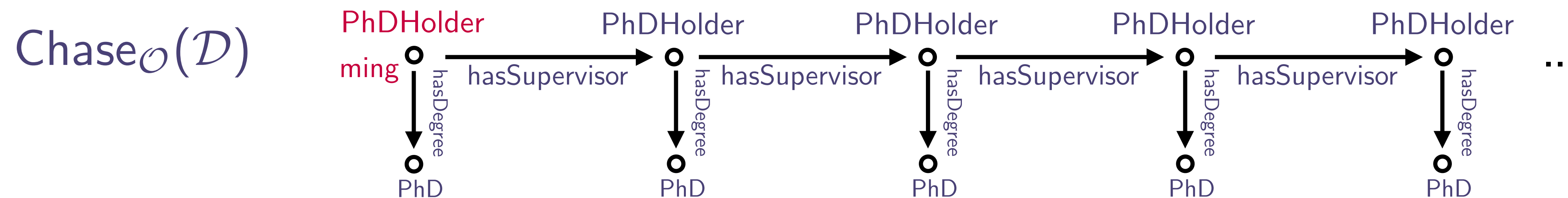


$\text{PhDHolder}(x) \rightarrow \exists z \text{ hasDegree}(x, z) \wedge \text{PhD}(z)$

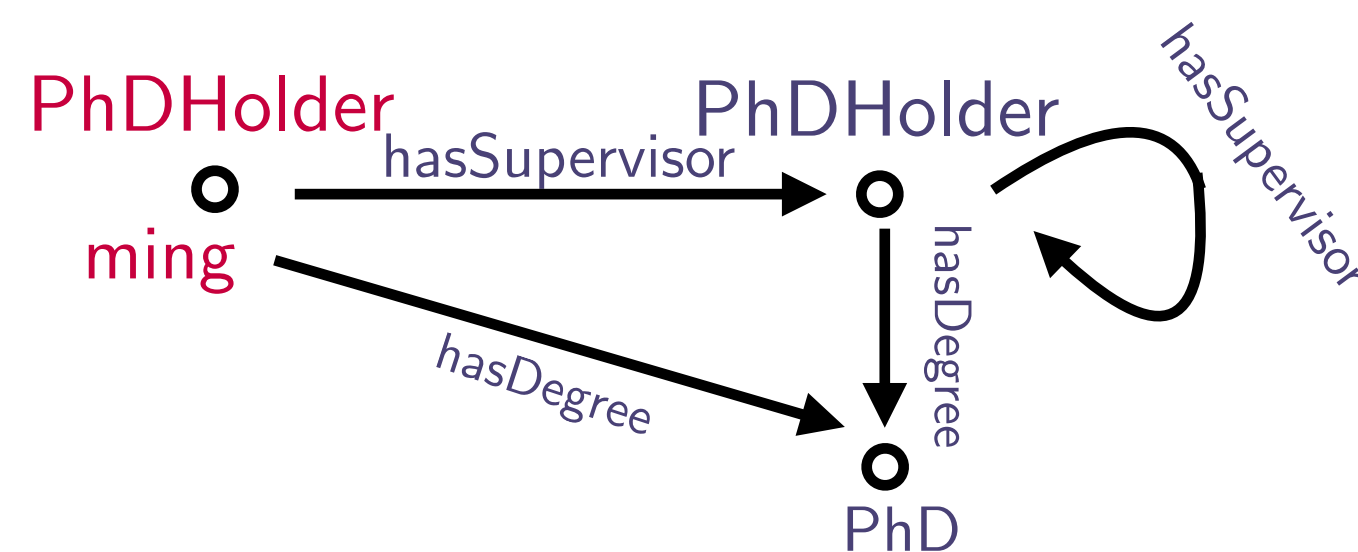
$\text{PhDHolder}(x) \rightarrow \exists y \text{ hasSupervisor}(x, y) \wedge \text{PhDHolder}(y)$



$\text{PhDHolder}(\text{ming})$



Sometimes possible to use **compact representation of chase** (called **combined approach**)



- evaluate original **query over compact chase**
 \Rightarrow **superset of certain answers**
- **rewriting / filtering phase** to block incorrect answers

Closer looks at materialization

When does
the chase
terminate?

Which classes of existential rules guarantee chase termination?

- Development of **acyclicity conditions** (sufficient) (over any dataset)
- Study of **decision problem**: check if given ruleset has terminating chase
- Several **different chase variants** (oblivious, Skolem, restricted, core, ...)

Carral & Urbani. Checking Chase Termination over Ontologies of Existential Rules with Equality. AAAI 2020.

Gogacz et al. All-Instances Restricted Chase Termination. PODS 2020.

Closer looks at materialization

When does
the chase
terminate?

Which classes of existential rules guarantee chase termination?

- Development of **acyclicity conditions** (sufficient) (over any dataset)
- Study of **decision problem**: check if given ruleset has terminating chase
- Several **different chase variants** (oblivious, Skolem, restricted, core, ...)

Carral & Urbani. Checking Chase Termination over Ontologies of Existential Rules with Equality. AAAI 2020.

Gogacz et al. All-Instances Restricted Chase Termination. PODS 2020.

Efficient
chase
computation

*How to **efficiently compute and maintain** the materialization?*

OMQA motivates need to develop efficient algorithms for Datalog, chase

New techniques & optimizations: handling **updates, parallelization**

- Example systems: **RDFox** (Datalog, OWL 2 RL), **VLog**, **Graal** (exist. rules)

OMQA Research

Topics in OMQA research

Expressive power

*Which queries can be captured using **ontology-mediated queries (OMQs)**?*

Bienvenu et al. *Ontology-Based Data Access: A Study through Disjunctive Datalog, CSP, and MMSNP*. TODS 2014

Bourgau et al. *Capturing Homomorphism-Closed Decidable Queries with Existential Rules*. KR 2021

(\mathcal{O}, q)

Topics in OMQA research

Expressive power

*Which queries can be captured using **ontology-mediated queries (OMQs)**?*

Bienvenu et al. *Ontology-Based Data Access: A Study through Disjunctive Datalog, CSP, and MMSNP*. TODS 2014

(\mathcal{O}, q)

Bourgau et al. *Capturing Homomorphism-Closed Decidable Queries with Existential Rules*. KR 2021

Fine-grained Complexity

Classify complexity at level of ontologies, OMQs

Aim: identify more tractable cases, aid in choosing algorithms

Lutz & Sabellek. *Ontology-Mediated Querying with the Description Logic EL: Trichotomy and Linear Datalog Rewritabilit.* IJCAI 2017

Hernich et al. *Dichotomies in Ontology-Mediated Querying with the Guarded Fragment*. TOCL 2020

Topics in OMQA research

Expressive power

*Which queries can be captured using **ontology-mediated queries (OMQs)**?*

Bienvenu et al. *Ontology-Based Data Access: A Study through Disjunctive Datalog, CSP, and MMSNP*. TODS 2014

(\mathcal{O}, q)

Bourgau et al. *Capturing Homomorphism-Closed Decidable Queries with Existential Rules*. KR 2021

Fine-grained Complexity

Classify complexity at level of ontologies, OMQs

Aim: identify more tractable cases, aid in choosing algorithms

Lutz & Sabellek. *Ontology-Mediated Querying with the Description Logic EL: Trichotomy and Linear Datalog Rewritabilit.* IJCAI 2017

Hernich et al. *Dichotomies in Ontology-Mediated Querying with the Guarded Fragment*. TOCL 2020

Approximation

Use approximation to handle more expressive ontology languages

Zhou et al. *PAGOdA: Pay-As-You-Go Ontology Query Answering Using a Datalog Reasoner*. JAIR 2015

Haga et al. *How to Approximate Ontology-Mediated Queries*. KR 2021

Topics in OMQA research

Richer queries,
other data formats

What about *non-relational data*?

graph data, key-value stores, temporal DBs

What about *other kinds of queries*?

path queries, *counting queries*, temporal queries

Bienvenu, [Quentin Manière](#), Thomazo. Cardinality Queries over DL-Lite Ontologies. IJCAI 2021

ANR CQFD (Federico Ulliana, LIRMM + 7 other French labs)

Topics in OMQA research

Richer queries,
other data formats

What about *non-relational data*?

graph data, key-value stores, temporal DBs

What about *other kinds of queries*? path queries, *counting queries*, temporal queries

Bienvenu, [Quentin Manière](#), Thomazo. Cardinality Queries over DL-Lite Ontologies. IJCAI 2021

ANR CQFD (Federico Ulliana, LIRMM + 7 other French labs)

Imperfect data &
knowledge

How to cope with *erroneous, imprecise, uncertain data*?

Bienvenu. A Short Survey on Inconsistency Handling in Ontology-Mediated Query Answering. KI 2020

INTENDED AI Chair (2020-2025): Intelligent Handling of Imperfect Data

Topics in OMQA research

Richer queries,
other data formats

What about *non-relational data*?

graph data, key-value stores, temporal DBs

What about *other kinds of queries*?

path queries, *counting queries*, temporal queries

Bienvenu, [Quentin Manière](#), Thomazo. Cardinality Queries over DL-Lite Ontologies. IJCAI 2021

ANR CQFD (Federico Ulliana, LIRMM + 7 other French labs)

Imperfect data &
knowledge

How to cope with *erroneous, imprecise, uncertain data*?

Bienvenu. A Short Survey on Inconsistency Handling in Ontology-Mediated Query Answering. KI 2020

INTENDED AI Chair (2020-2025): Intelligent Handling of Imperfect Data

Explanation

How to *explain / justify / trace back query results*?

Learning

How to *learn ontologies, queries, mappings*?

CQFD - Complex ontological Queries over Federated and heterogeneous Data

lirmm.fr/cqfd

Questions?

INTENDED

› Intelligent handling of imperfect information

We're hiring!

Internships, PhD, & postdoc positions

intended.labri.fr/hiring