

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Supervised by Sebastian Rudolph (TU Dresden) and Emanuel Kieroński (University of Wrocław)

Bartosz Jan Bednarczyk



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**



Uniwersytet  
Wrocławski



European Research Council

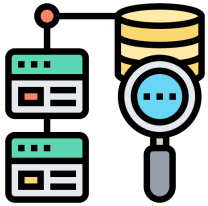
Established by the European Commission

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

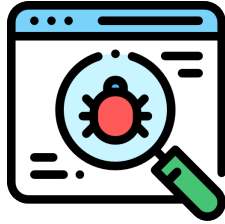
Supervised by Sebastian Rudolph (TU Dresden) and Emanuel Kieroński (University of Wrocław)

Bartosz Jan Bednarczyk

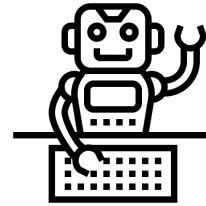
Query languages?



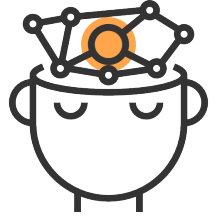
Formal verification?



Formal languages?



Complexity?



*Serious Math Inside*

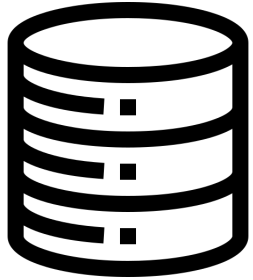
**Warning!** More **mathematics** than computer science is coming!



# Our Setting: Incomplete Databases, Description Logics, and Queries

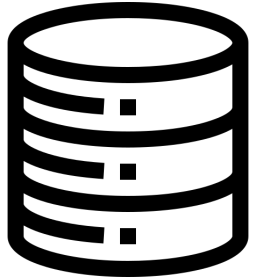
# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)



# Our Setting: Incomplete Databases, Description Logics, and Queries

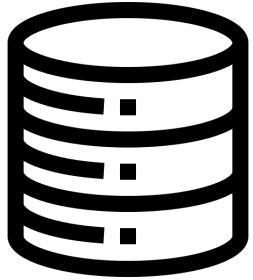
Incomplete Database (ABox)



Wizard(HarryPotter)

# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)

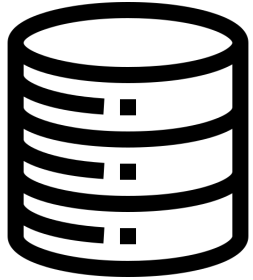


Wizard(HarryPotter)

MuggleBorn(HermioneGranger)

# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)



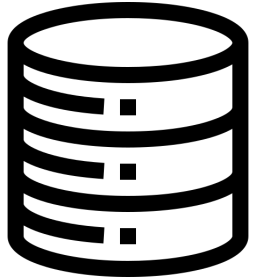
Wizard(HarryPotter)

MuggleBorn(HermioneGranger)

friends(HarryPotter, RonWeasley)

## Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)



Wizard(HarryPotter)

MuggleBorn(HermioneGranger)

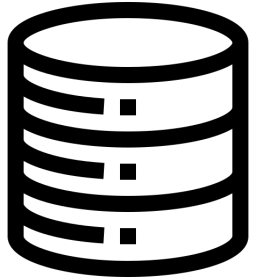
friends(HarryPotter, RonWeasley)

parent(LilyPotter, HarryPotter)



# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)



Wizard(HarryPotter)

MuggleBorn(HermioneGranger)

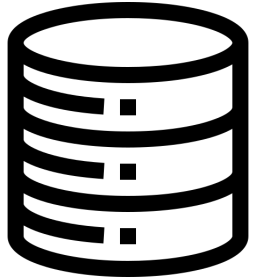
friends(HarryPotter, RonWeasley)

parent(LilyPotter, HarryPotter)

...

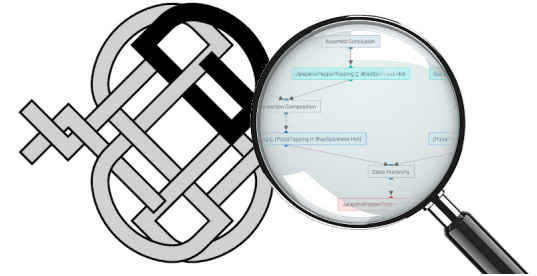
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



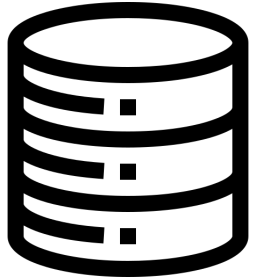
Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## External Knowledge (TBox)



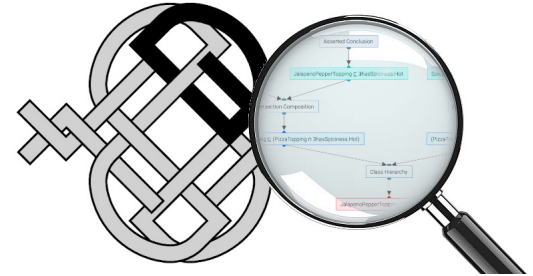
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

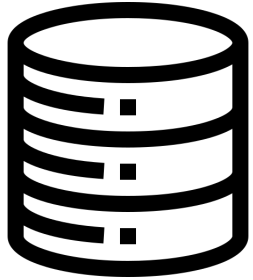
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”

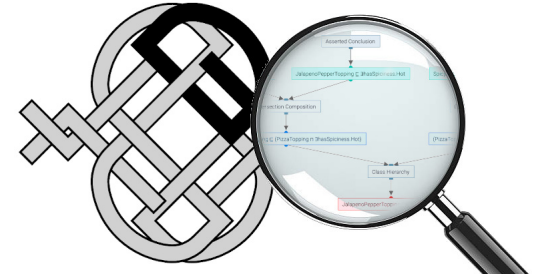
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## External Knowledge (TBox)

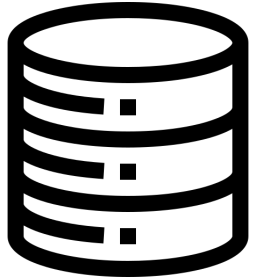


“A Muggle-born person has only non-wizard parents.”

$\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$

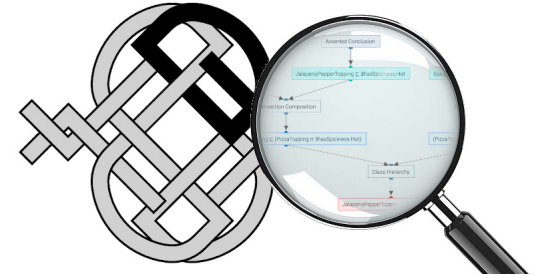
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

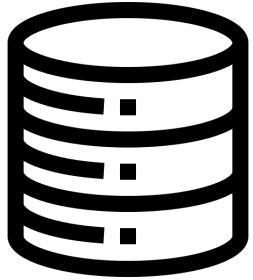
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
$$\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$$
“Voldemort is a descendant of Salazar Slytherin.”

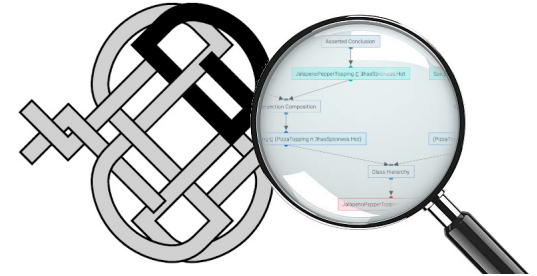
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

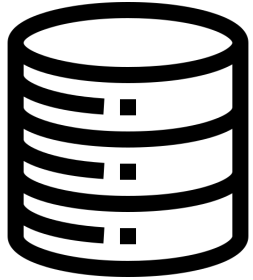
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
 $\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$

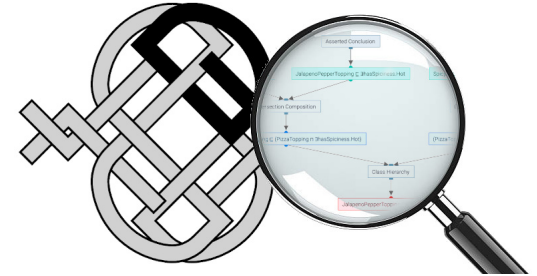
# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

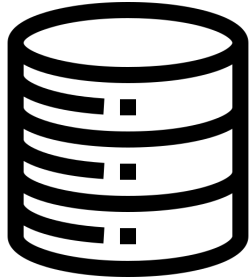
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
 $\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$   
...

# Our Setting: Incomplete Databases, Description Logics, and Queries

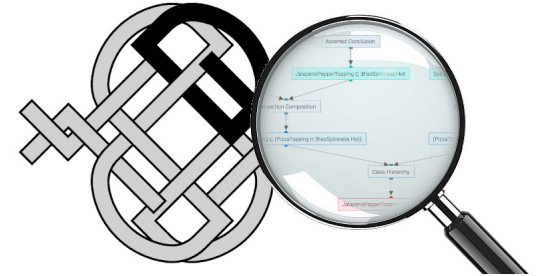
Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

Why?

External Knowledge (TBox)

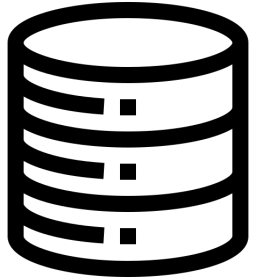


“A Muggle-born person has only non-wizard parents.”  
 $MuggleBorn \sqsubseteq \forall parent. \neg Wizard$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{Voldemort\} \sqsubseteq \exists parent^+. \{SalazarSlytherin\}$   
...



# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)

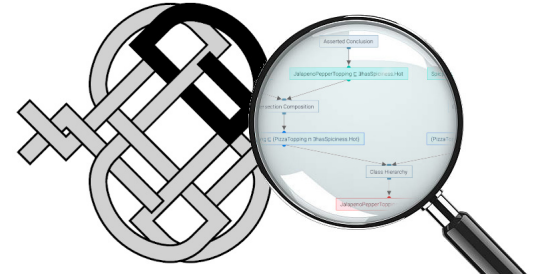


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

Why?

1. Real data can be too big.

External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”

$\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$

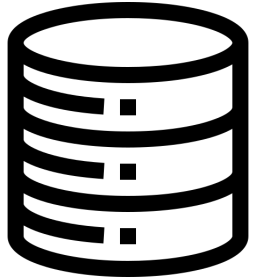
“Voldemort is a descendant of Salazar Slytherin.”

$\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$

...

# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)

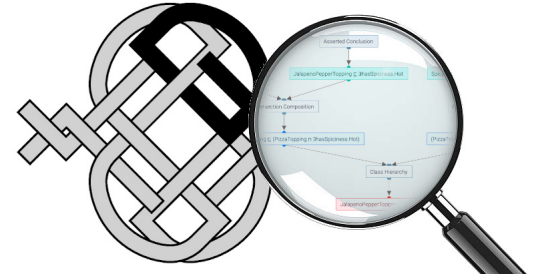


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.

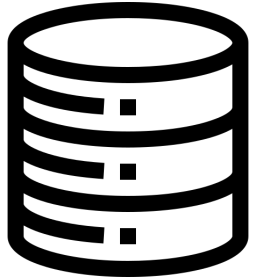
External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
 $\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$   
...

# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)

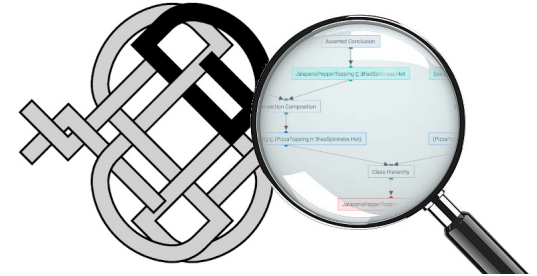


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.

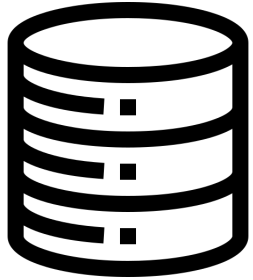
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
 $\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$   
...

# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)

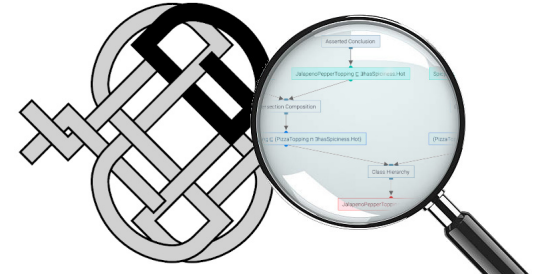


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

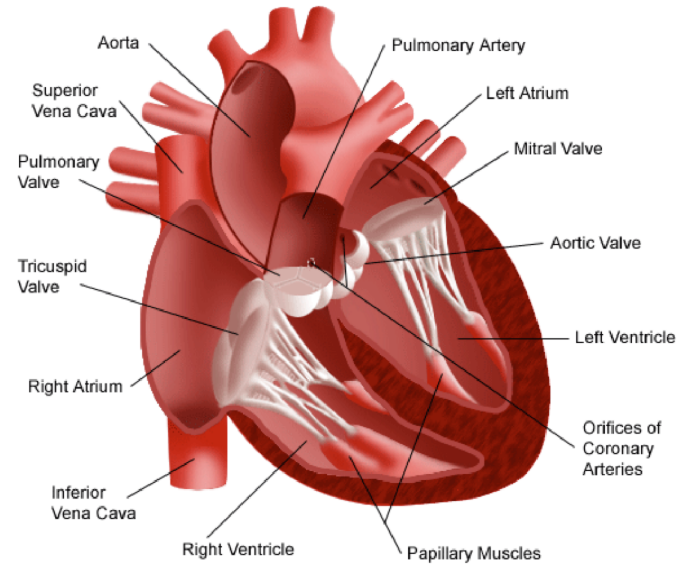
1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.
4. Story of **Success**: SNOMED CT.

## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
 $\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$   
“Voldemort is a descendant of Salazar Slytherin.”  
 $\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$   
...

# Interior View of the Heart



Class hierarchy: 'Aortic valve'

- 'Anatomical structure'
- 'Acellular anatomical structure'
- 'Anatomical cluster'
- 'Anatomical junction'
- 'Biological macromolecule'
- 'Body'
- 'Cardinal body part'
- 'Cardinal cell part'
- 'Cardinal organ part'
- ▼ ● 'Organ component'
- ▼ ● 'Anatomical valve'
- ▼ ● 'Anal valve'
- ▼ ● 'Cardiac valve'
- ▼ ● 'Aortic valve'
- ▶ ● 'Atrioventricular valve'
- ▶ ● 'Pulmonary valve'
- ▶ ● 'Variant cardiac valve'
- ▶ ● 'Vestigial cardiac valve'
- ▶ ● 'Circular fold of small intestine'
- ▶ ● 'Ileocecal valve'
- ▶ ● 'Lacrimal fold'
- ▶ ● 'Lymphatic valve'
- ▶ ● 'Lymphatic valvule'
- ▶ ● 'Spiral valve of cystic duct'
- ▼ ● 'Venous valve'
- ▼ ● 'Valve of axillary vein'
- ▼ ● 'Valve of external jugular vein'
- ▼ ● 'Anular tracheal ligament'

Annotations: 'Aortic valve'

Annotations +

- label [language: en]
- Aortic valve
- hasDefinition
- └ \_genid74267
- hasOBONamespace
- fma

---

Description: 'Aortic valve'

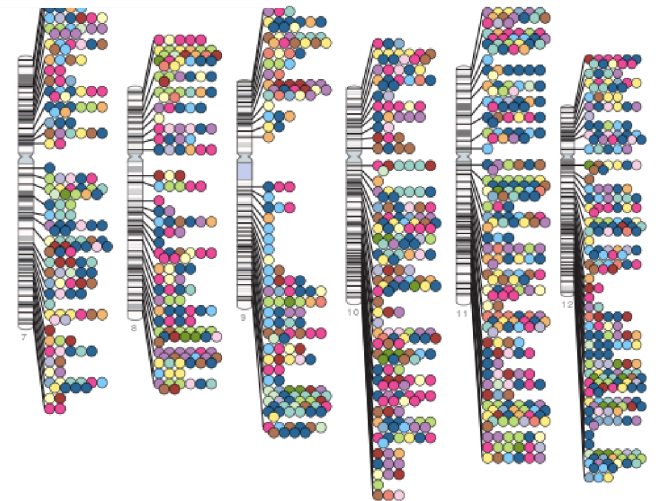
Equivalent To +

SubClass Of +

- 'Cardiac valve'
- attaches\_to some 'Fibrous ring of aortic valve'
- constitutional\_part\_of some 'Left side of heart'
- constitutional\_part\_of some 'Left ventricle'
- constitutional\_part\_of some 'Outflow part of left ventricle'
- constitutional\_part\_of some Heart

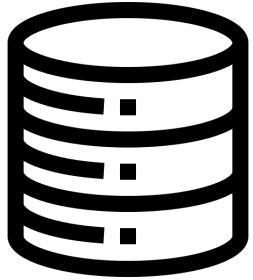
SubClass Of (Anonymous Ancestor)

- attaches\_to some 'Fibrous ring of heart'



# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)

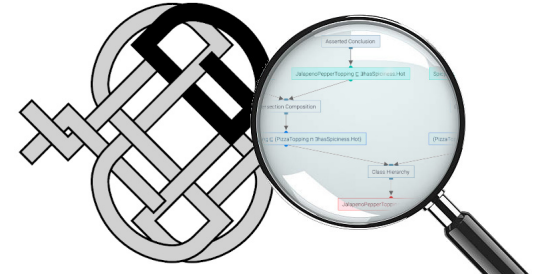


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.
4. Story of **Success**: SNOMED CT.

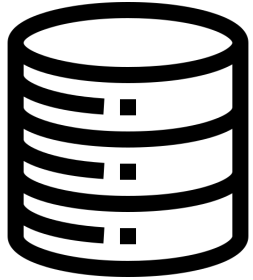
## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”  
$$\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$$
  
“Voldemort is a descendant of Salazar Slytherin.”  
$$\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+. \{\text{SalazarSlytherin}\}$$
  
...

# Our Setting: Incomplete Databases, Description Logics, and Queries

Incomplete Database (ABox)

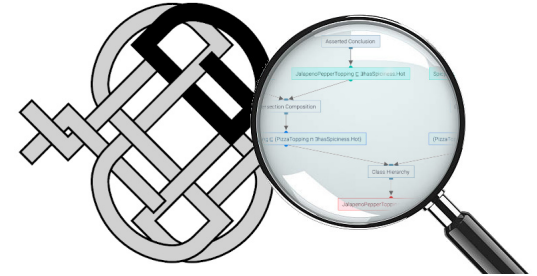


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.
4. Story of **Success**: SNOMED CT.

External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”

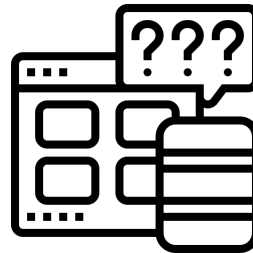
$MuggleBorn \sqsubseteq \forall parent. \neg Wizard$

“Voldemort is a descendant of Salazar Slytherin.”

$\{Voldemort\} \sqsubseteq \exists parent^+. \{SalazarSlytherin\}$

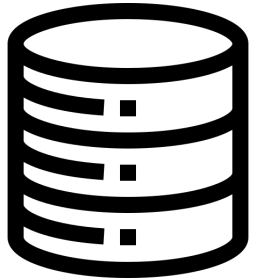
...

User's query



# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)

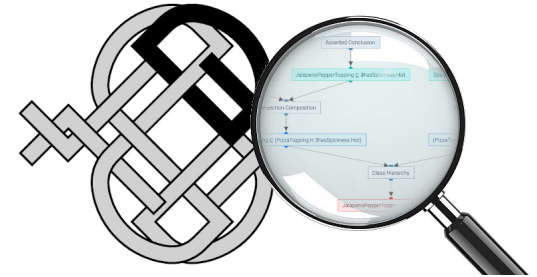


Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.
4. Story of **Success**: SNOMED CT.

## External Knowledge (TBox)



“A Muggle-born person has only non-wizard parents.”

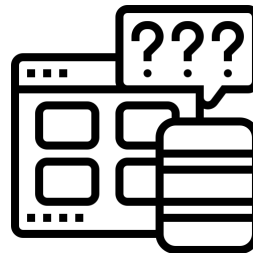
$\text{MuggleBorn} \sqsubseteq \forall \text{parent}. \neg \text{Wizard}$

“Voldemort is a descendant of Salazar Slytherin.”

$\{\text{Voldemort}\} \sqsubseteq \exists \text{parent}^+ . \{\text{SalazarSlytherin}\}$

...

## User's query



## Graph query

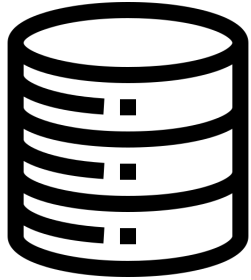
Give me all Muggle-born relatives of Harry.

```
MATCH (:Person {name: 'Harry Potter'})-  
[:parent*]->(:Person)<-[:parent*]-(:Person)  
WHERE p.blood_status = 'MuggleBorn'  
RETURN p.name
```



# Our Setting: Incomplete Databases, Description Logics, and Queries

## Incomplete Database (ABox)



Wizard(HarryPotter)  
MuggleBorn(HermioneGranger)  
friends(HarryPotter, RonWeasley)  
parent(LilyPotter, HarryPotter)  
...

### RelationalDB query

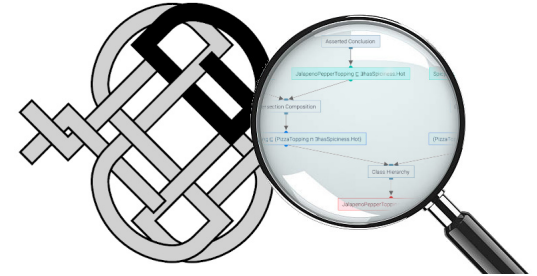
Give me names and houses of Harry's friends.

```
SELECT name, house
From Students JOIN Friends
ON Students.name = Friends.name
WHERE friend_name = 'Harry Potter'
```

## Why?

1. Real data can be **too big**.
2. **Standardization** of terminology.
3. **Automated** reasoning.
4. Story of **Success**: SNOMED CT.

## External Knowledge (TBox)



"A Muggle-born person has only non-wizard parents."

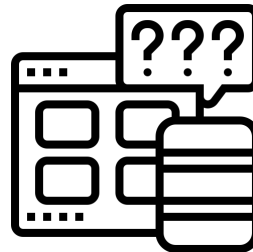
$MuggleBorn \sqsubseteq \forall parent. \neg Wizard$

"Voldemort is a descendant of Salazar Slytherin."

$\{Voldemort\} \sqsubseteq \exists parent^+. \{SalazarSlytherin\}$

...

## User's query



## Graph query

Give me all Muggle-born relatives of Harry.

```
MATCH (:Person {name: 'Harry Potter'})-
[:parent*]->(:Person)<-[:parent*]-(:Person)
WHERE p.blood_status = 'MuggleBorn'
RETURN p.name
```

# Database-Inspired Reasoning Problems

# Database-Inspired Reasoning Problems

1. Satisfiability (consistency) problem.

## Database-Inspired Reasoning Problems

1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

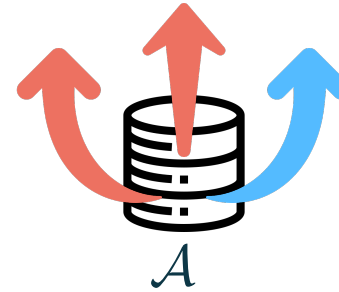


## Database-Inspired Reasoning Problems

1. Satisfiability (consistency) problem.

IN: ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

OUT: Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

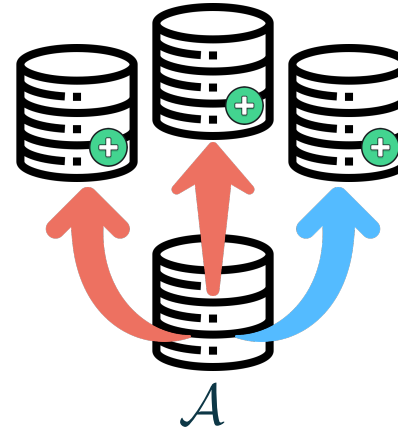


## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

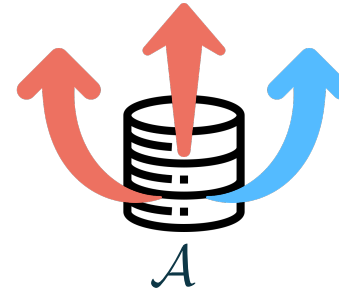


## Database-Inspired Reasoning Problems

1. Satisfiability (consistency) problem.

IN: ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

OUT: Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

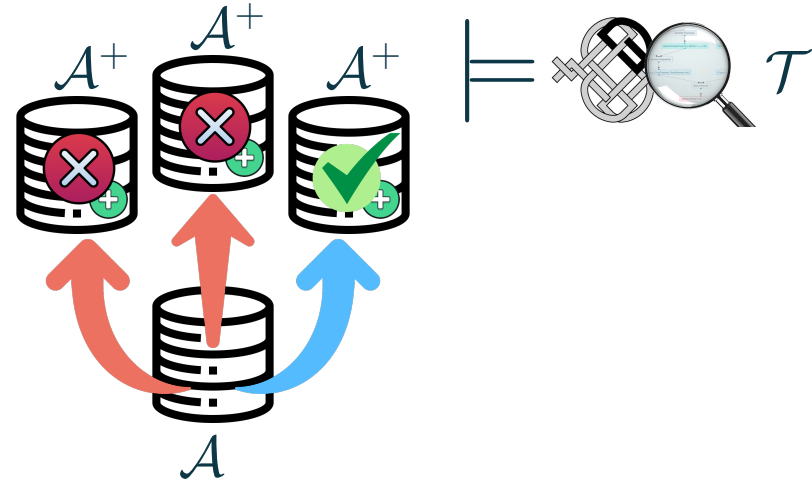


## Database-Inspired Reasoning Problems

1. **Satisfiability** (consistency) **problem**.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an **extension** of  $\mathcal{A}$  **satisfying**  $\mathcal{T}$ ?





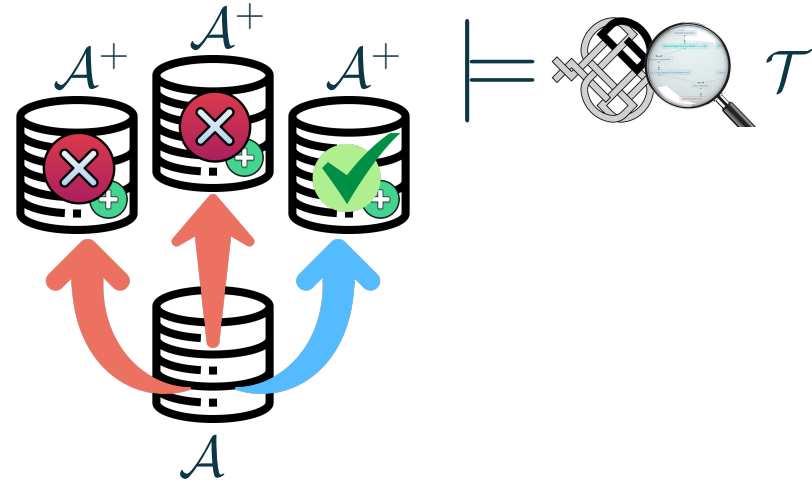
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!



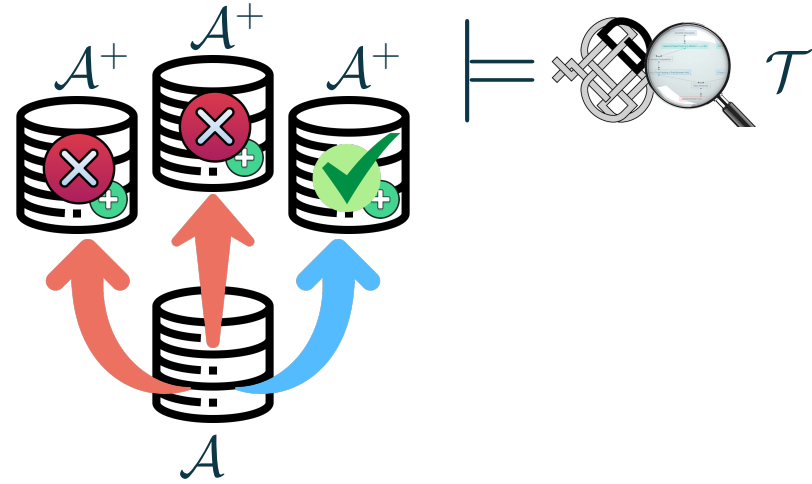
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.



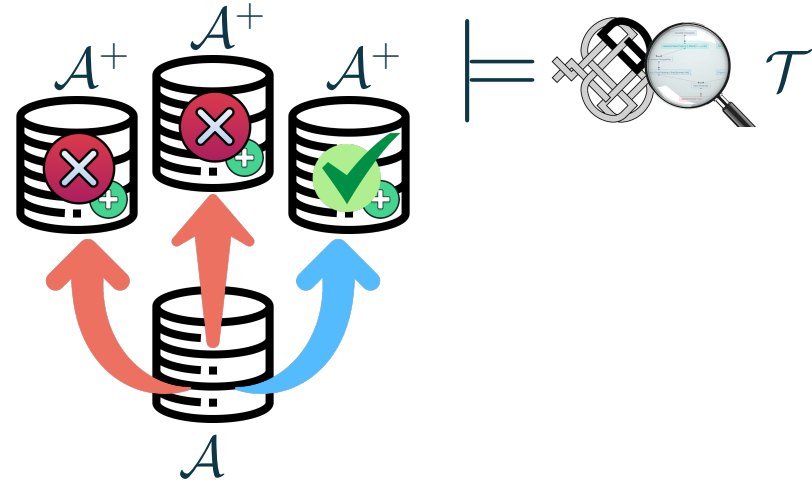
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



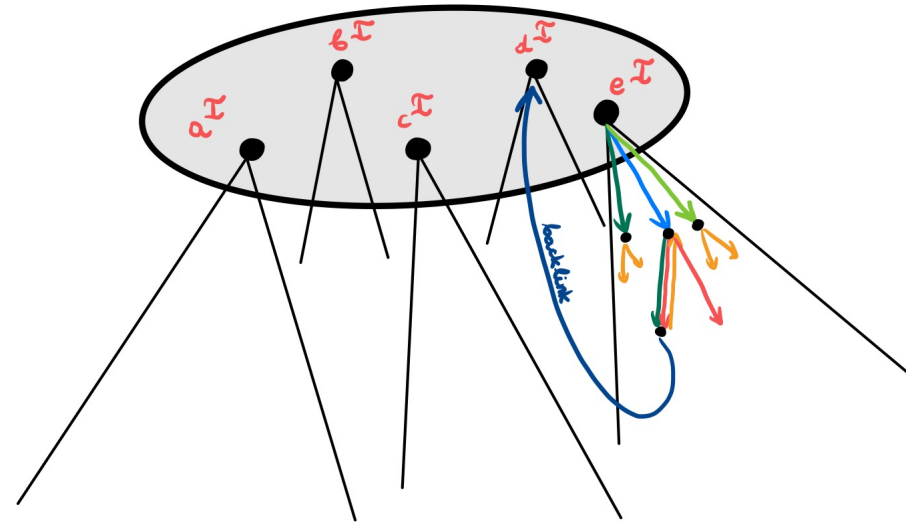
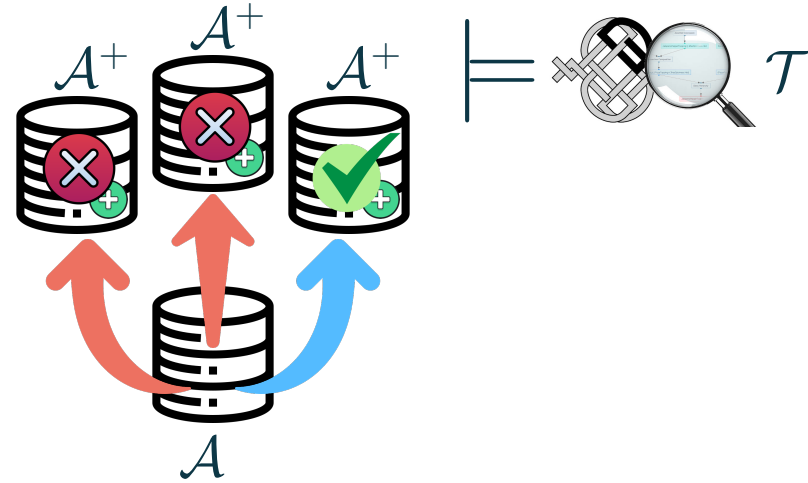
# Database-Inspired Reasoning Problems

## 1. Satisfiability (consistency) problem.

IN: ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

OUT: Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



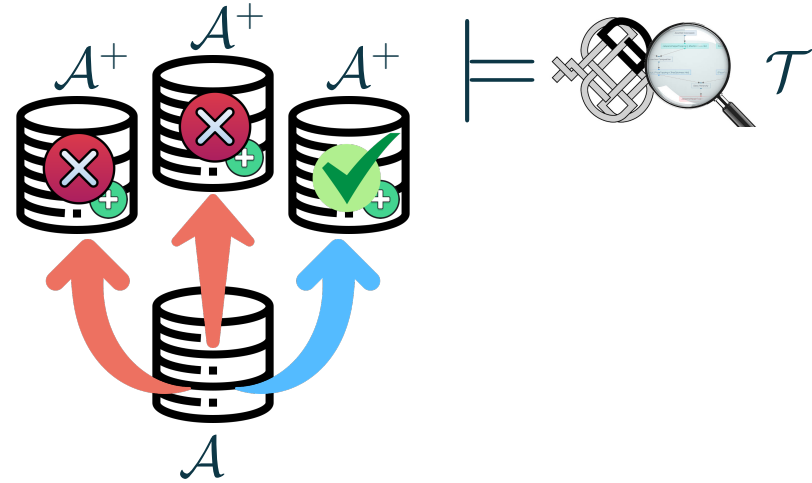
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



## Database-Inspired Reasoning Problems

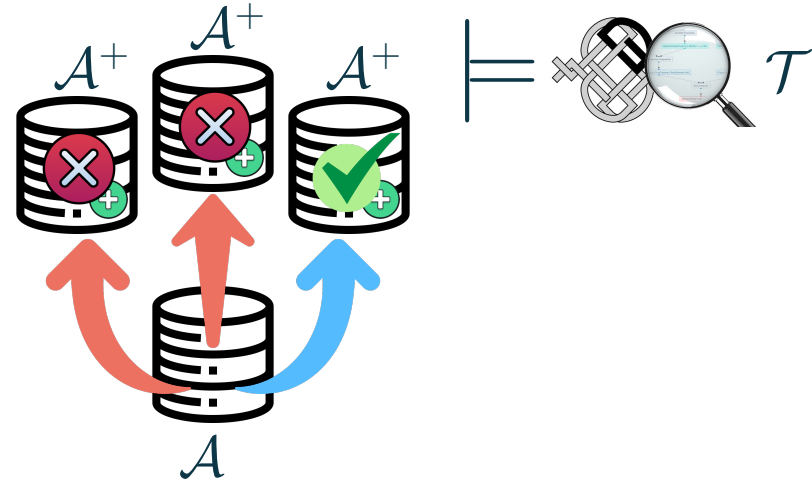
### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.



## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

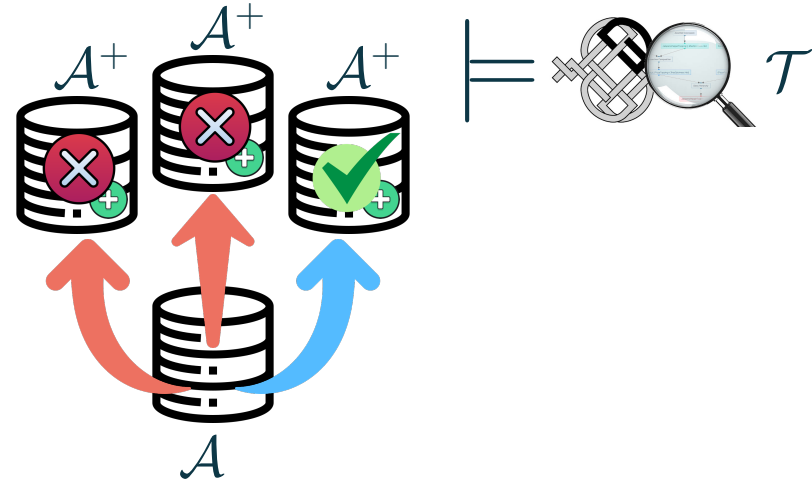
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Do all extensions of  $\mathcal{A}$  satisfying  $\mathcal{T}$  match  $q$ ?



## Database-Inspired Reasoning Problems

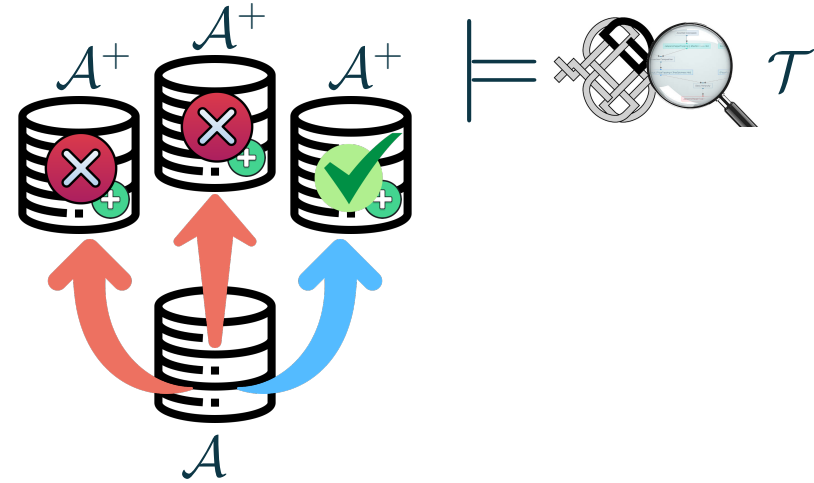
### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.





## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A_{Box}$   $\mathcal{A}$  (DB) +  $T_{Box}$   $\mathcal{T}$  (Knowledge)

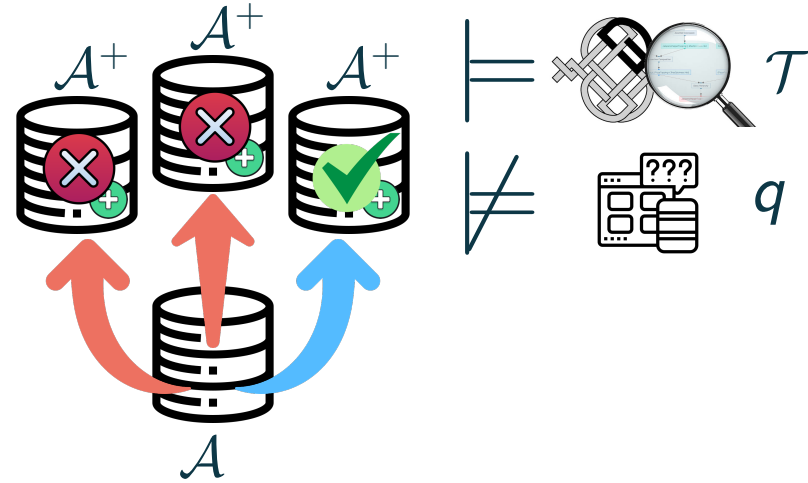
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $A_{Box}$   $\mathcal{A}$  (DB) +  $T_{Box}$   $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?



## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge)

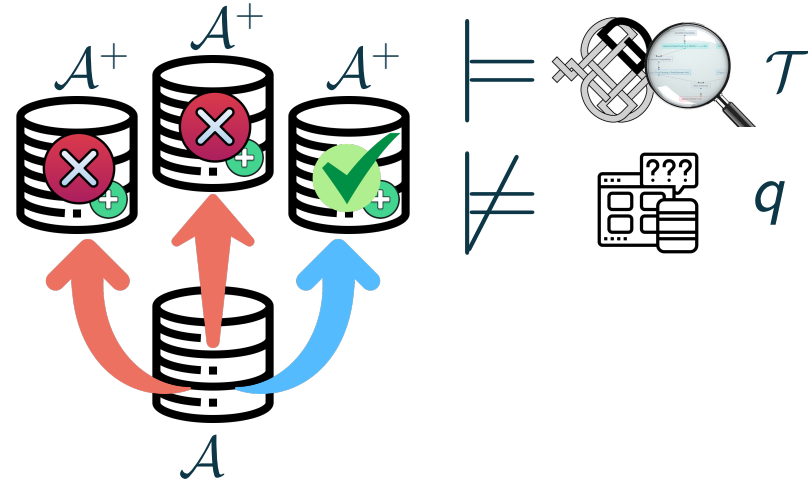
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?



- (Unions of) Conjunctive Queries

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

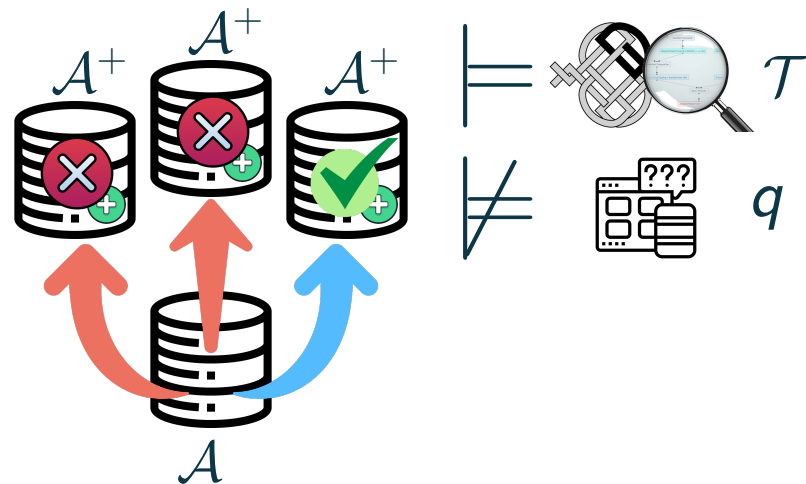
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?



### • (Unions of) Conjunctive Queries

Tell me the names and houses of Harry's friends.

```
SELECT name, house
FROM Students JOIN Friends
ON Students.name = Friends.name
WHERE friend_name = 'Harry Potter'
```

$\text{friends}(n, \text{Harry Potter}) \wedge \text{belongsTo}(n, h) \wedge \text{House}(h)$

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A_{Box}$   $\mathcal{A}$  (DB) +  $T_{Box}$   $\mathcal{T}$  (Knowledge)

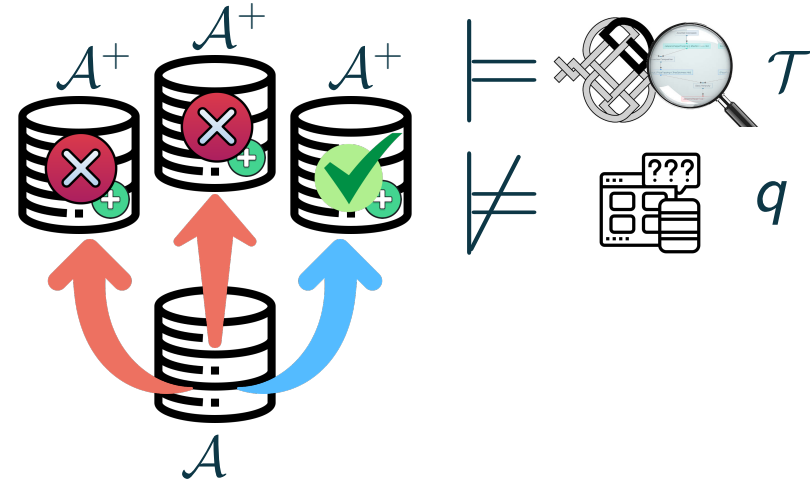
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $A_{Box}$   $\mathcal{A}$  (DB) +  $T_{Box}$   $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?



- (Unions of) Conjunctive Queries

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

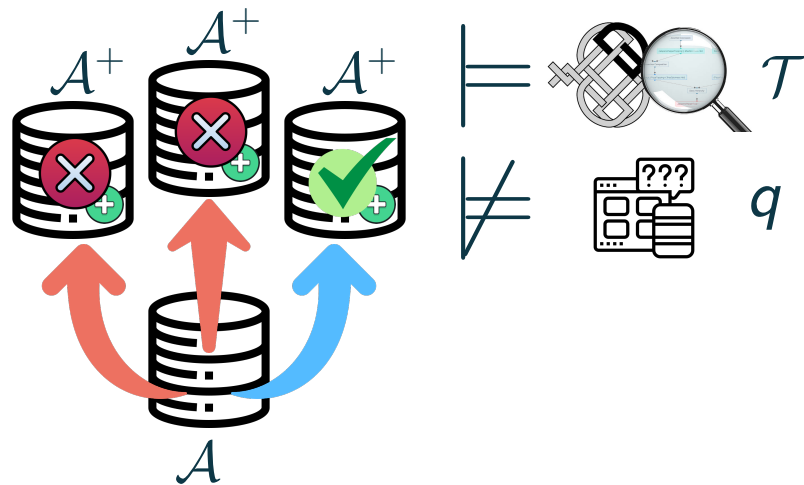
- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries



## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

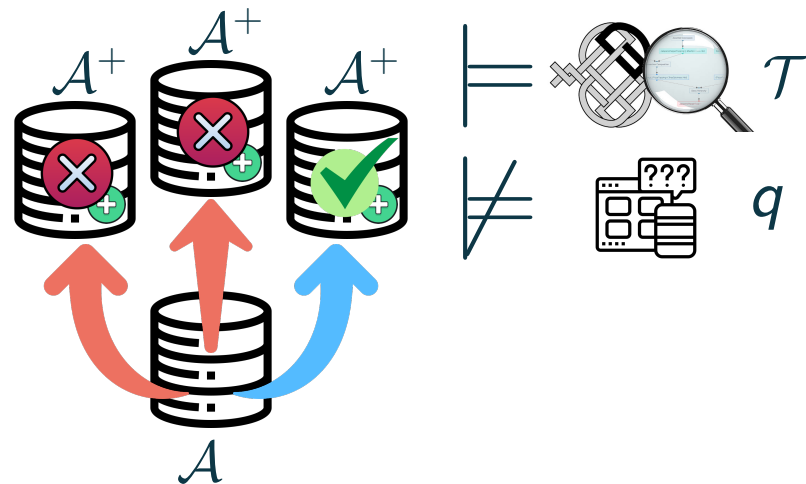
**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?



- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Give me all Muggle-born relatives of Harry.

```
MATCH (:Person {name: 'Harry Potter'})-  
[:parent*]->(:Person)<-[:parent*]-(:Person)  
WHERE p.blood_status = 'MuggleBorn'  
RETURN p.name
```

$\exists a \text{ parent}^*(\text{HarryPotter}, a) \wedge (\text{parent}^-)^*(a, n) \wedge \text{MuggleBorn}(n)$

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A_{\text{Box}} \mathcal{A}$  (DB) +  $T_{\text{Box}} \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

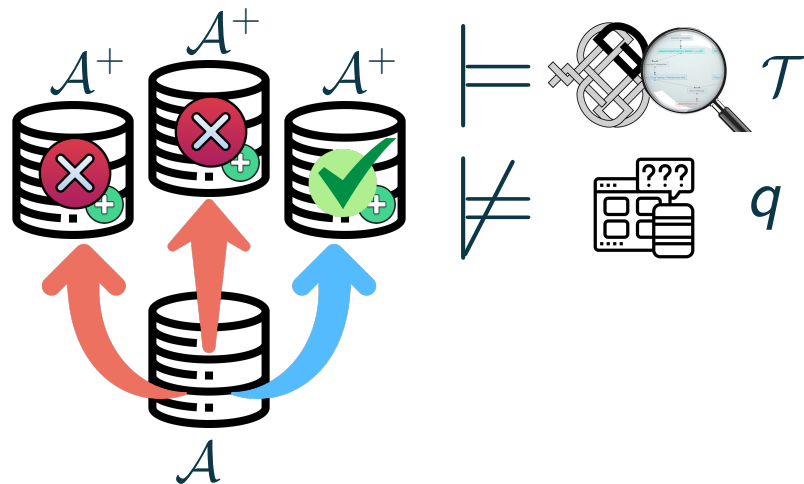
- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:**  $A_{\text{Box}} \mathcal{A}$  (DB) +  $T_{\text{Box}} \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries



## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

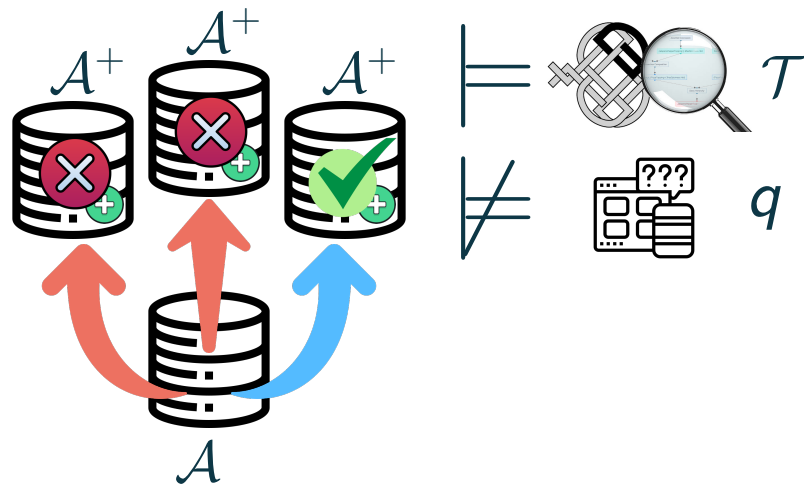
- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

### 2. Certain query answering.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

Complexity measures: combined (everything as input) vs data (all but data is fixed)



- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries



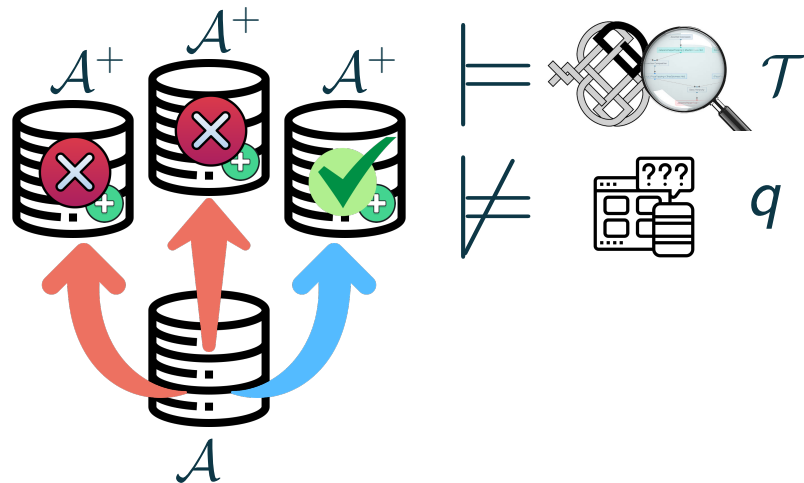
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

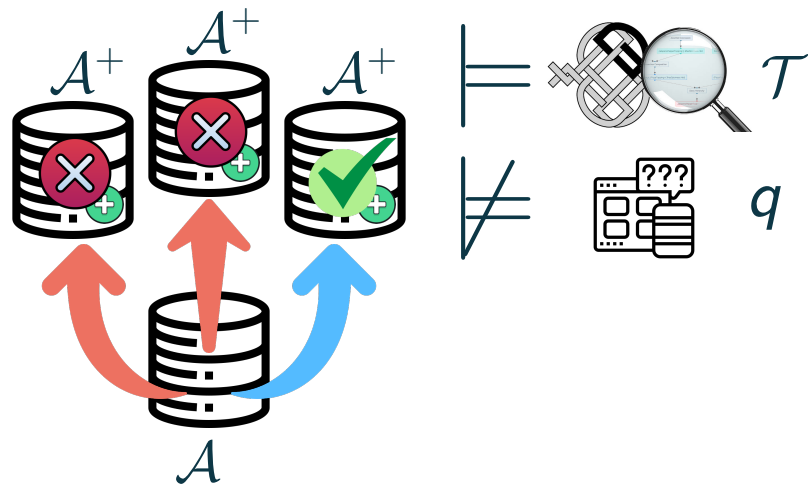
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

# Finite vs Unrestricted Semantics

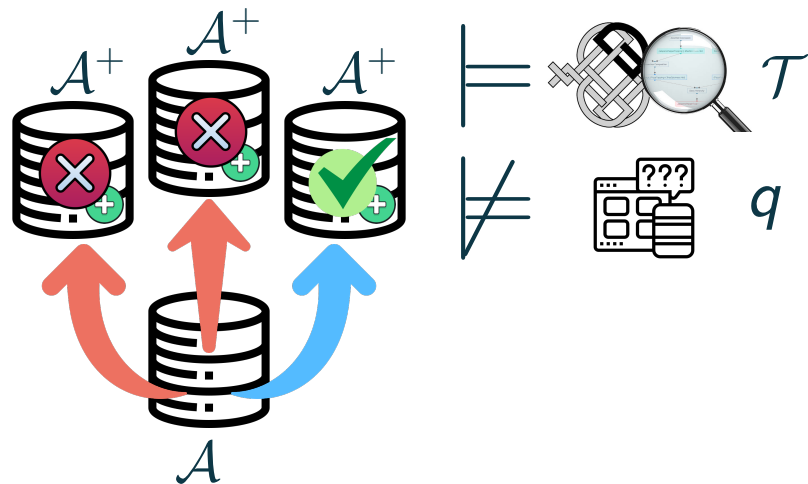
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics

Makes a difference and a challenge!

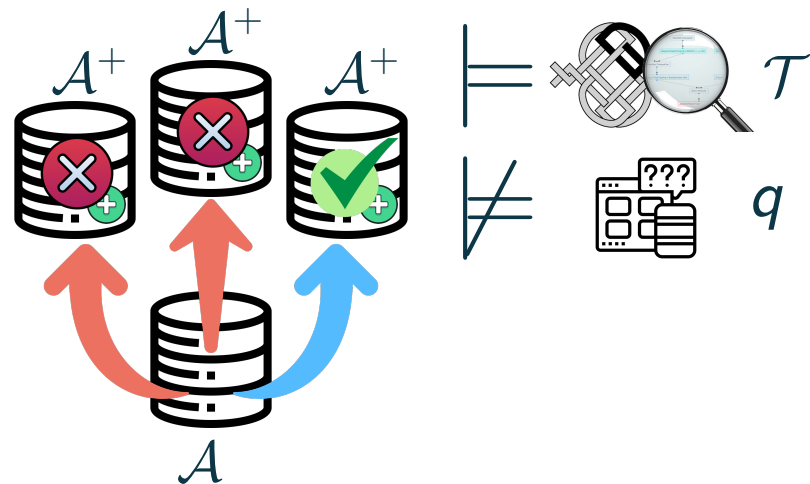
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

**IN:**  $A\Box \mathcal{A}$  (DB) +  $T\Box \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics

Makes a difference and a challenge!

$\mathcal{T} :=$  "everyone has a parent".

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

IN: ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

OUT: Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

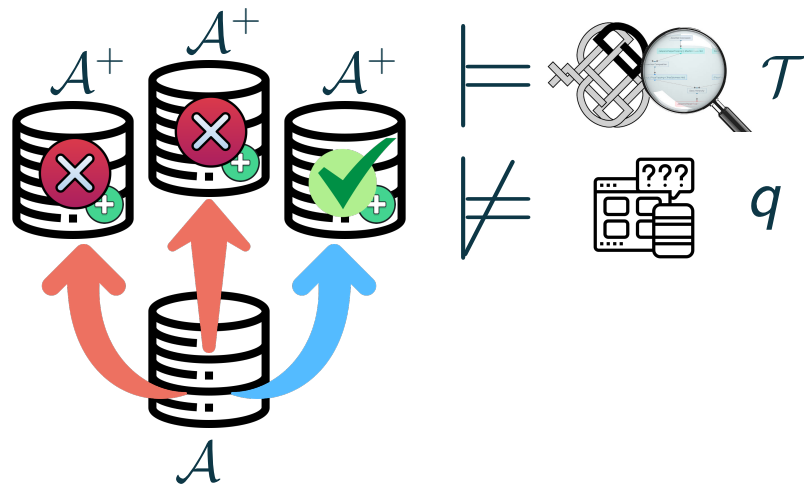
### 2. Certain query answering.

IN: ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

OUT: Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)



## Finite vs Unrestricted Semantics

Makes a difference and a challenge!

$\mathcal{T} :=$  "everyone has a parent".

$q := \exists x \text{parent}^+(x, x)$ .

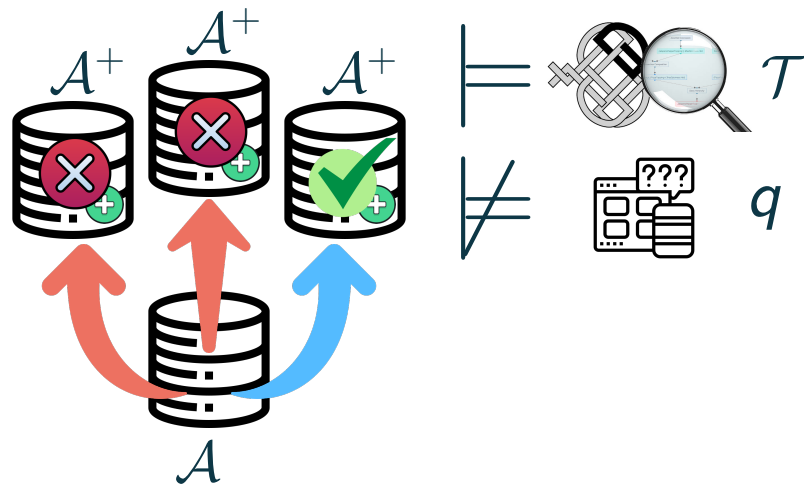
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics

Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

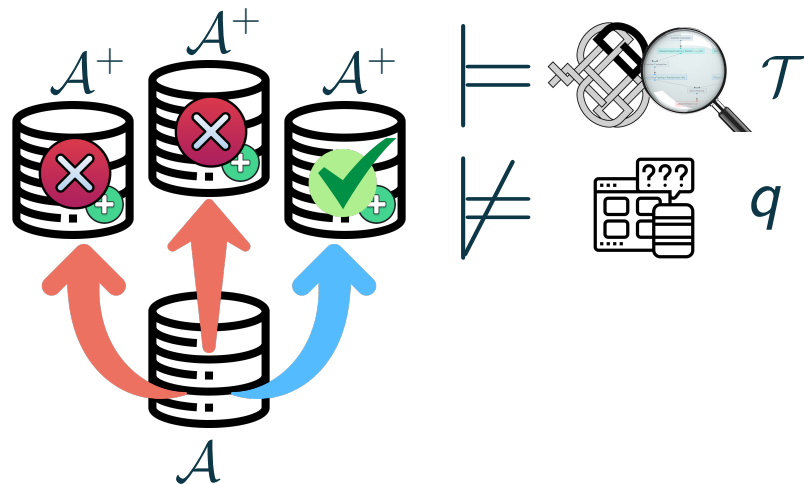
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

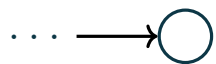
**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{ parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.

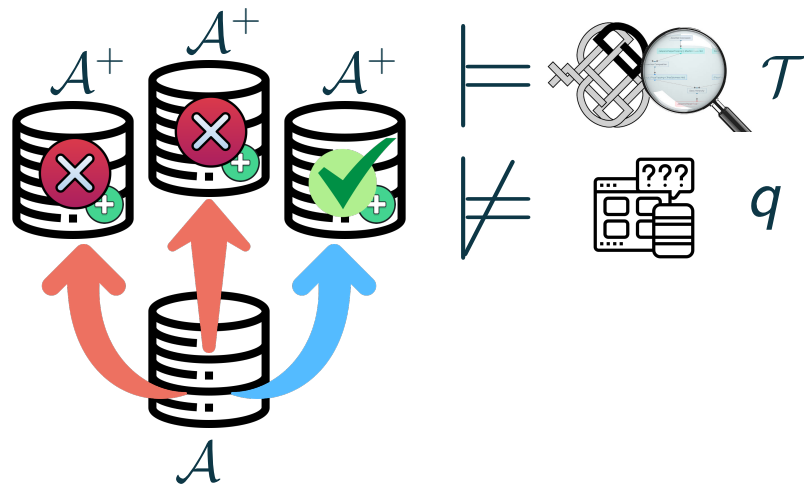
### 2. Certain query answering.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)



## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{ parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .



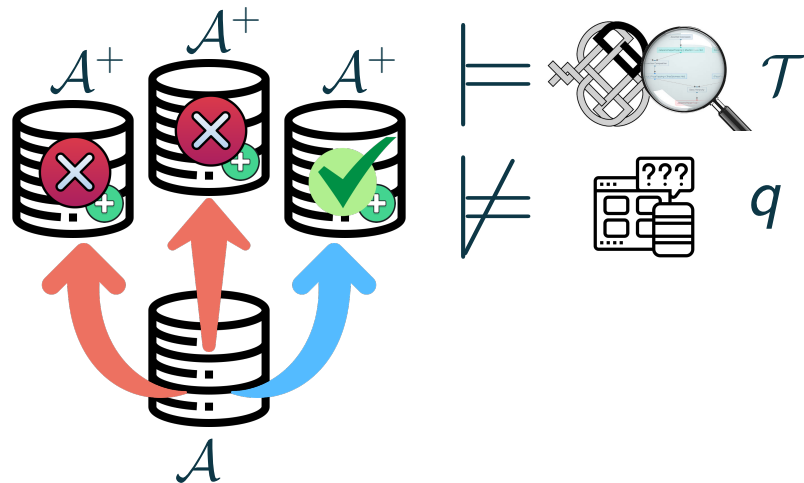
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

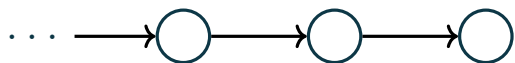
**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{ parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

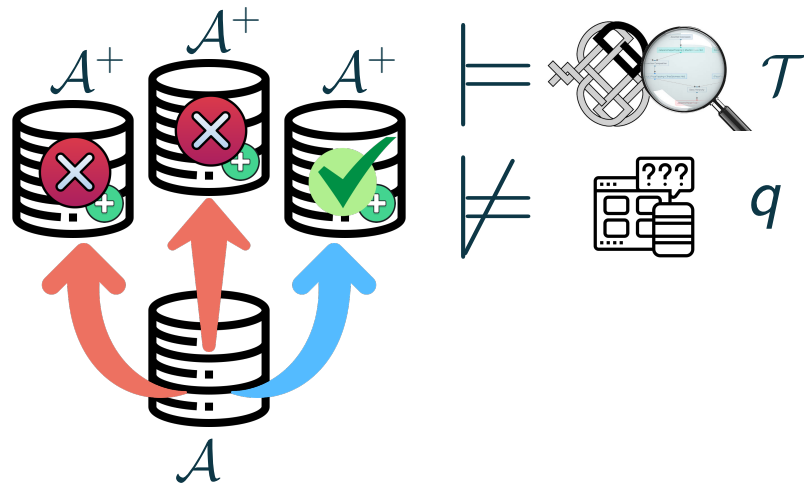
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

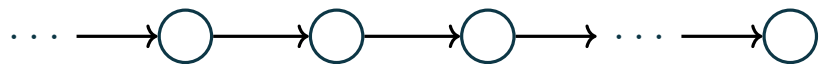
**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{ parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

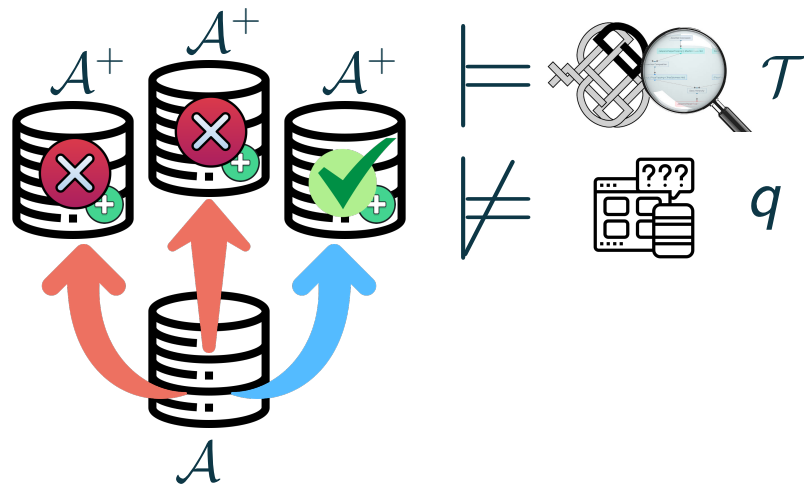
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

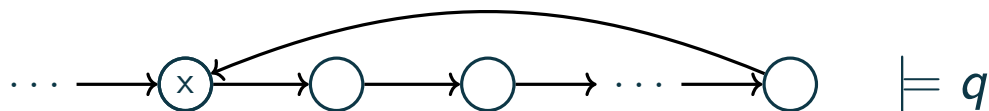
**IN:** ABox  $\mathcal{A}$  (DB) + TBox  $\mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{ parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

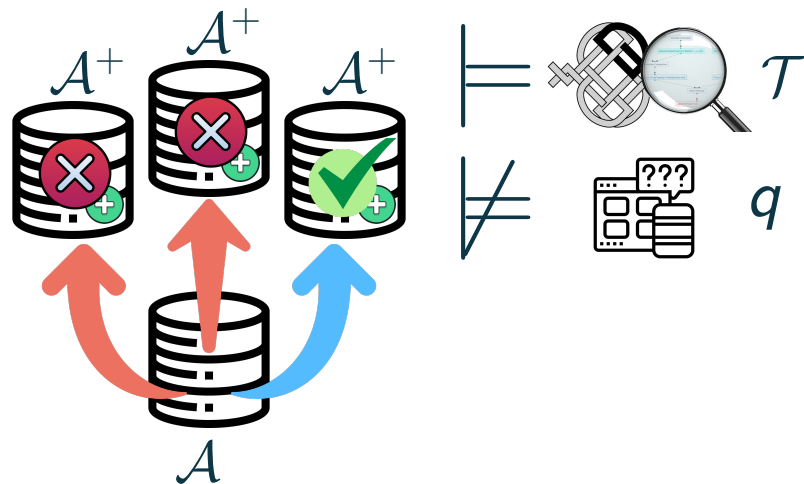
## Database-Inspired Reasoning Problems

### 1. Satisfiability (consistency) problem.

**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge)

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$ ?

- No a priori bounds on sizes of such extensions!
- Easily undecidable or at least computationally hard.
- Use of model and automata theory: nicely-shaped models.



### 2. Certain query answering.

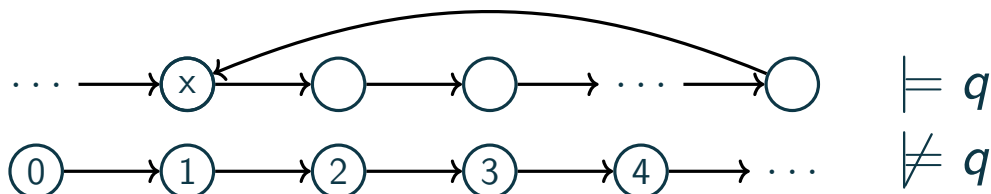
**IN:**  $ABox \mathcal{A}$  (DB) +  $TBox \mathcal{T}$  (Knowledge) + query  $q$

**OUT:** Is there an extension of  $\mathcal{A}$  satisfying  $\mathcal{T}$  but not  $q$ ?

- (Unions of) Conjunctive Queries
- (Unions of) Conj. 2-way Regular Path Queries

Complexity measures: combined (everything as input) vs data (all but data is fixed)

## Finite vs Unrestricted Semantics



Makes a difference and a challenge!

$\mathcal{T} :=$  “everyone has a parent”.

$q := \exists x \text{parent}^+(x, x)$ .

We have  $\mathcal{T} \models_{\text{fin}} q$  but  $\mathcal{T} \not\models q$ .

# Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

- ( $\mathcal{Hb}$ ) role combinations



## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists$ loves.Self

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg$ Muggle,  $\top \sqsubseteq \exists$ hasParent.Male  $\sqcap \exists$ hasParent.Female

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists$ loves.Self

- (reg) regular path expressions

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

$\mathcal{ALC}\mathcal{Hb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}$ ,  $\top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

$\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}$ ,  $\top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>,  $\{\text{Voldemort}\} \sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$



## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>, {Voldemort}  $\sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$

### Motivation?

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>, {Voldemort}  $\sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$

### Motivation?

- On the verge of decidability

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>, {Voldemort}  $\sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALCHb}^{\text{Self}}_{\text{reg}}$   
a.k.a.  $\mathcal{Z}$

### Motivation?

- On the verge of decidability
- Encodes OWL 2, W3C Web Ontology Language

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}$ ,  $\top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>,  $\{\text{Voldemort}\} \sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALC}\mathcal{Hb}\text{Self}_{\text{reg}}$   
a.k.a.  $\mathcal{Z}$

### Motivation?

- On the verge of decidability
- Generalize positive results for  $\mathcal{ALC}$
- Encodes OWL 2, W3C Web Ontology Language

## Description Logics with Path Expressions, a.k.a. the $\mathcal{Z}$ family [2009]

- Basic description logic  $\mathcal{ALC}$

Wizard  $\sqsubseteq \neg\text{Muggle}, \top \sqsubseteq \exists\text{hasParent.Male} \sqcap \exists\text{hasParent.Female}$

- ( $\mathcal{Hb}$ ) role combinations

hasParent = hasMother  $\cup$  hasFather

- (Self) operator

Narcissist  $\sqsubseteq \exists\text{loves.Self}$

- (reg) regular path expressions

PureBloodWizard  $\sqsubseteq \forall\text{isParent}^+.\text{PureBloodWizard}$

- (I) inverses, (O) nominals, and (Q) counting

isAncestor = isDescendant<sup>-</sup>, {Voldemort}  $\sqsubseteq (=7 \text{ created}).\text{Horcrux}$

$\mathcal{ALC}\mathcal{Hb}_{\text{reg}}^{\text{Self}}$   
a.k.a.  $\mathcal{Z}$

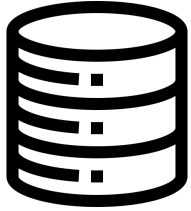
### Motivation?

- On the verge of decidability
- Encodes OWL 2, W3C Web Ontology Language
- Generalize positive results for  $\mathcal{ALC}$
- $\mathcal{ALC}_{\text{reg}}$  is a notational variant of Propositional Dynamic Logic

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

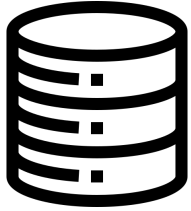
# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Incomplete Database

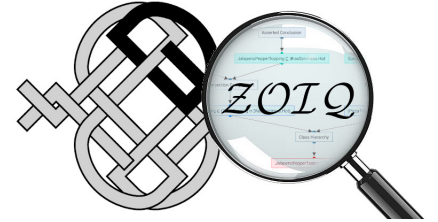


# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Incomplete Database



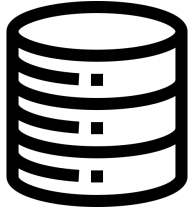
Knowledge ( $\mathcal{Z}$  family of DLs)





# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

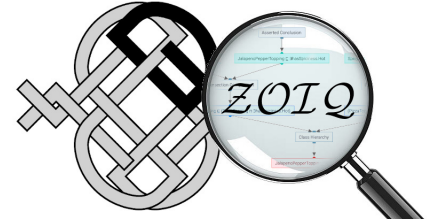
Incomplete Database



Query (Relational vs Graph)

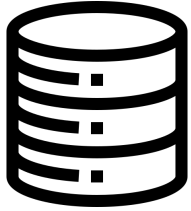


Knowledge ( $\mathcal{Z}$  family of DLs)



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

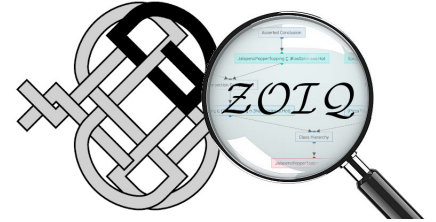
Incomplete Database



Query (Relational vs Graph)



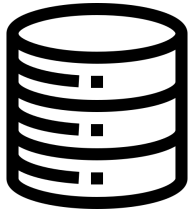
Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{Z}OIQ$ .

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Incomplete Database



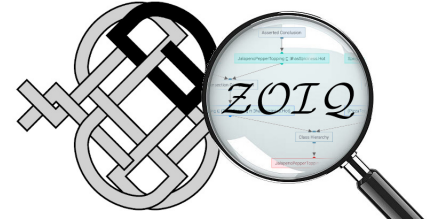
- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .

Query (Relational vs Graph)



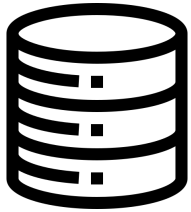
- Uniform approach: cover many existing DLs in one-go.

Knowledge ( $\mathcal{Z}$  family of DLs)



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

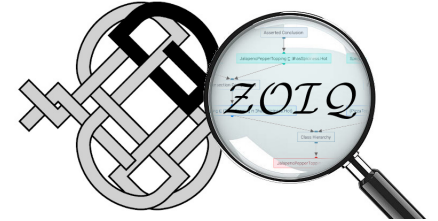
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .

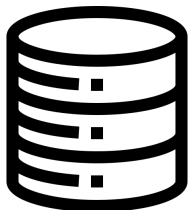
- Uniform approach: cover many existing DLs in one-go.

---

Our main results?

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

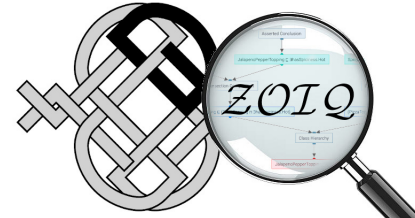
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .

- Uniform approach: cover many existing DLs in one-go.

---

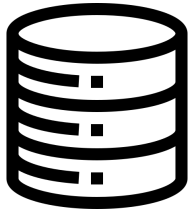
## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ )?



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

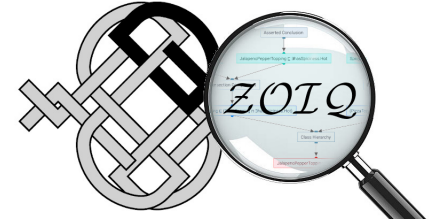
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{Z}OIQ$ .

- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?

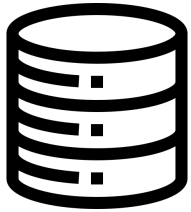


$\mathcal{Z}Q$  without  $Self \in EXP$

$\mathcal{ALC}Self$  is  $2EXP$ -hard

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

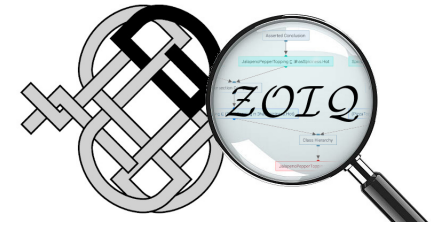
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{Z}OIQ$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?

SAT of (tamed)  $\mathcal{Z}OIQ$ ?

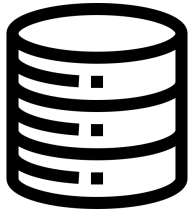


$\mathcal{Z}Q$  without  $Self \in EXP$

$\mathcal{ALC}Self$  is  $2EXP$ -hard

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

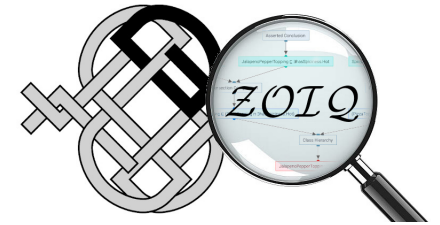
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{Z}OIQ$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?

SAT of (tamed)  $\mathcal{Z}OIQ$ ?



$\mathcal{Z}Q$  without Self  $\in$  EXP

SAT in NP (data-comp)

$\mathcal{ALC}Self$  is 2EXP-hard



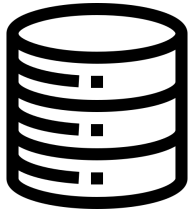
sole author





# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

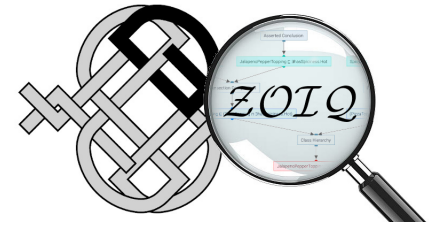
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{Z}OIQ$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?



$\mathcal{Z}Q$  without Self  $\in$  EXP

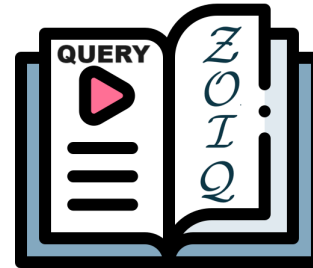
$\mathcal{ALC}Self$  is 2EXP-hard

SAT of (tamed)  $\mathcal{Z}OIQ$ ?



SAT in NP (data-comp)

Querying (tamed)  $\mathcal{Z}OIQ$ ?

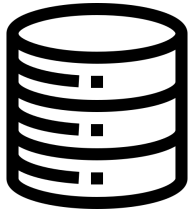


sole author



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

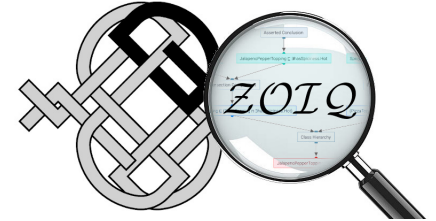
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?



$\mathcal{ZQ}$  without Self  $\in$  EXP

$\mathcal{ALC}Self$  is 2EXP-hard

SAT of (tamed)  $\mathcal{ZOIQ}$ ?

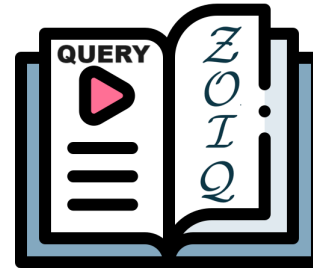


SAT in NP (data-comp)



sole author

Querying (tamed)  $\mathcal{ZOIQ}$ ?



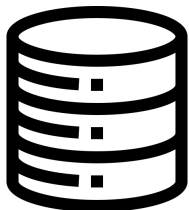
New 2EXP upper bounds

$\mathcal{ZOI}$  and  $\mathcal{ZOQ}$  are FC



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

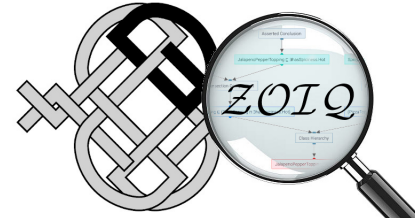
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?

SAT of (tamed)  $\mathcal{ZOIQ}$ ?

Querying (tamed)  $\mathcal{ZOIQ}$ ? Beyond Regularity?



$\mathcal{ZQ}$  without Self  $\in$  EXP

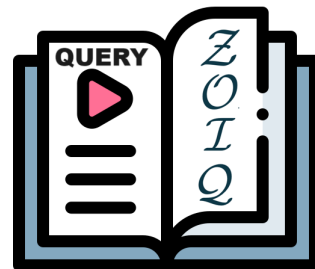
$\mathcal{ALC}^{Self}$  is 2EXP-hard



SAT in NP (data-comp)

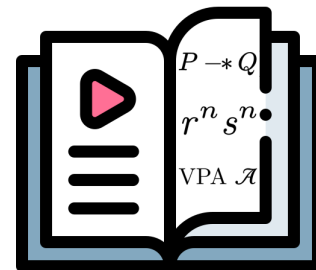


sole author



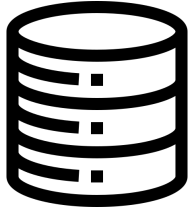
New 2EXP upper bounds

$\mathcal{ZOI}$  and  $\mathcal{ZOQ}$  are FC



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

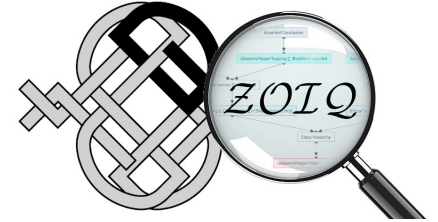
Incomplete Database



Query (Relational vs Graph)



Knowledge ( $\mathcal{Z}$  family of DLs)



- Complexity of (fragments of)  $\mathcal{ZOIQ}$ .
- Uniform approach: cover many existing DLs in one-go.

## Our main results?

Querying  $\mathcal{Z}$  ( $\mathcal{ALCHb}_{reg}^{Self}$ )?



$\mathcal{ZQ}$  without Self  $\in$  EXP  
 $\mathcal{ALC}^{Self}$  is 2EXP-hard

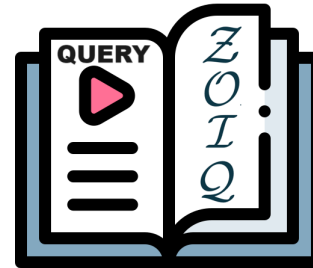
SAT of (tamed)  $\mathcal{ZOIQ}$ ?



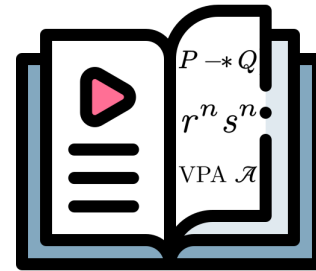
SAT in NP (data-comp)



Querying (tamed)  $\mathcal{ZOIQ}$ ? Beyond Regularity?



New 2EXP upper bounds  
 $\mathcal{ZOI}$  and  $\mathcal{ZOQ}$  are FC



Undecidability  
**JELIA**   
 solo + award

More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



2. Compl. of SAT < Compl. of Querying



## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



2. Compl. of SAT < Compl. of Querying



- ( $\mathcal{H}$ ) role inclusions

hasMother  $\subseteq$  hasParent



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



2. Compl. of SAT < Compl. of Querying



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )

2. Compl. of SAT < Compl. of Querying



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )

2. Compl. of SAT < Compl. of Querying



- ( $\mathcal{Q}$ ) counting

(=7 created).Horcrux

More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )

2. Compl. of SAT < Compl. of Querying



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying



## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )

( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying



- ( $\mathcal{S}$ ) transitivity

isAncestor relation is transitive

More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying



More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )





## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )



- ( $\mathcal{I}$ ) inverses

isParent is the inverse of isChild role

More general question: what features  $\Theta$  make CQ answering hard for  $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )



More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )



More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )



- ( $\mathcal{O}$ ) nominals

{Voldemort}

More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )



More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



How about other features  $\Theta$ ?

More general question: what features  $\Theta$  make CQ answering hard for  $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



How about other features  $\Theta$ ?

- Regular expressions



## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



$(\mathcal{H})$   
 $(\mathcal{Q})$

2. Compl. of SAT < Compl. of Querying

$(\mathcal{S})$   
 $(\mathcal{I})$   
 $(\mathcal{O})$



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints

## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator

## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Presburger counting

- Statistical constraints

- Self Operator

## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations

## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying



( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )

### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points

## More general question: what features $\Theta$ make CQ answering hard for $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



## How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points
- More???

## More general question: what features $\Theta$ make CQ answering hard for $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points
- More???



## More general question: what features $\Theta$ make CQ answering hard for $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points
- More???



Is there any pattern behind this?



# More general

1. Compl. of SAT = C

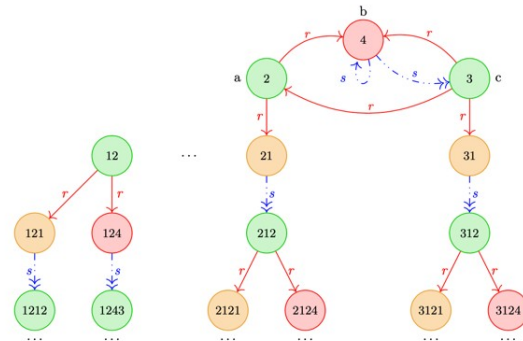
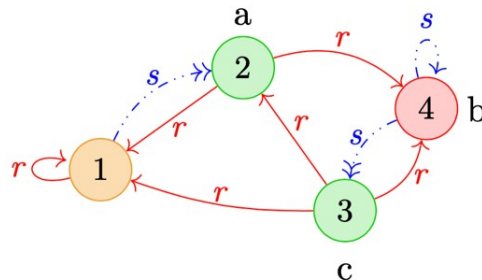


- Regular expressions
- Presburger counting



d for  $ALCO?$

$\Gamma < \text{Compl. of Querying}$



More???

## More general question: what features $\Theta$ make CQ answering hard for $ALC\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points
- More???



Is there any pattern behind this?

## More general question: what features $\Theta$ make CQ answering hard for $\mathcal{ALCC}\Theta$ ?

1. Compl. of SAT = Compl. of Querying



( $\mathcal{H}$ )  
( $\mathcal{Q}$ )

2. Compl. of SAT < Compl. of Querying

( $\mathcal{S}$ )  
( $\mathcal{I}$ )  
( $\mathcal{O}$ )



### How about other features $\Theta$ ?

- Regular expressions
- Statistical constraints
- Self Operator
- Presburger counting
- Boolean role combinations
- Fixed-points
- More???



### Is there any pattern behind this?

#### Theorem (Part I of the Dissertation, Both Semantics)

If  $\mathcal{ALCC}\Theta$  has locally-forward models then SAT = QueryEntl.

The Self operator is bad. Querying  $\mathcal{ZQ}$  without Self is EXPTIME-compl.

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

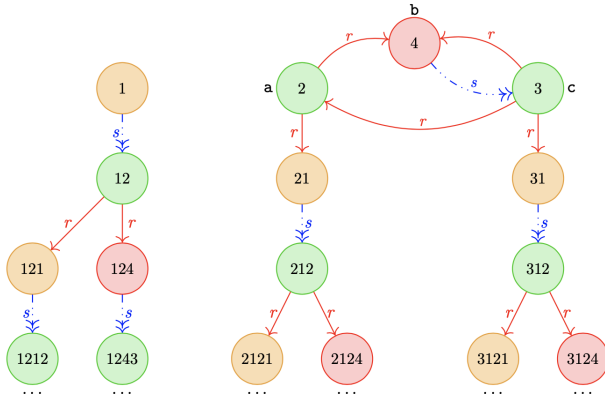
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then,

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

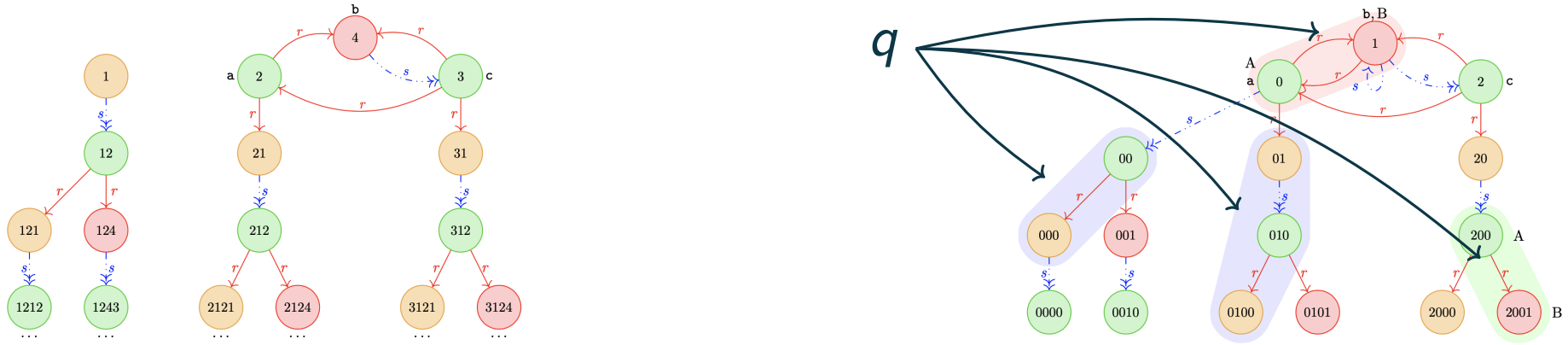
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .





# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

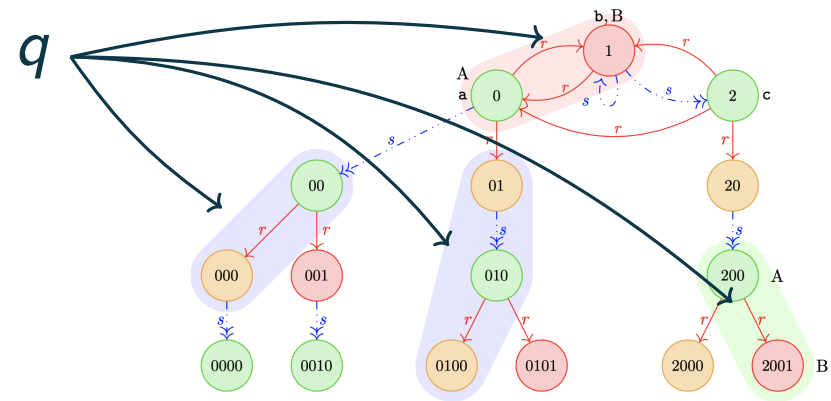
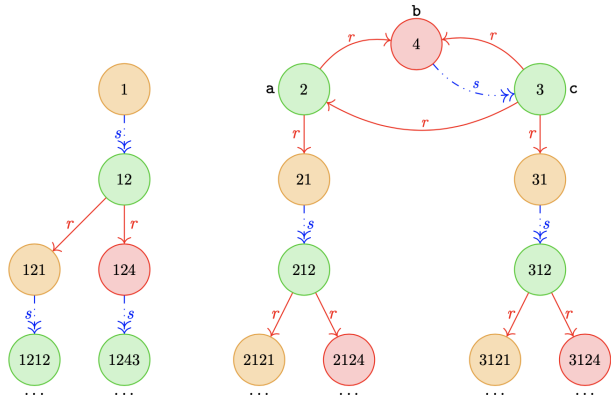
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into:

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

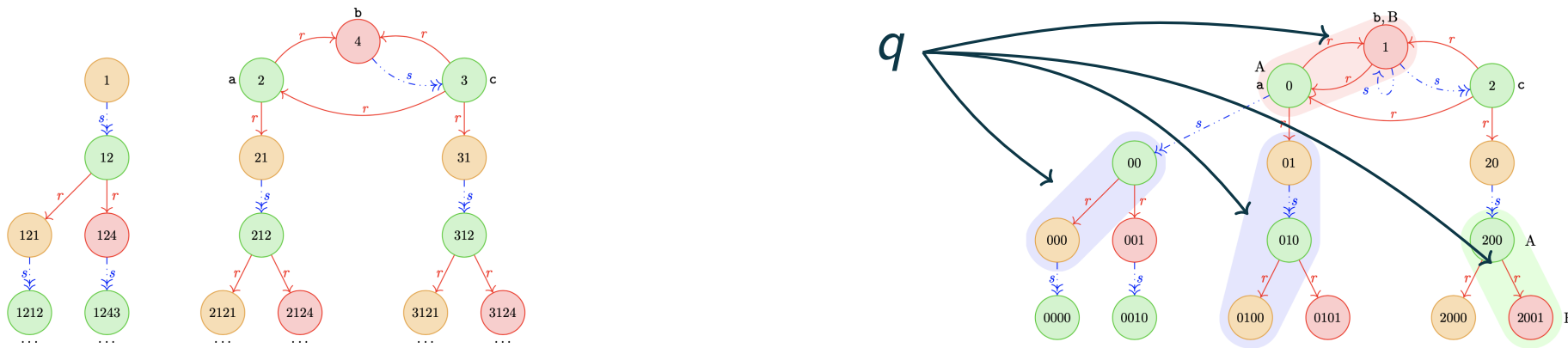
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots,

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

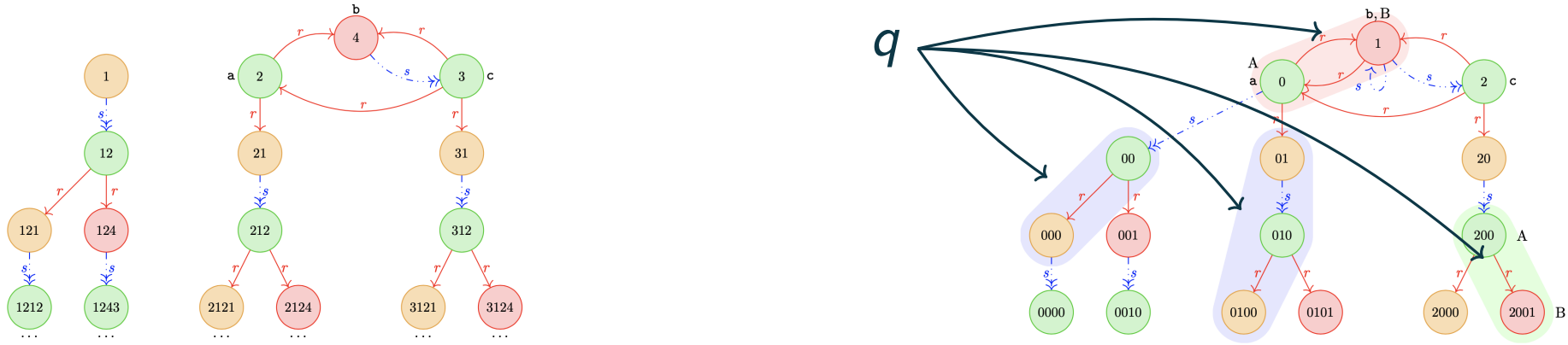
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots,

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

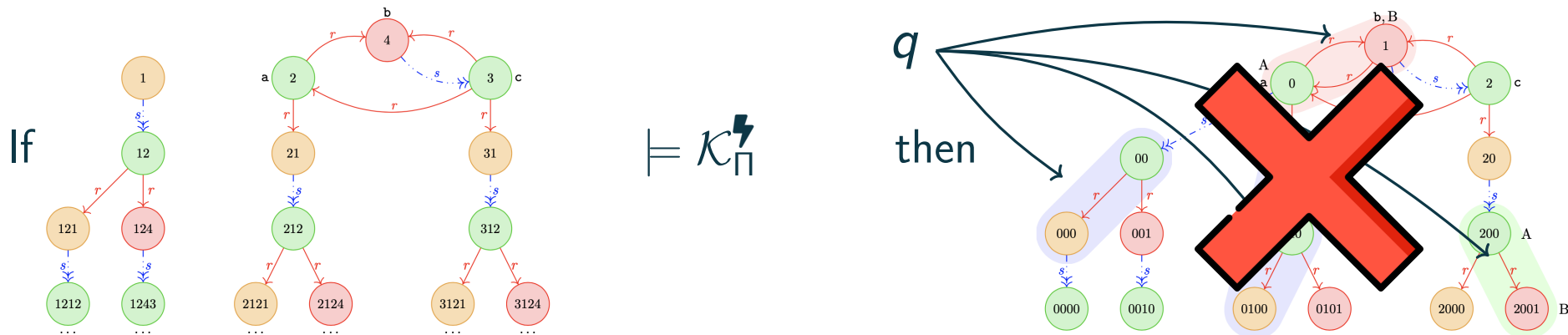
- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

## CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .

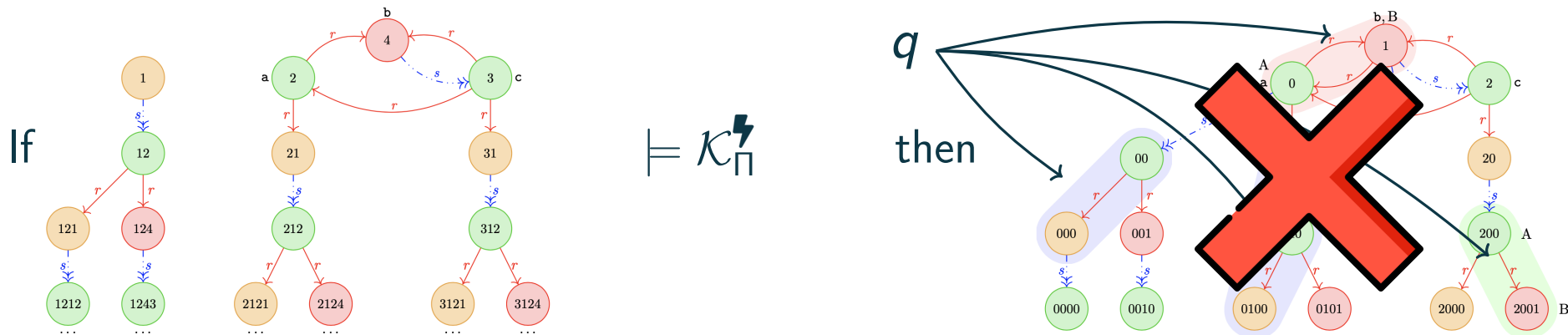


A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{⚡}}$  “blocking” all matches yielding such a partition

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



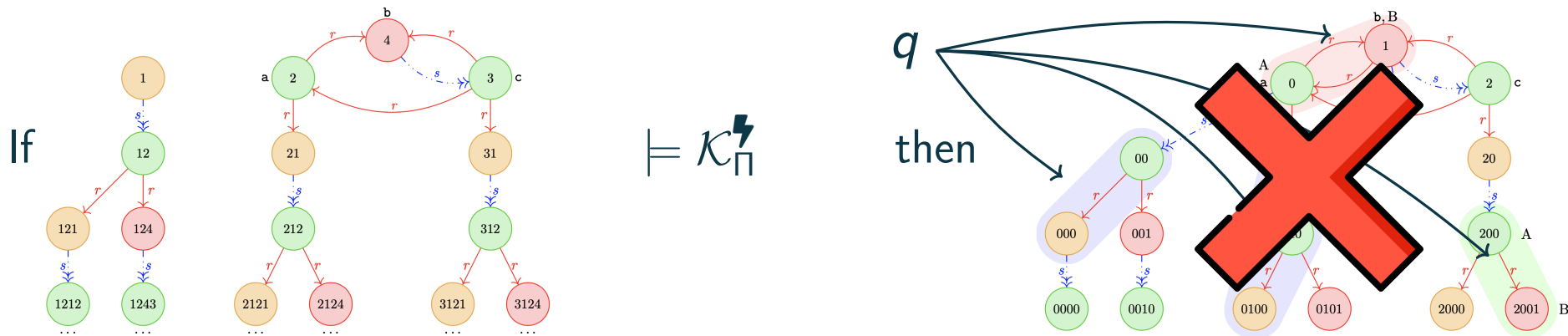
A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{lightning bolt}}$  “blocking” all matches yielding such a partition

## Key Lemma

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

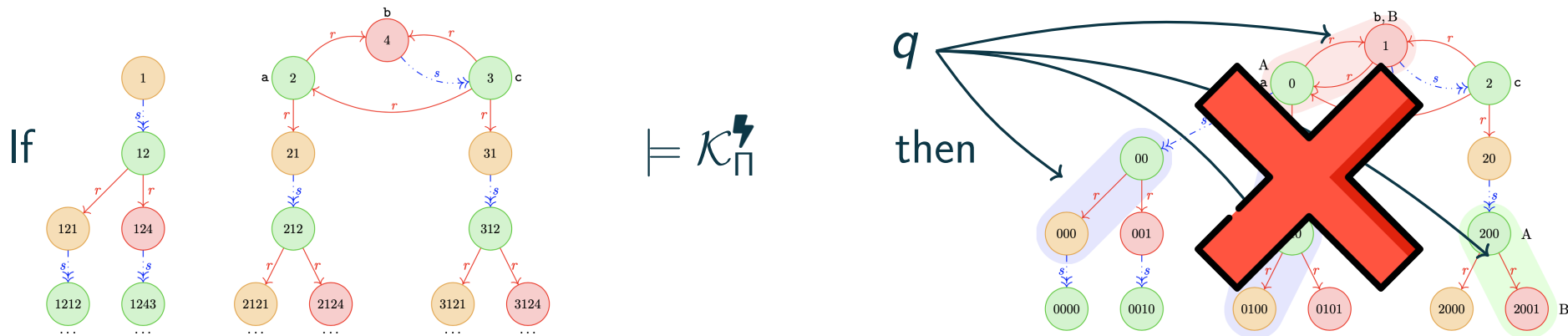
Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{LF}}$  “blocking” all matches yielding such a partition

## Key Lemma

$(\mathcal{A}, \mathcal{T}) \not\models q$  if and only if

# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{lightning}}$  “blocking” all matches yielding such a partition

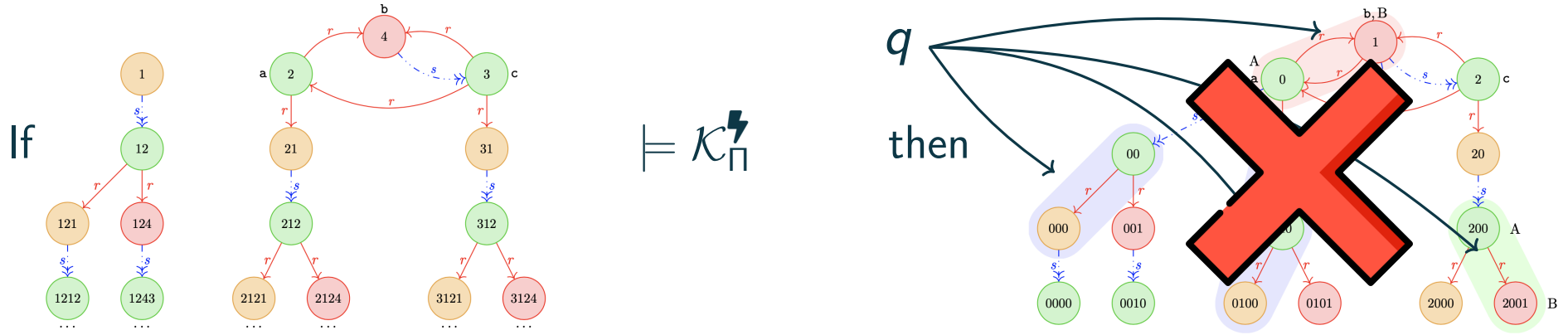
## Key Lemma

$(\mathcal{A}, \mathcal{T}) \not\models q$  if and only if  $(\mathcal{A}, \mathcal{T}) \cup \bigcup_{\text{all}} \mathcal{K}_{\Pi}^{\text{lightning}}$  is satisfiable.



# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{lightning}}$  “blocking” all matches yielding such a partition

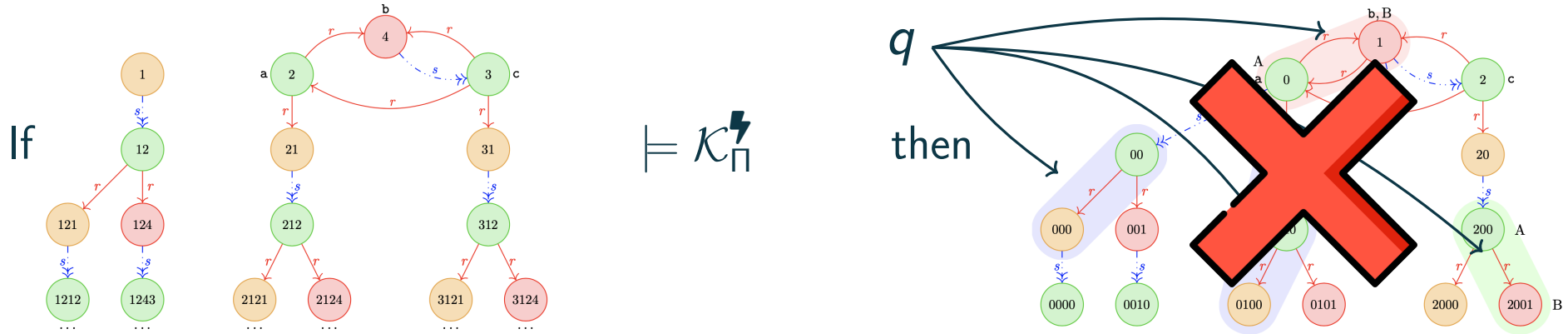
## Key Lemma

$(\mathcal{A}, \mathcal{T}) \not\models q$  if and only if  $(\mathcal{A}, \mathcal{T}) \cup \bigcup_{\text{all}} \mathcal{K}_{\Pi}^{\text{lightning}}$  is satisfiable.



# CQ Entailment over Locally-Forward Extensions of $\mathcal{ALC}$ (Simplified For This Talk)

- **Input:** an ABox  $\mathcal{A}$ , a TBox  $\mathcal{T}$ , and a conjunctive query  $q$ .
- **Proviso:**  $(\mathcal{A}, \mathcal{T}) \not\models q$  then, there is a locally-forward model of  $(\mathcal{A}, \mathcal{T})$  violating  $q$ .



A match of  $q$  partition query variables into: **(a)** roots, **(b)** trees dangling from roots, **(c)** trees far from roots.

Partition  $\Pi$  of  $q \rightsquigarrow$  a KB  $\mathcal{K}_{\Pi}^{\text{LF}}$  “blocking” all matches yielding such a partition

## Key Lemma

$(\mathcal{A}, \mathcal{T}) \not\models q$  if and only if  $(\mathcal{A}, \mathcal{T}) \cup \bigcup_{\text{all}} \mathcal{K}_{\Pi}^{\text{LF}}$  is satisfiable.



Effective reduction

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

Consequences?

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is  $2\text{EXPTIME}$ -hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is  $2\text{EXPTIME}$ -hard.
- Fluted Guarded Fragment with  $=$  has  $2\text{EXPTIME}$ -hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

- A reduction from the acceptance problem for the empty-tape  $\text{AEXPSPACE}$  TMs.



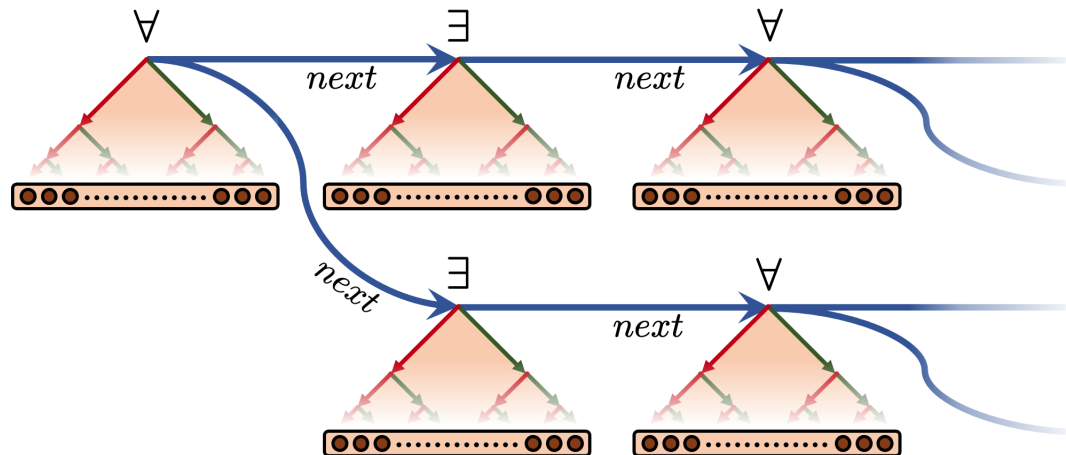
Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

- A reduction from the acceptance problem for the empty-tape AEXPSPACE TMs.



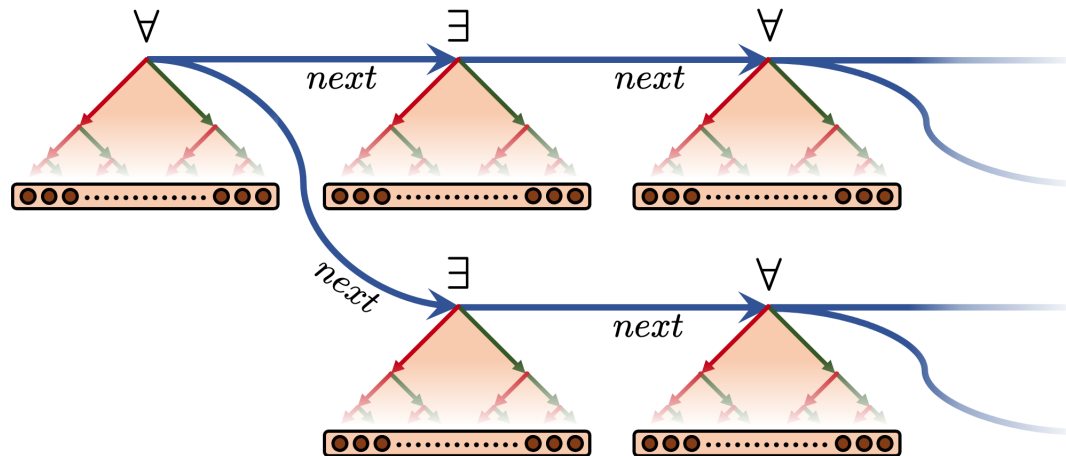
Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

- A reduction from the acceptance problem for the empty-tape AEXPSPACE TMs.



- The models of an  $\mathcal{ALC}^{\text{Self}}$ -KB  $\mathcal{K}_{\mathcal{M}}$  describe possibly faulty runs of a given ATM  $\mathcal{M}$ .

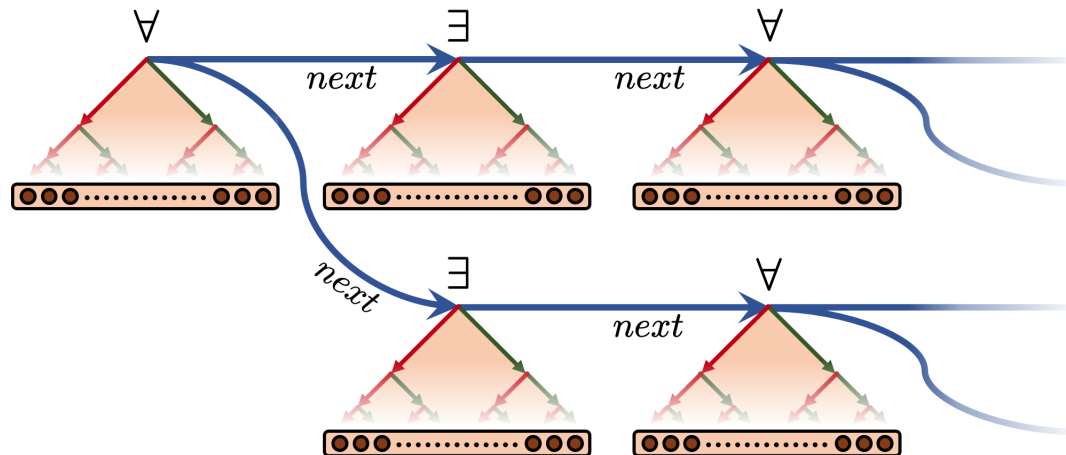
Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

- A reduction from the acceptance problem for the empty-tape AEXPSPACE TMs.



- The models of an  $\mathcal{ALC}^{\text{Self}}$ -KB  $\mathcal{K}_{\mathcal{M}}$  describe possibly faulty runs of a given ATM  $\mathcal{M}$ .
- A CQ  $q_{\mathcal{M}}$  detects mismatches in the consecutive transitions.

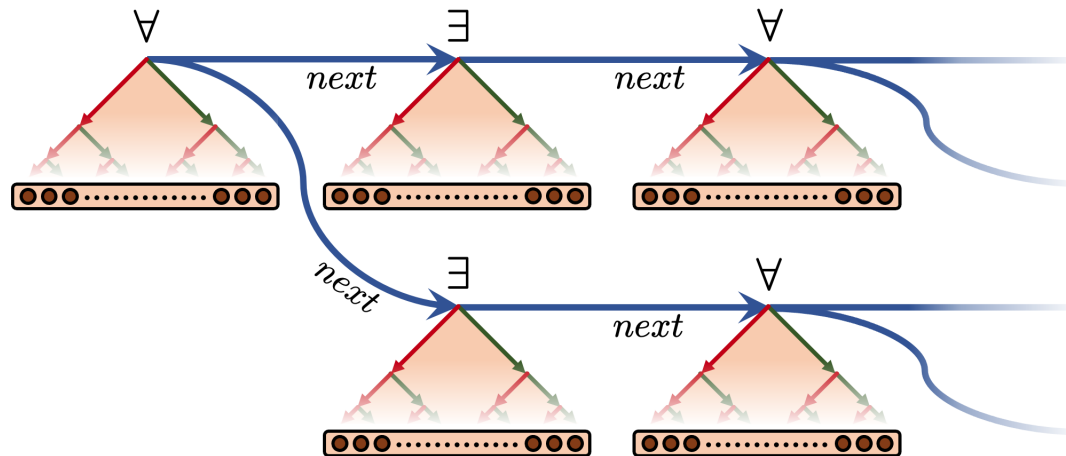
Conjunctive query entailment over  $\mathcal{ALC}^{\text{Self}}$  TBoxes is 2EXPTIME-hard.

## Consequences?

- Querying the  $\mathcal{Z}$  (a.k.a.  $\mathcal{ALCHb}_{\text{reg}}^{\text{Self}}$ ) family is 2EXPTIME-hard.
- Fluted Guarded Fragment with  $=$  has 2EXPTIME-hard CQ querying (contrasts [B'21, JELIA])

## Proof scheme?

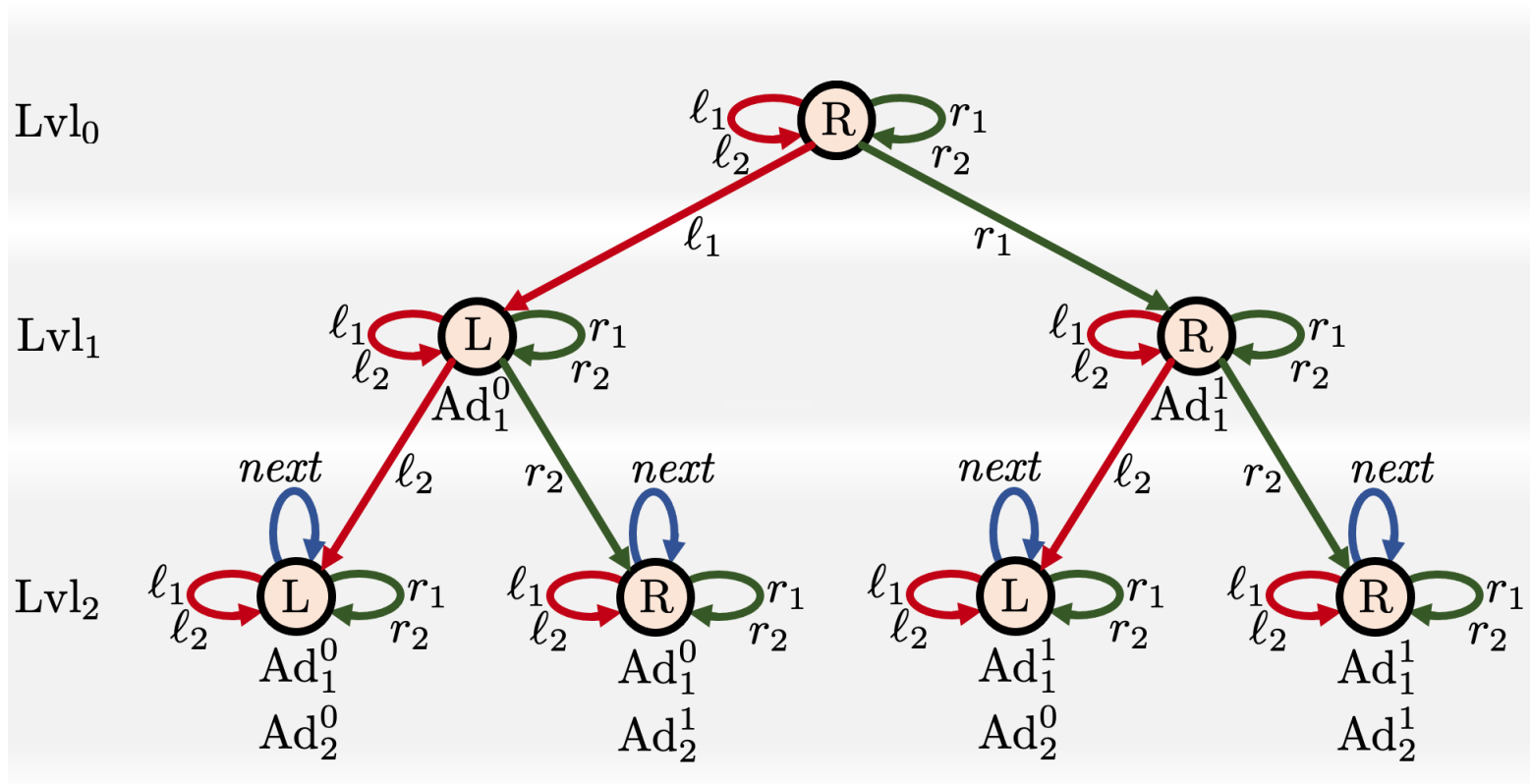
- A reduction from the acceptance problem for the empty-tape AEXPSPACE TMs.



- The models of an  $\mathcal{ALC}^{\text{Self}}$ -KB  $\mathcal{K}_{\mathcal{M}}$  describe possibly faulty runs of a given ATM  $\mathcal{M}$ .
- A CQ  $q_{\mathcal{M}}$  detects mismatches in the consecutive transitions.
- $\mathcal{K}_{\mathcal{M}} \not\models q_{\mathcal{M}}$  iff there is a (non-faulty) accepting run of  $\mathcal{M}$ .

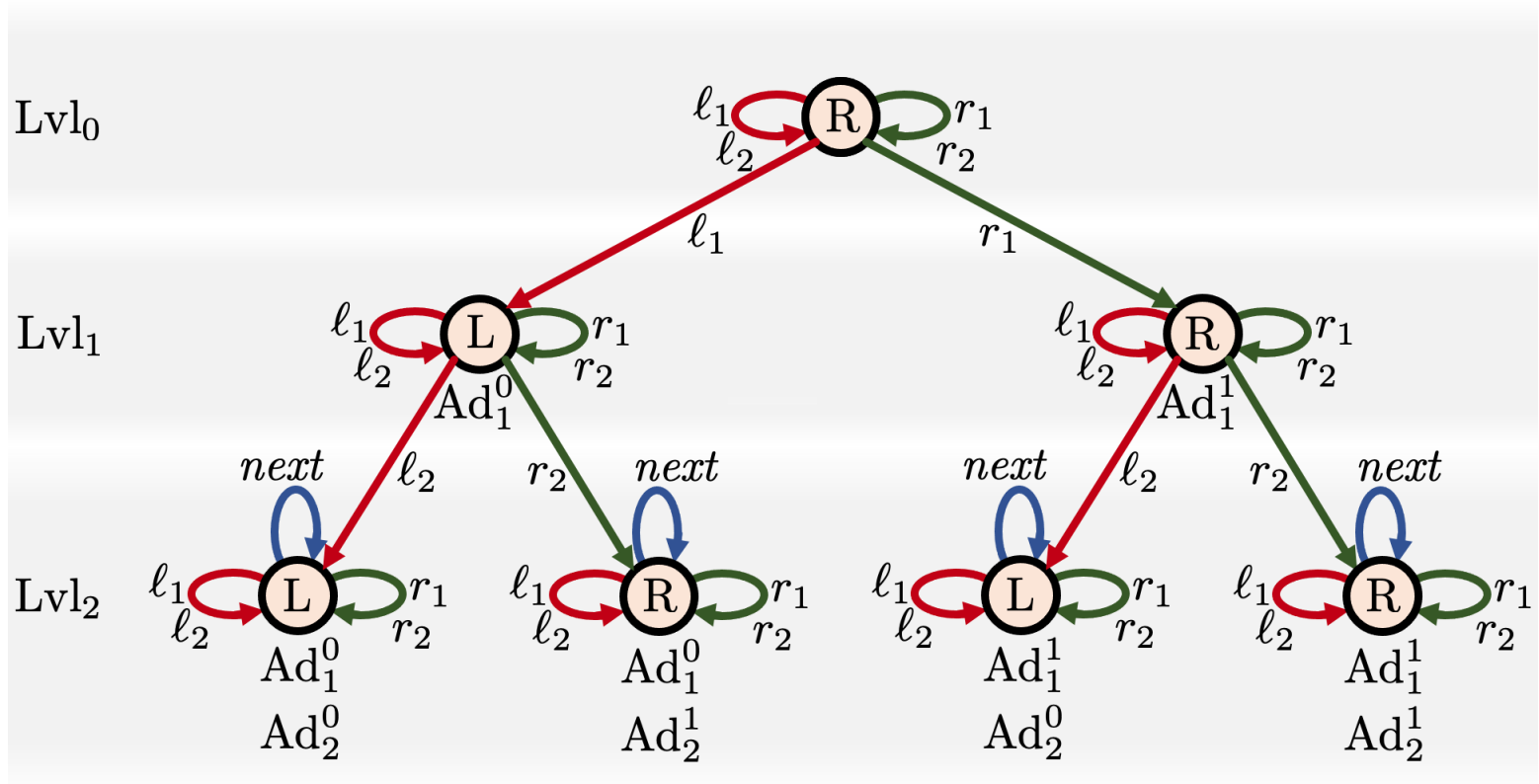
## Trick no. 1: A single root-to-leaves conjunctive query

## Trick no. 1: A single root-to-leaves conjunctive query



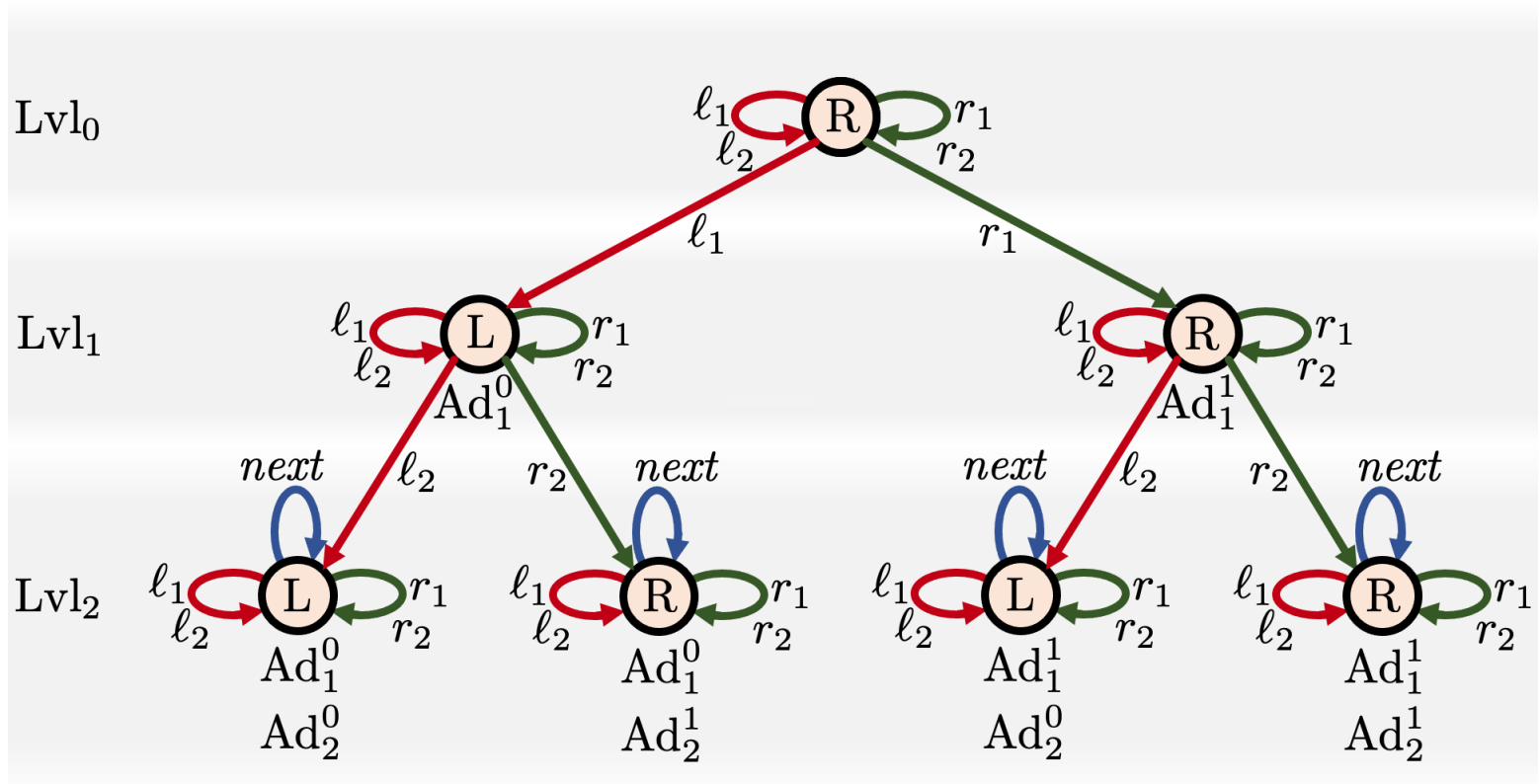
## Trick no. 1: A single root-to-leaves conjunctive query

Goal: Design a CQ  $q(x, y)$  such that  $x$  matches the root and  $y$  matches any of the leaves.



## Trick no. 1: A single root-to-leaves conjunctive query

Goal: Design a CQ  $q(x, y)$  such that  $x$  matches the root and  $y$  matches any of the leaves.

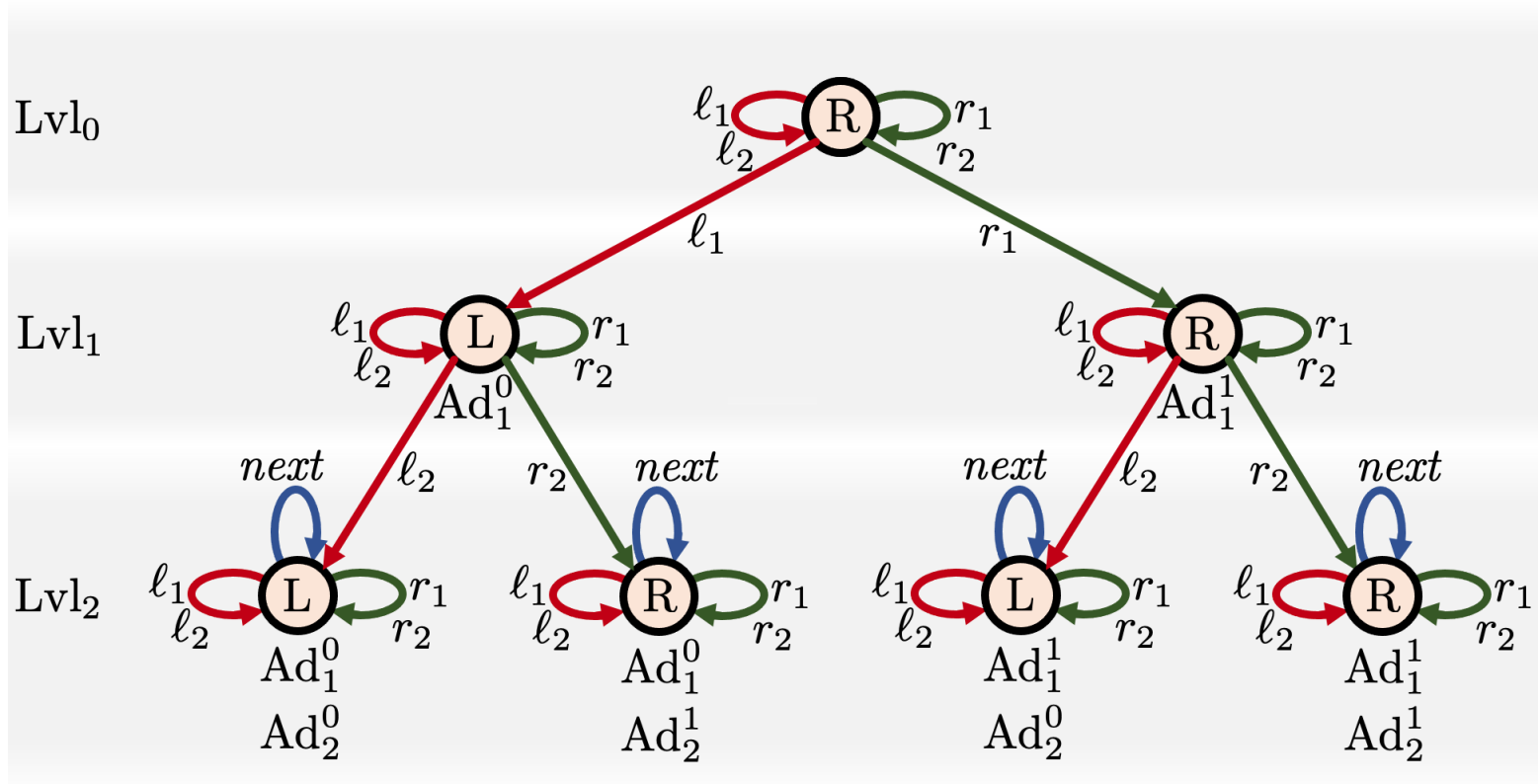


$$\exists x_1 \exists x_2 \exists x_3 \text{ Lvl}_0(x) \wedge l_1(x, x_1) \wedge r_1(x_1, x_2) \wedge l_2(x_2, x_3) \wedge r_2(x_3, y) \wedge \text{Lvl}_2(y)$$



## Trick no. 1: A single root-to-leaves conjunctive query

Goal: Design a CQ  $q(x, y)$  such that  $x$  matches the root and  $y$  matches any of the leaves.

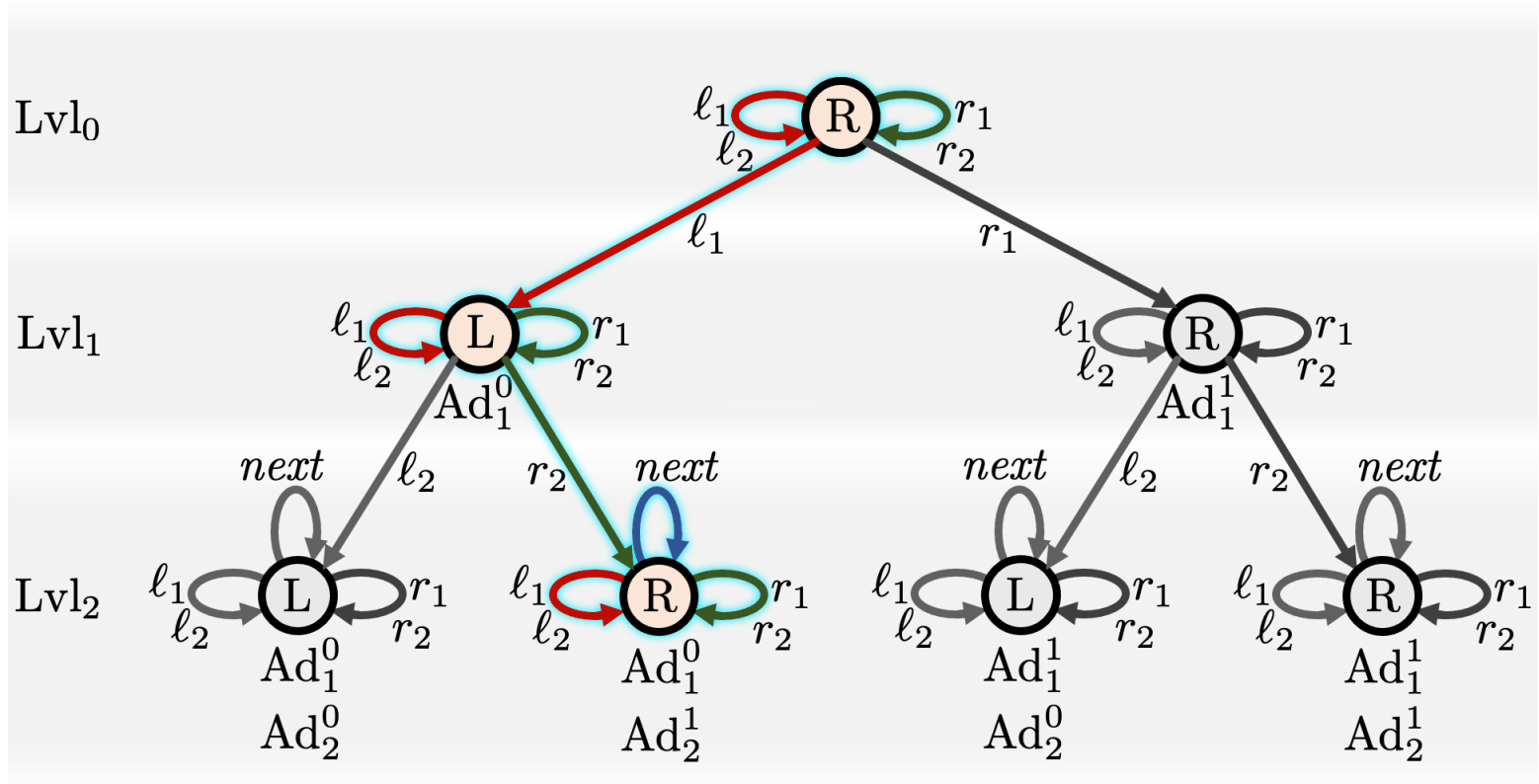


$$\exists x_1 \exists x_2 \exists x_3 \text{ Lvl}_0(x) \wedge l_1(x, x_1) \wedge r_1(x_1, x_2) \wedge l_2(x_2, x_3) \wedge r_2(x_3, y) \wedge \text{Lvl}_2(y)$$

For brevity we write:  $(\text{Lvl}_0?; l_1; r_1; l_2; r_2; \text{Lvl}_2?)(x, y)$ .

## Trick no. 1: A single root-to-leaves conjunctive query

Goal: Design a CQ  $q(x, y)$  such that  $x$  matches the root and  $y$  matches any of the leaves.

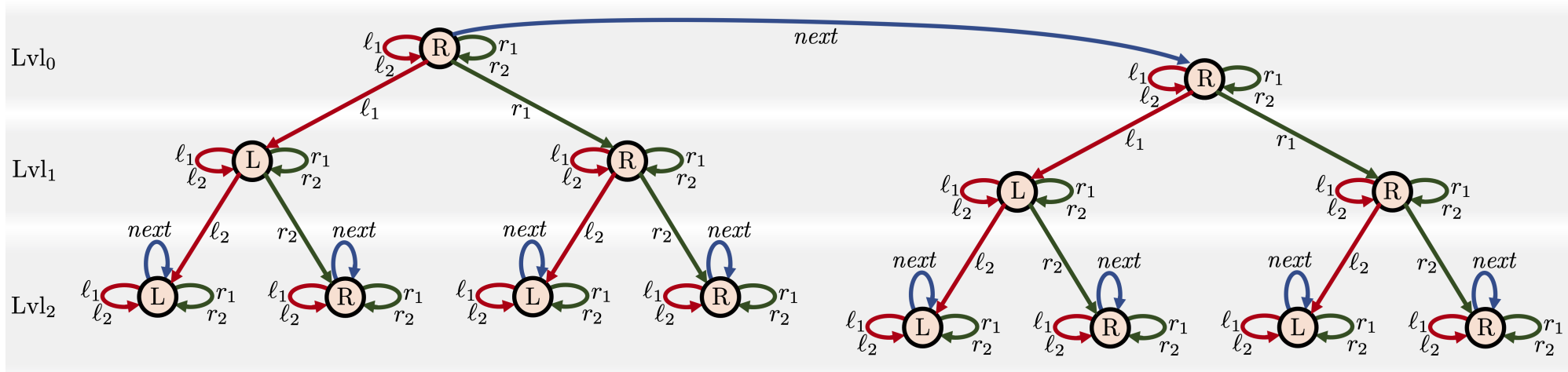


$$\exists x_1 \exists x_2 \exists x_3 \text{Lvl}_0(x) \wedge l_1(x, x_1) \wedge r_1(x_1, x_2) \wedge l_2(x_2, x_3) \wedge r_2(x_3, y) \wedge \text{Lvl}_2(y)$$

For brevity we write:  $(\text{Lvl}_0?; l_1; r_1; l_2; r_2; \text{Lvl}_2?)(x, y)$ .

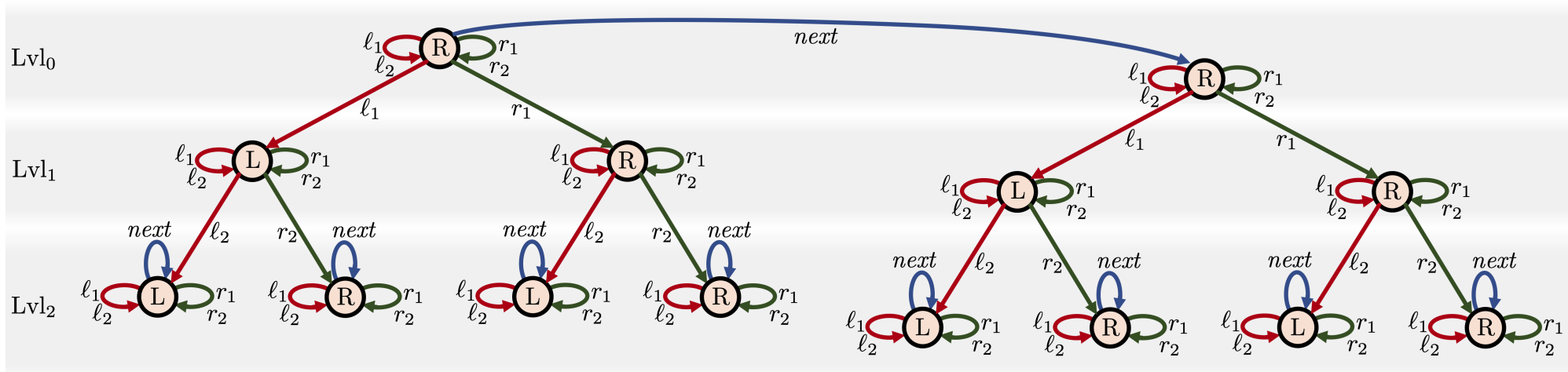
## Trick no. 2: Synchronisation of leaves among two trees

## Trick no. 2: Synchronisation of leaves among two trees



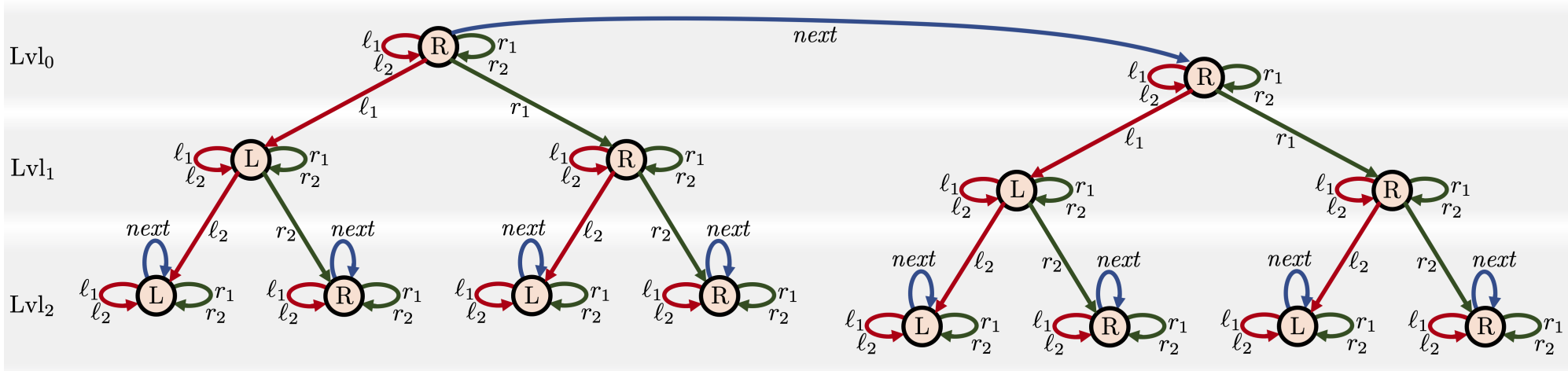
## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



## Trick no. 2: Synchronisation of leaves among two trees

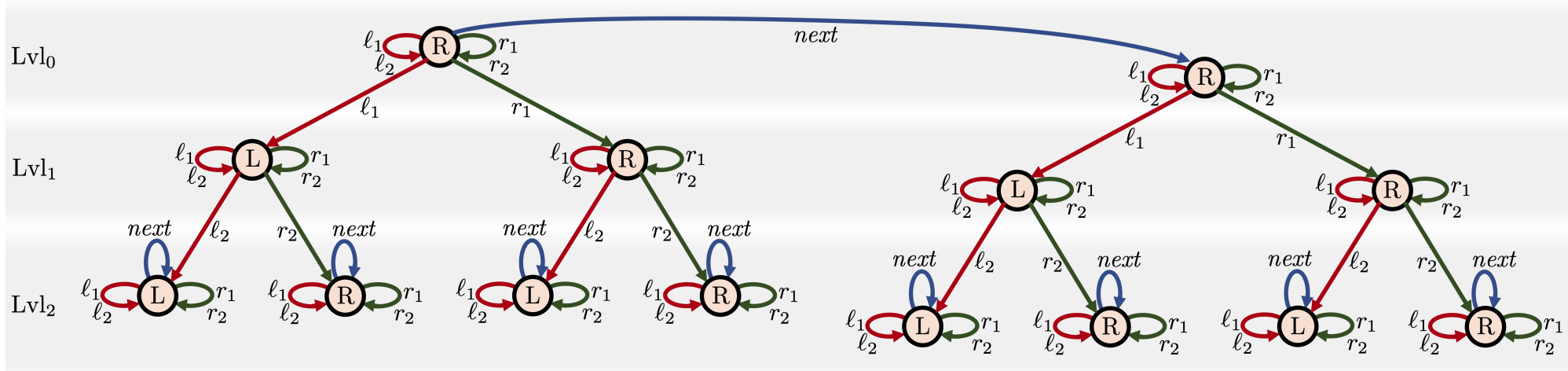
Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select two leaves located in different trees:

## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.

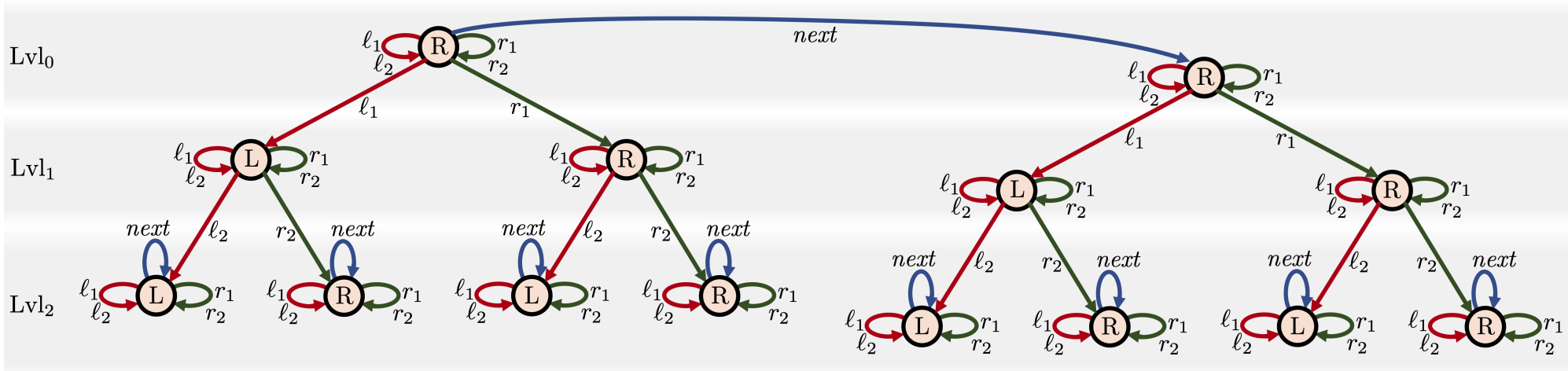


Select two leaves located in different trees:

$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select **two leaves** located in **different trees**:

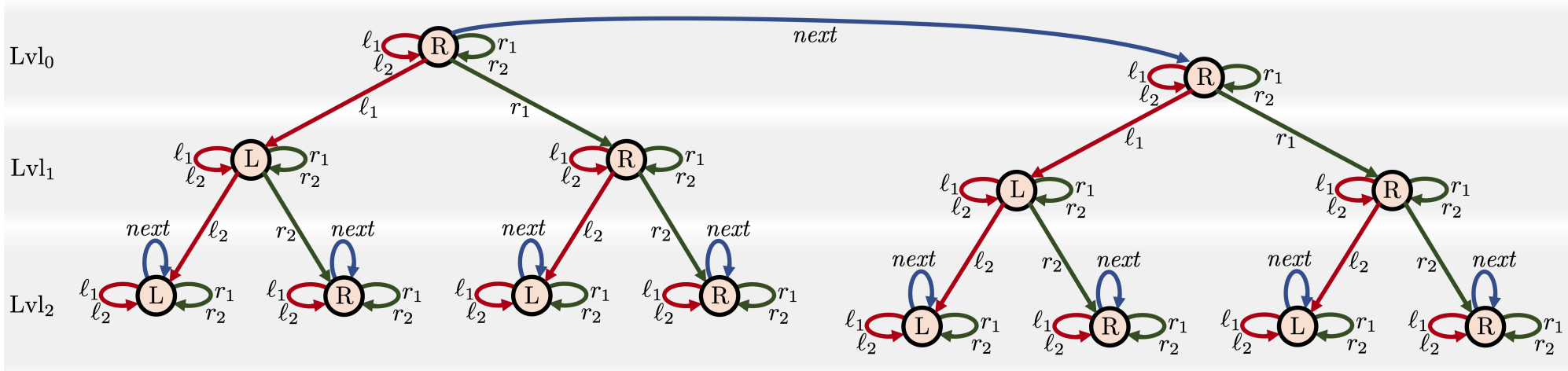
$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

Impose that they have **the same first bit** of their address:



## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select **two leaves** located in **different trees**:

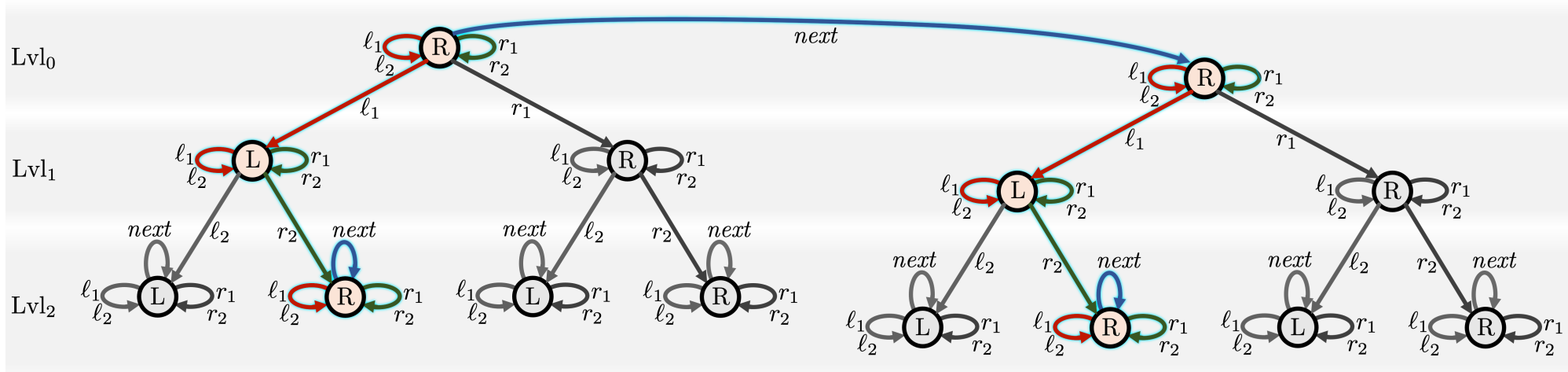
$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

Impose that they have **the same first bit** of their address:

$$\wedge (r_2^-; l_2^-; l_1^-; next; l_1; l_2; r_2; Lvl_2?; r_2^-; l_2^-; r_1^-; next; r_1; l_2; r_2)(x, y)$$

## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select **two leaves** located in **different trees**:

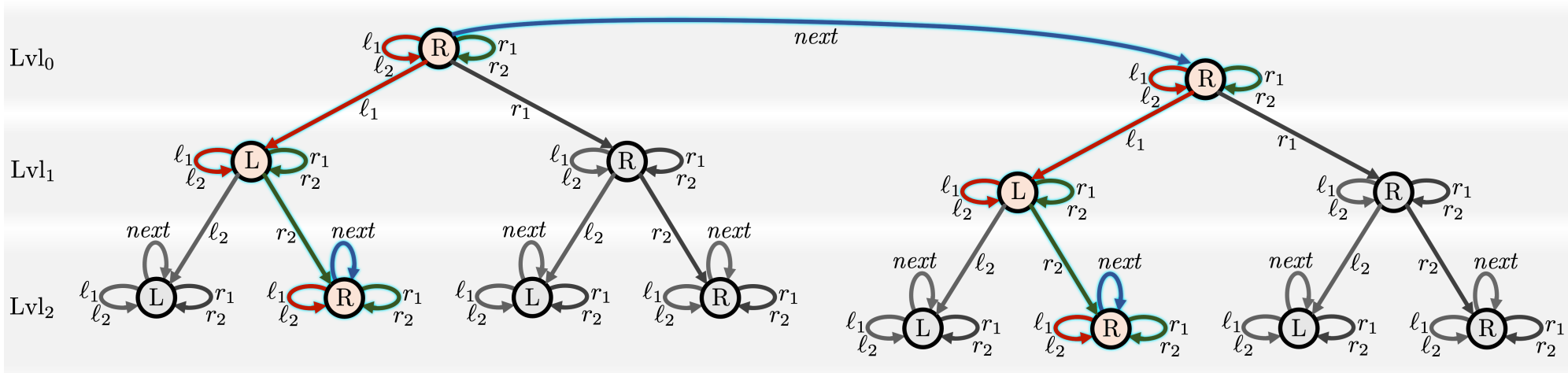
$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

Impose that they have **the same first bit** of their address:

$$\wedge (r_2^-; l_2^-; l_1^-; next; l_1; l_2; r_2; Lvl_2?; r_2^-; l_2^-; r_1^-; next; r_1; l_2; r_2)(x, y)$$

## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select **two leaves** located in **different trees**:

$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

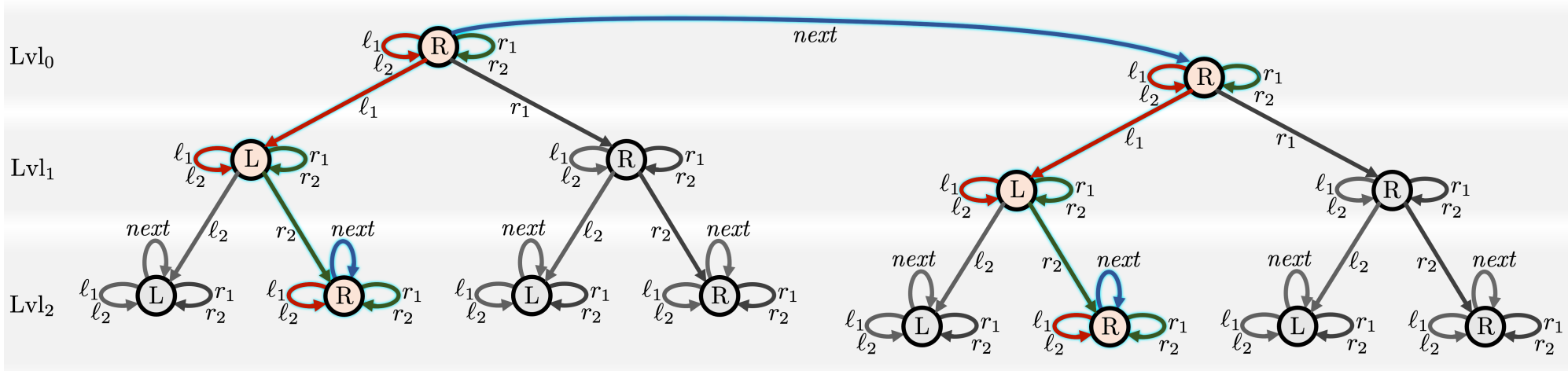
Impose that they have **the same first bit** of their address:

$$\wedge (r_2^-; l_2^-; l_1^-; next; l_1; l_2; r_2; Lvl_2?; r_2^-; l_2^-; r_1^-; next; r_1; l_2; r_2)(x, y)$$

as well as **the same second bit** of their address:

## Trick no. 2: Synchronisation of leaves among two trees

Goal: Design a CQ  $q(x, y)$  that matches leaves  $x, y$  with equal addresses.



Select **two leaves** located in **different trees**:

$$(Lvl_2?; r_2^-; l_2^-; r_1^-; l_1^-; Lvl_0?; next; Lvl_0?; l_1; r_1; l_2; r_2; Lvl_2?)(x, y)$$

Impose that they have **the same first bit** of their address:

$$\wedge (r_2^-; l_2^-; l_1^-; next; l_1; l_2; r_2; Lvl_2?; r_2^-; l_2^-; r_1^-; next; r_1; l_2; r_2)(x, y)$$

as well as **the same second bit** of their address:

$$\wedge (l_2^-; r_1^-; l_1^-; next; l_1; r_1; l_2; Lvl_2?; r_2^-; r_1^-; l_1^-; next; l_1; r_1; r_2)(x, y)$$

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions



Sebastian Rudolph (TU Dresden)



European Research Council  
Established by the European Commission



Emanuel Kieroński (Univ. of Wrocław)

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Querying  $\mathcal{Z}$  ( $ALCHb_{reg}^{Self}$ )?

SAT of (tamed)  $ZOIQ$ ?

Querying (tamed)  $ZOIQ$ ? Beyond Regularity?



$ZQ$  without Self  $\in$  EXP

$ALC$ Self is 2EXP-hard

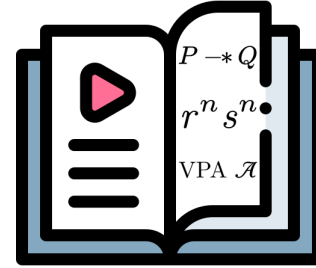


SAT in NP (data-comp)



New 2EXP upper bounds

$ZOI$  and  $ZOQ$  are FC



Undecidability



solo + award



# Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Querying  $\mathcal{Z}$  ( $ALCHb_{reg}^{Self}$ )?

SAT of (tamed)  $ZOIQ$ ?

Querying (tamed)  $ZOIQ$ ? Beyond Regularity?



$ZQ$  without Self  $\in$  EXP

$ALC$ Self is 2EXP-hard

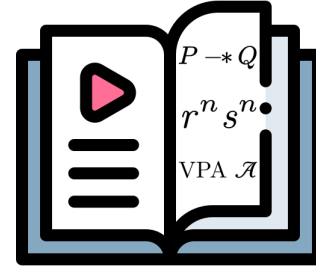


SAT in NP (data-comp)



New 2EXP upper bounds

$ZOI$  and  $ZOQ$  are FC

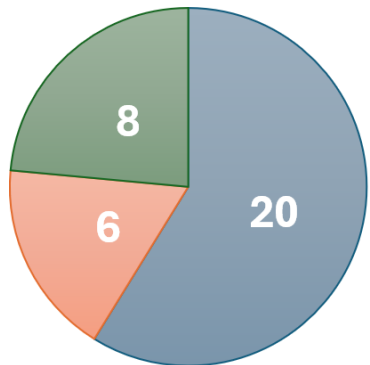


Undecidability



solo + award

## Other relevant statistics about the PhD Candidate



■ Conference ■ Workshop ■ Journal

- Dissertation = 6 conference papers (**4 A\*-ranked** papers!) + 3 journal papers
- Best student **paper awards** at JELIA 2021 and JELIA 2023.
- PI in own **prestigious polish research grant** 2018–2022 realised **with distinction**.
- Polish Ministry of Science **Award for Outstanding Young Scientists** ( $\approx$  45k euro).
- Supervised **8 students** resulting in **3 joint publications**.