Data Complexity in
Expressive Description Logics With Path Expressions

Bartosz Bednarczyk

Dresden Seminar 12.06.24 & DL Workshop 21.06.24 & IJCAI'24 05–08.08.24

TECHNISCHE UNIVERSITÄT DRESDEN

Uniwersytet Wrocławski

DeciGUT

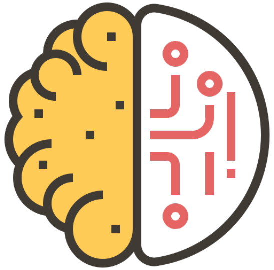
European Research Council
Established by the European Commission

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

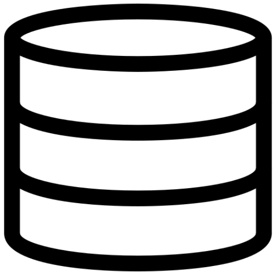
Database (ABox)

Knowledge (TBox)

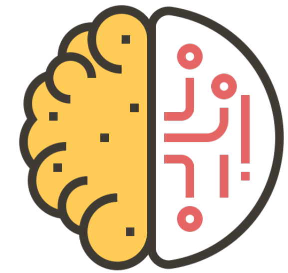


Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



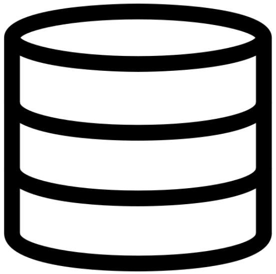
Knowledge (TBox)



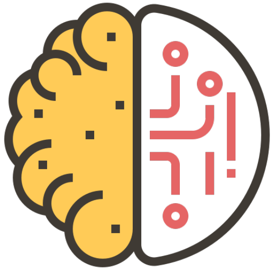
$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



Knowledge (TBox)

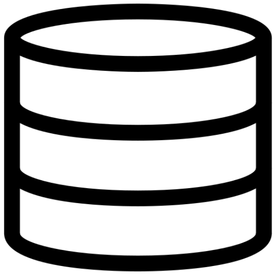


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

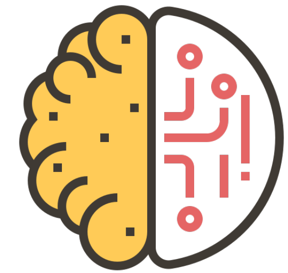
$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



Knowledge (TBox)



$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

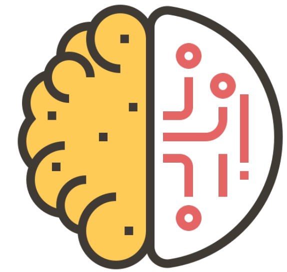
$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

Knowledge (TBox)



$hasParent(\texttt{Heracles}, \texttt{Zeus})$

$Diety(\texttt{Zeus})$, $Female(\texttt{Rhea})$

$Narcissist(\texttt{Narcissus})$

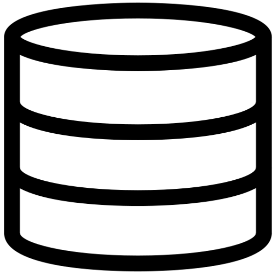
$\mathcal{ALC}$

$Mortal \sqsubseteq \neg Diety$

$\top \sqsubseteq \exists hasParent.Male \sqcap \exists hasParent.Female$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

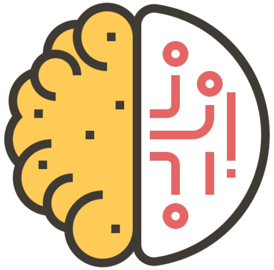


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

Knowledge (TBox)



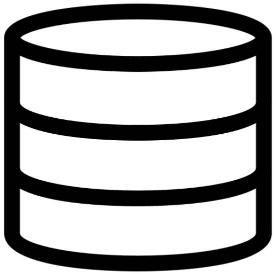
$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

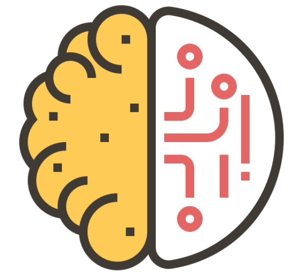


$\mathit{hasParent}(\texttt{Heracles}, \texttt{Zeus})$

$\mathit{Diety}(\texttt{Zeus})$, $\mathit{Female}(\texttt{Rhea})$

$\mathit{Narcissist}(\texttt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

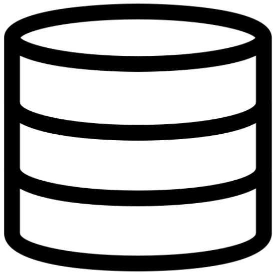
$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Self

$$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

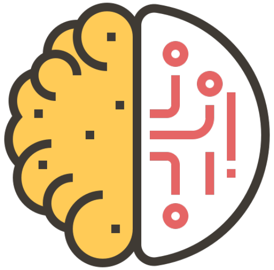


$\mathit{hasParent}(\texttt{Heracles}, \texttt{Zeus})$

$\mathit{Diety}(\texttt{Zeus}), \mathit{Female}(\texttt{Rhea})$

$\mathit{Narcissist}(\texttt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

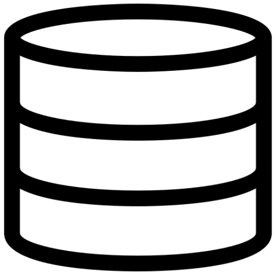
$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Self

$$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

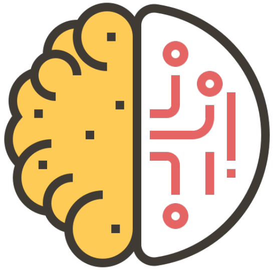


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Self

$$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$$

$\mathcal{H}b$

$$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

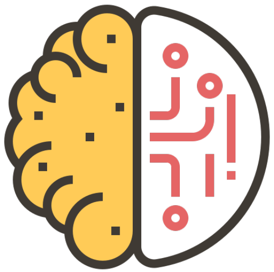
Database (ABox)

$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Self

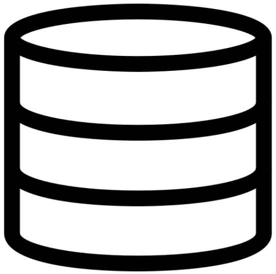
$$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$$

$\mathcal{H}b$

$$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

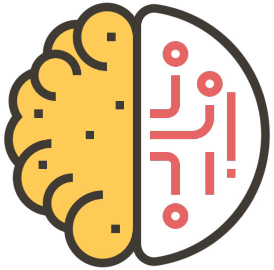


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus}), \mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

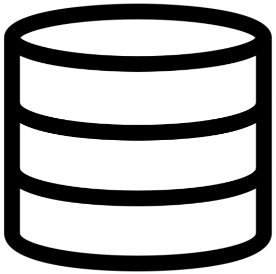
Knowledge (TBox)



$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$ $\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

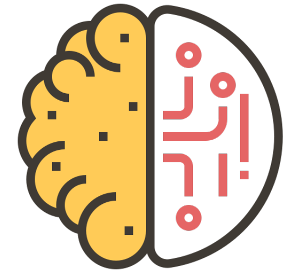


$\mathit{hasParent}(\texttt{Heracles}, \texttt{Zeus})$

$\mathit{Diety}(\texttt{Zeus}), \mathit{Female}(\texttt{Rhea})$

$\mathit{Narcissist}(\texttt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$

$$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$$

$$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$$

Self

$$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$$

$\mathcal{H}b$

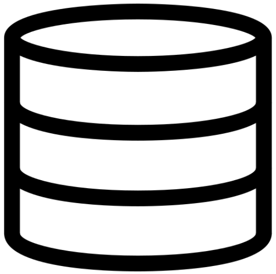
$$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$$

reg

$$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

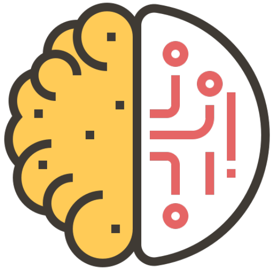


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus}), \mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

Knowledge (TBox)



$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$ $\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathrm{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\mathtt{Zeus}\} \sqsubseteq (= 54 \mathit{hasChildren}).\top$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

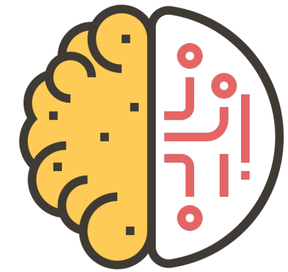
Database (ABox)

$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus}), \mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

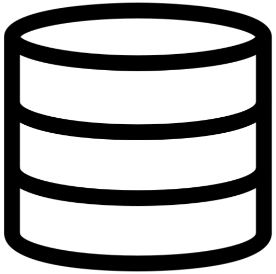
Knowledge (TBox)



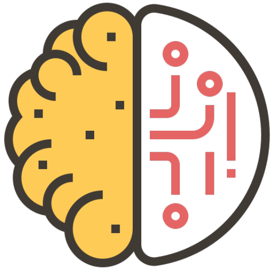
$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$ $\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\mathtt{Zeus}\} \sqsubseteq (= 54 \mathit{hasChildren}).\top$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



Knowledge (TBox)



$\mathit{hasParent}(\texttt{Heracles}, \texttt{Zeus})$

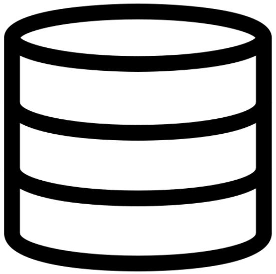
$\mathit{Diety}(\texttt{Zeus})$, $\mathit{Female}(\texttt{Rhea})$

$\mathit{Narcissist}(\texttt{Narcissus})$

$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$ $\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\texttt{Zeus}\} \sqsubseteq (= 54 \mathit{hasChildren}).\top$
$\mathcal{O}$ & $\mathcal{I}$	$\{\texttt{Ares}\} \sqsubseteq \exists \mathit{hasChildren}^{-}.\{\texttt{Zeus}\}$

Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)

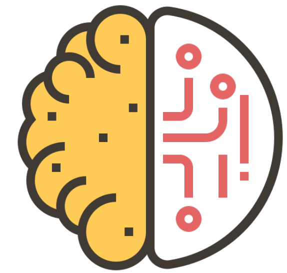


$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

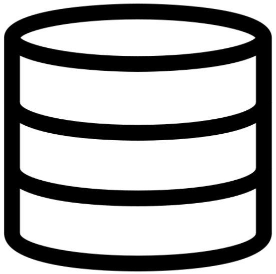
Knowledge (TBox)



$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$ $\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\mathtt{Zeus}\} \sqsubseteq (= 54\,\mathit{hasChildren}).\top$
$\mathcal{O}$ & $\mathcal{I}$	$\{\mathtt{Ares}\} \sqsubseteq \exists \mathit{hasChildren}^{-}.\{\mathtt{Zeus}\}$

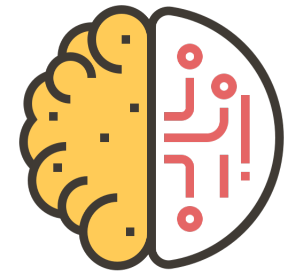
Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



We study the DL $\mathcal{ZOIQ}$ a.k.a. $\mathcal{ALCH}b^{\mathsf{reg}}_{\mathsf{Self}}\mathcal{OIQ}$.

Knowledge (TBox)



$hasParent(\mathtt{Heracles}, \mathtt{Zeus})$

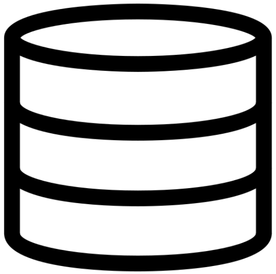
$Diety(\mathtt{Zeus})$, $Female(\mathtt{Rhea})$

$Narcissist(\mathtt{Narcissus})$

$\mathcal{ALC}$	$Mortal \sqsubseteq \neg Diety$ $\top \sqsubseteq \exists hasParent.Male \sqcap \exists hasParent.Female$
Self	$Narcissist \sqsubseteq \exists loves.\mathsf{Self}$
$\mathcal{H}b$	$hasParent \equiv hasMother \cup hasFather$
reg	$diety \sqsubseteq \forall hasParent^{*}.diety$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\mathtt{Zeus}\} \sqsubseteq (= 54 hasChildren).\top$
$\mathcal{O}$ & $\mathcal{I}$	$\{\mathtt{Ares}\} \sqsubseteq \exists hasChildren^{-}.\{\mathtt{Zeus}\}$

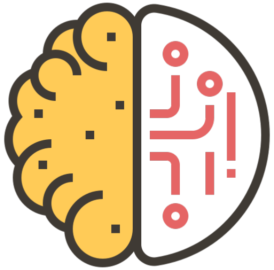
Running example: Greek mythology $\mathcal{ZOIQ}$ knowledge base

Database (ABox)



We study the DL $\mathcal{ZOIQ}$ a.k.a. $\mathcal{ALCHb}^{\mathrm{reg}}_{\mathrm{Self}}\mathcal{OIQ}$.

Knowledge (TBox)



$\mathit{hasParent}(\mathtt{Heracles}, \mathtt{Zeus})$

$\mathit{Diety}(\mathtt{Zeus})$, $\mathit{Female}(\mathtt{Rhea})$

$\mathit{Narcissist}(\mathtt{Narcissus})$

$\mathcal{ALC}$	$\mathit{Mortal} \sqsubseteq \neg \mathit{Diety}$
	$\top \sqsubseteq \exists \mathit{hasParent}.\mathit{Male} \sqcap \exists \mathit{hasParent}.\mathit{Female}$
Self	$\mathit{Narcissist} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$
$\mathcal{H}b$	$\mathit{hasParent} \equiv \mathit{hasMother} \cup \mathit{hasFather}$
reg	$\mathit{diety} \sqsubseteq \forall \mathit{hasParent}^{*}.\mathit{diety}$
$\mathcal{O}$ & $\mathcal{Q}$	$\{\mathtt{Zeus}\} \sqsubseteq (= 54\, \mathit{hasChildren}).\top$
$\mathcal{O}$ & $\mathcal{I}$	$\{\mathtt{Ares}\} \sqsubseteq \exists \mathit{hasChildren}^{-}.\{\mathtt{Zeus}\}$

This work: further study of KBSat for decidable fragments of $\mathcal{ZOIQ}$.

Reasoning Problems & Our Results

Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

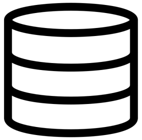
OUT: Is there an extension of $\mathcal{A}$ satisfying $\mathcal{T}$?

Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



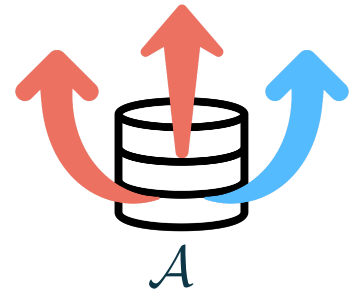
$\mathcal{A}$

Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

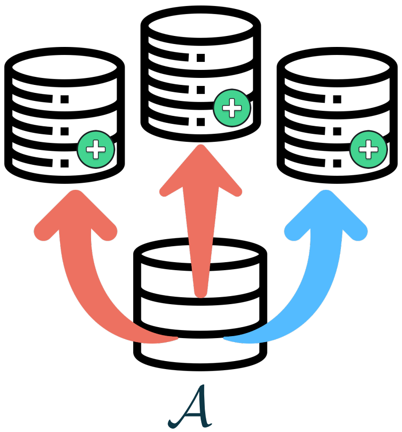


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

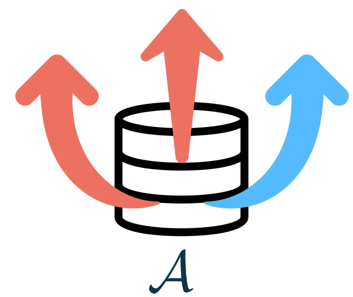


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

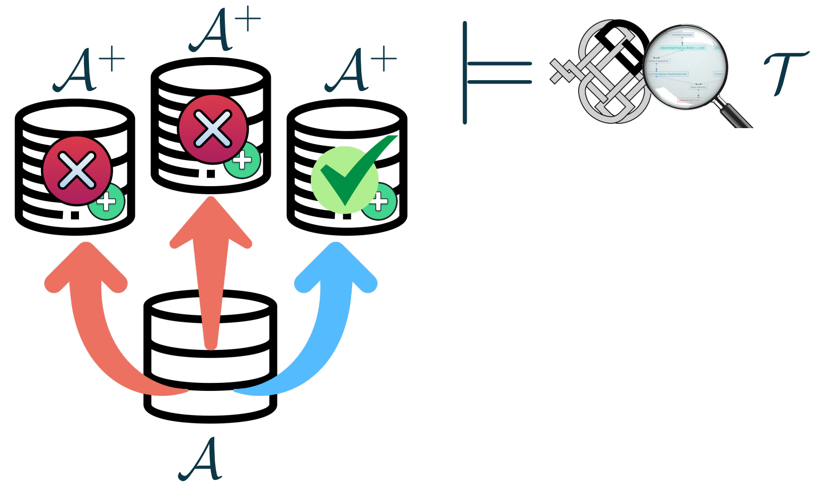


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



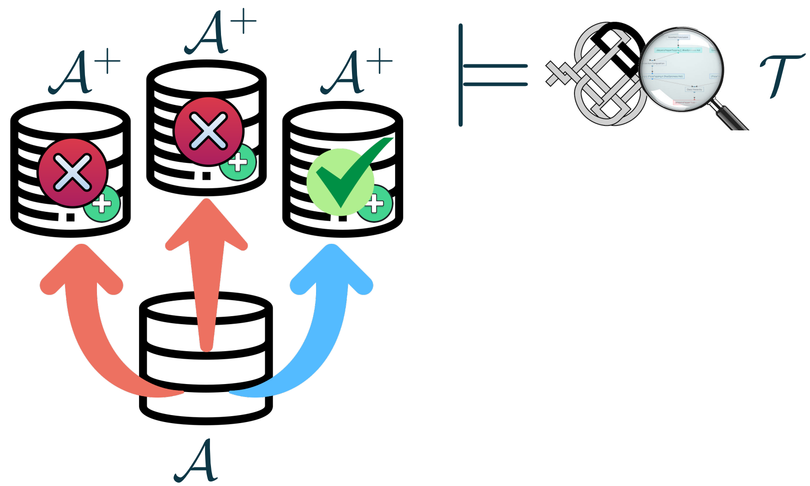
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

- Decidability of full $\mathcal{ZOIQ}$ is unknown!



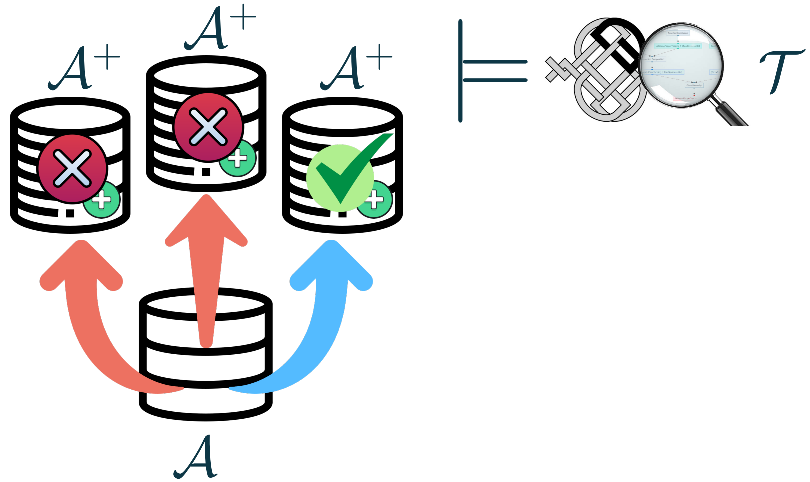
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.



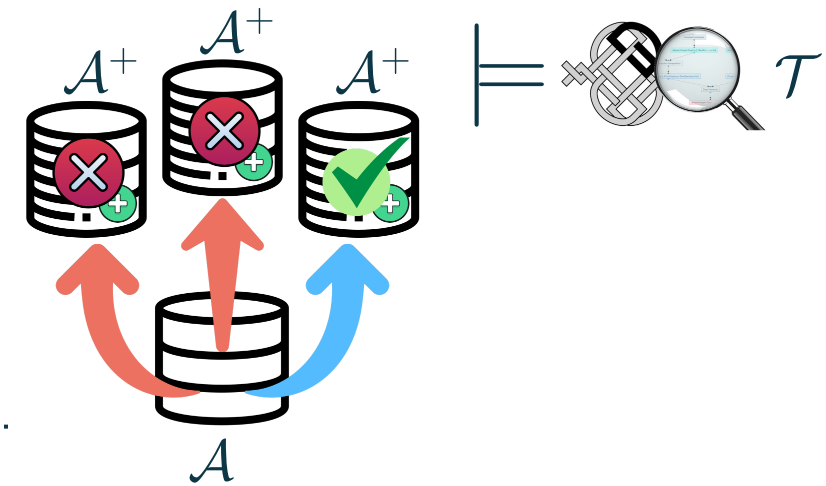
Reasoning Problems & Our Results

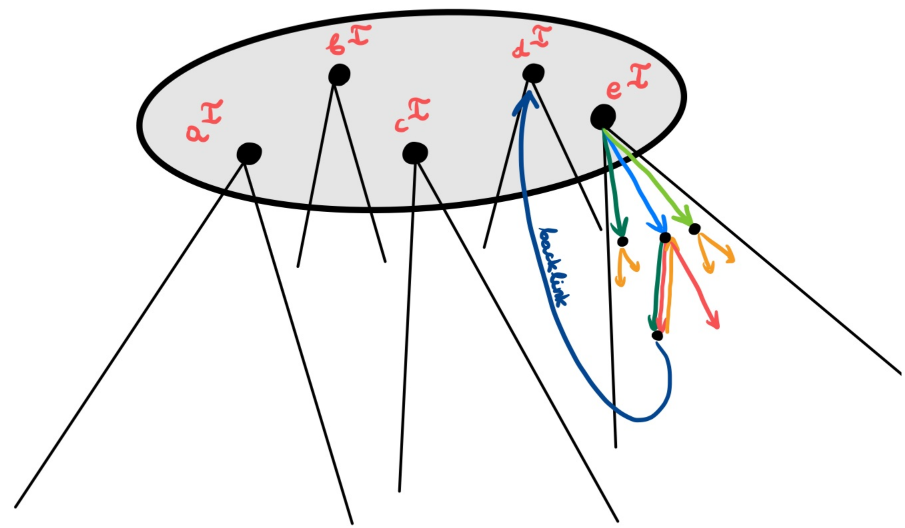
1. Satisfiability (consistency) problem.

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

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

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.





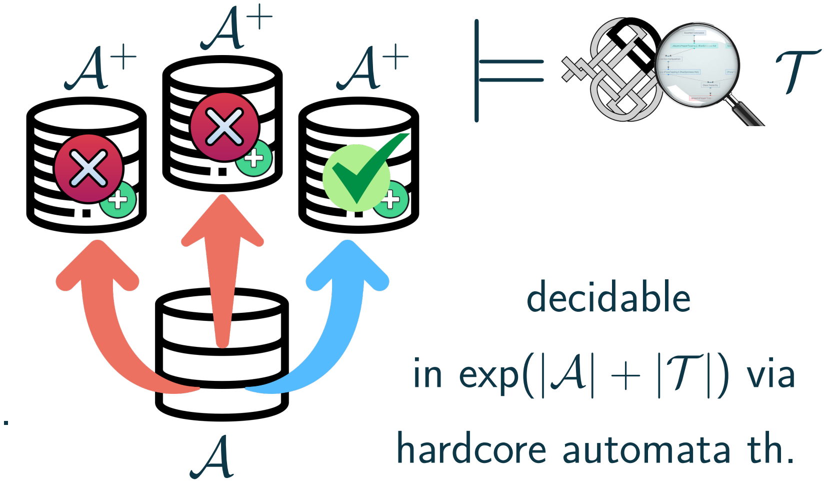
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

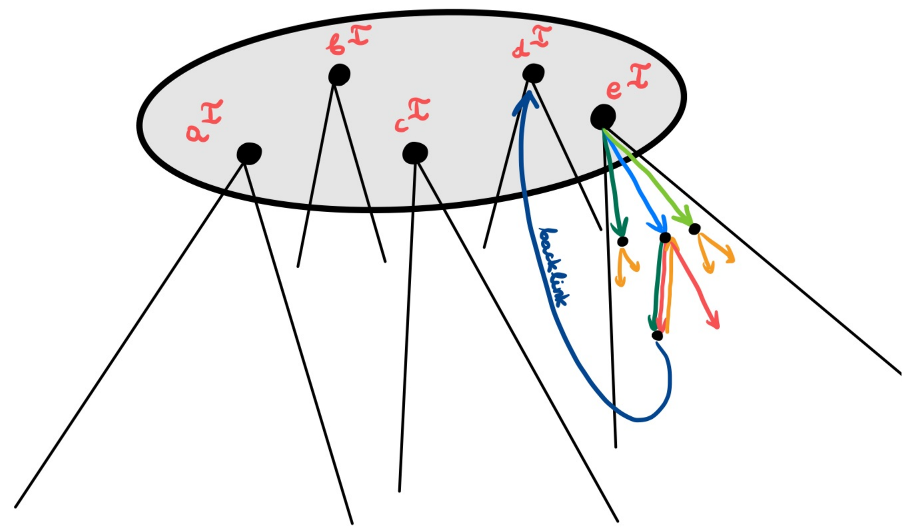
- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.



decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

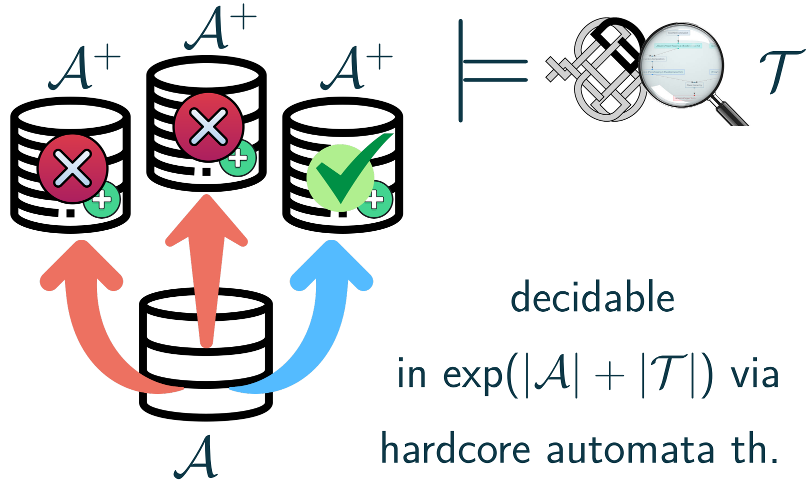


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



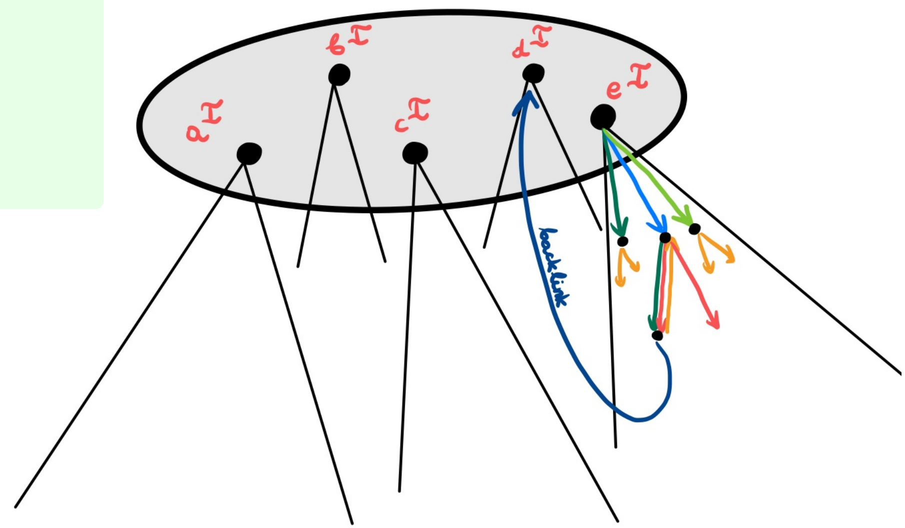
- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.

decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

The Main Result

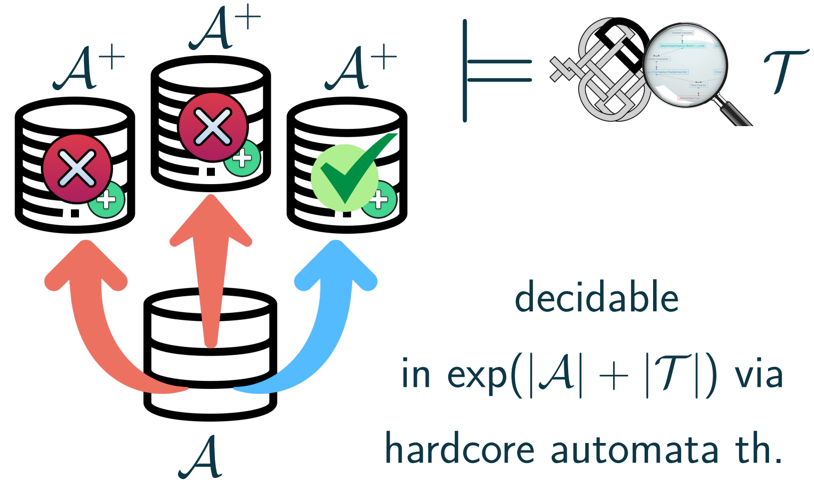


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.

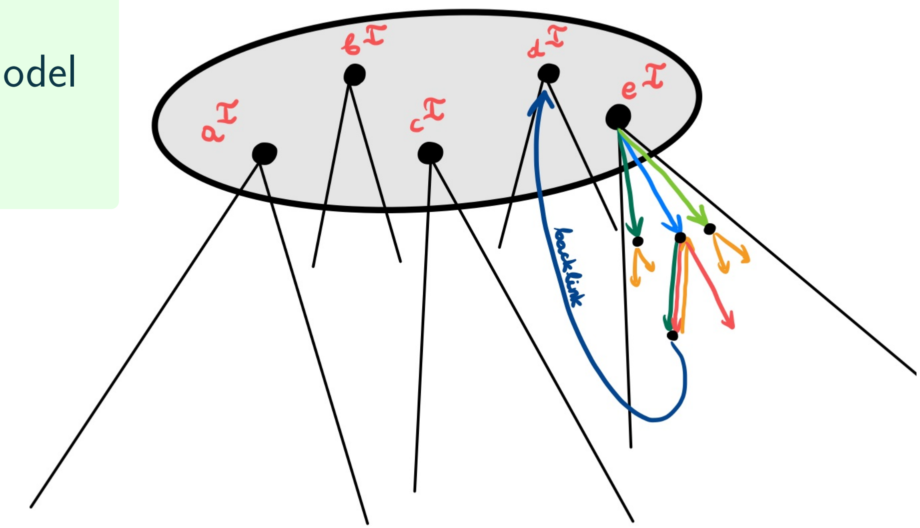
decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model

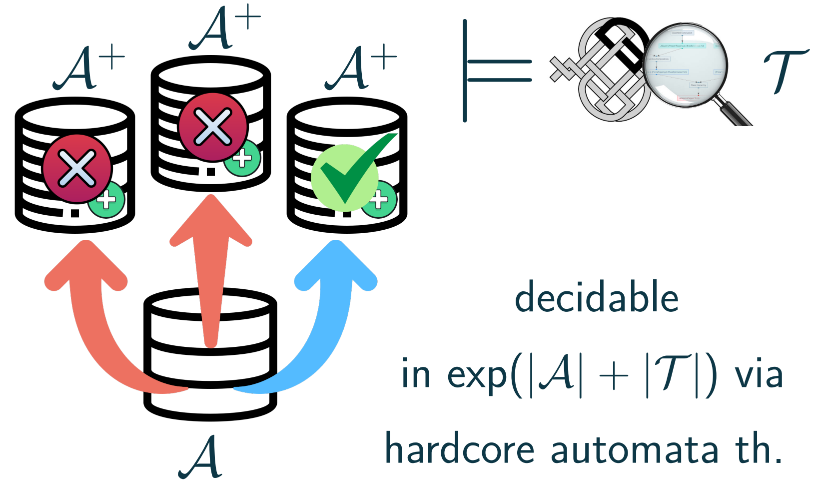


Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



decidable

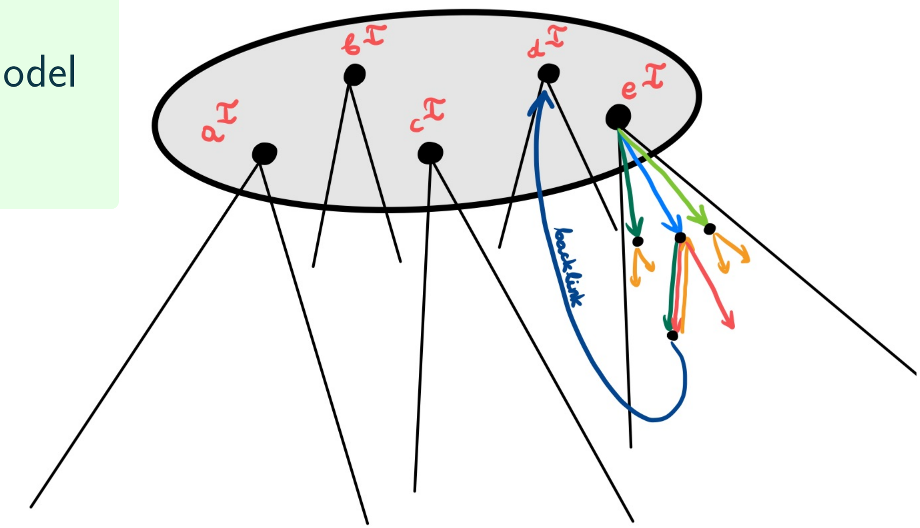
in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in



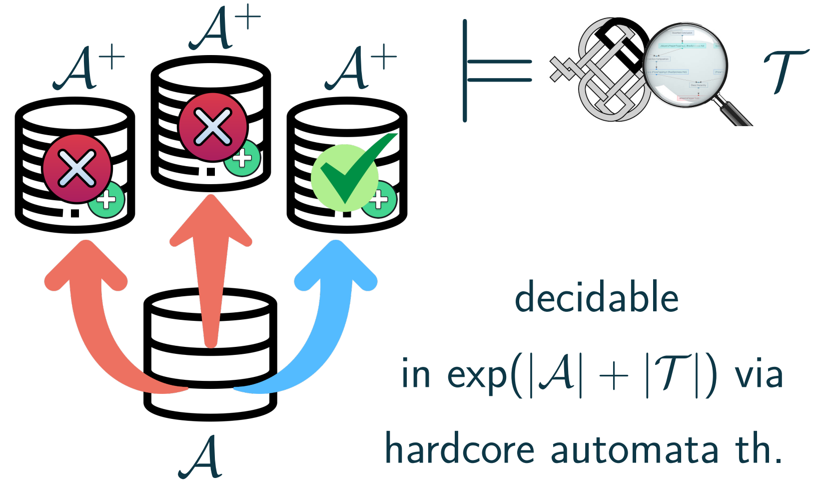
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.



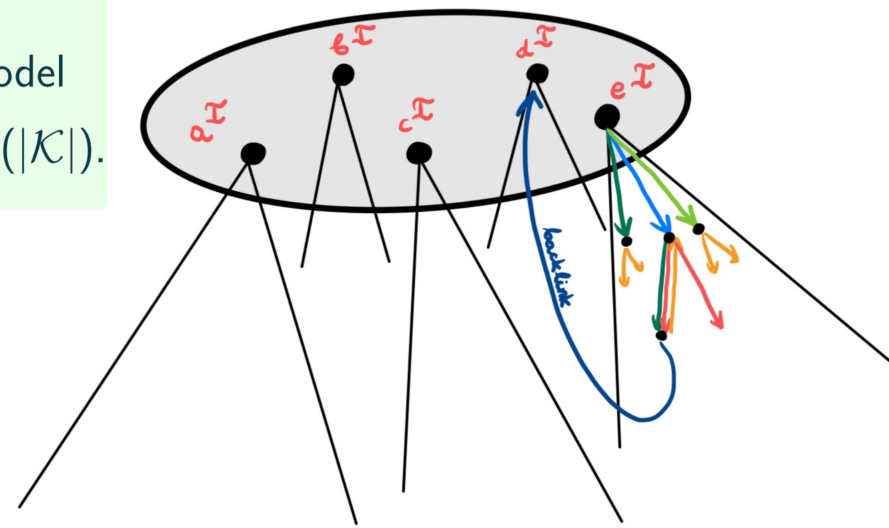
decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|)\cdot\text{poly}(|\mathcal{K}|)$.

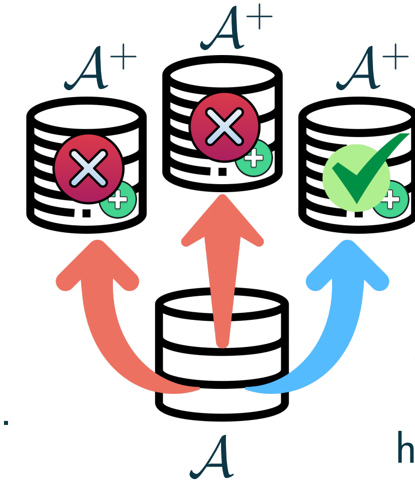


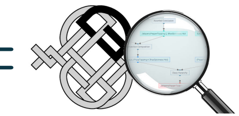
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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



$\models$  $\mathcal{T}$

decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

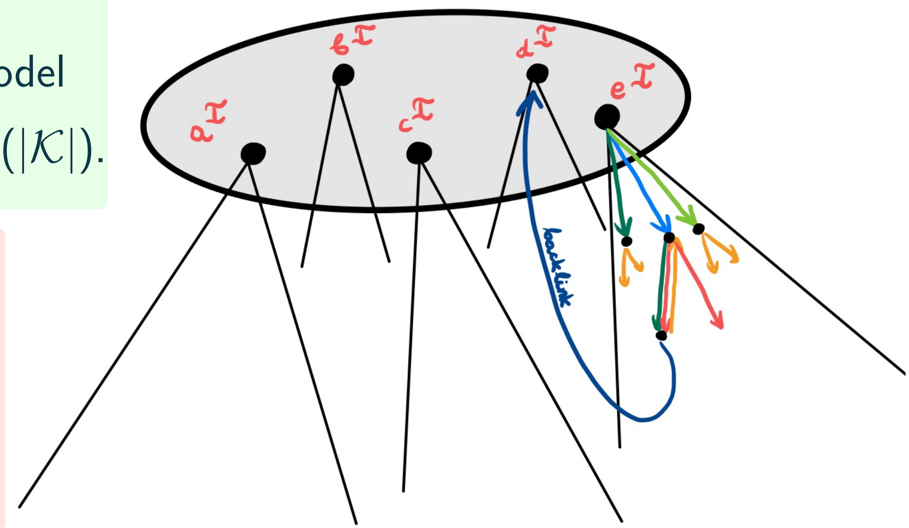
hardcore automata th.

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|)\cdot\text{poly}(|\mathcal{K}|)$.

Corollaries

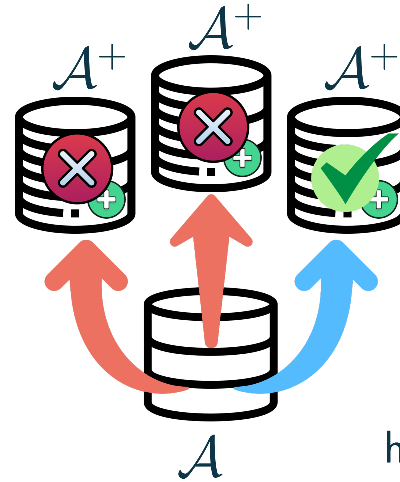


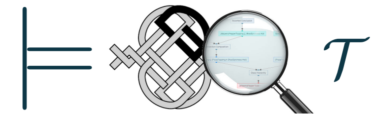
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

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

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





decidable

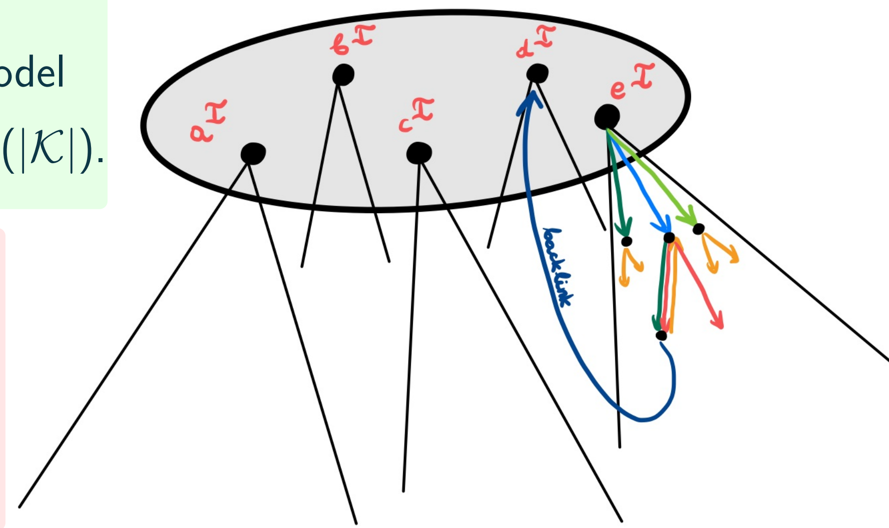
in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|)\cdot\text{poly}(|\mathcal{K}|)$.



Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.

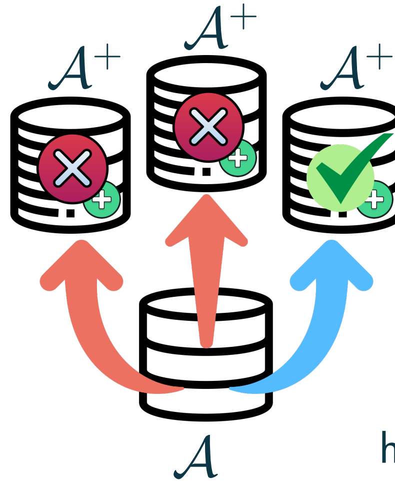
Reasoning Problems & Our Results

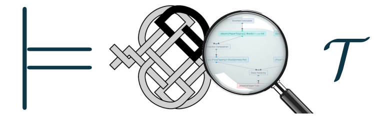
1. Satisfiability (consistency) problem.

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

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

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.





decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

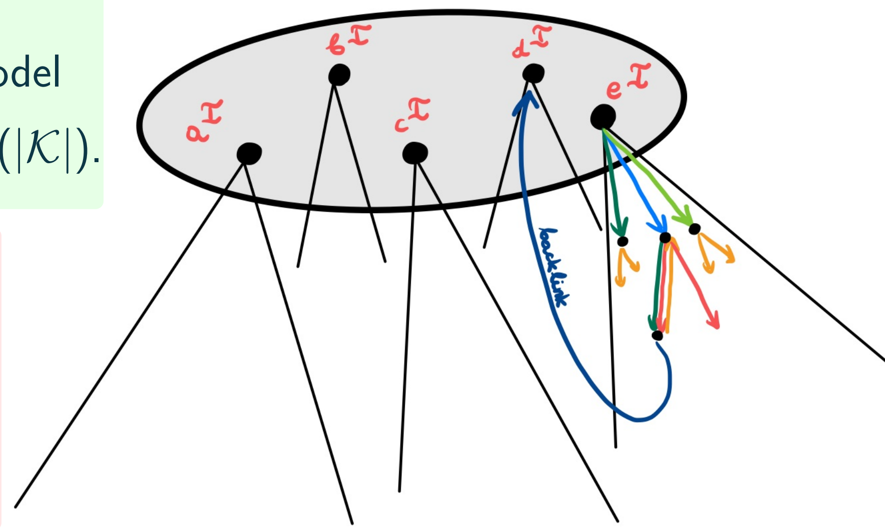
hardcore automata th.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|)$ + nondet-poly$(|\mathcal{K}|)$ + $\exp(|\mathcal{T}|)\cdot$poly$(|\mathcal{K}|)$.

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.

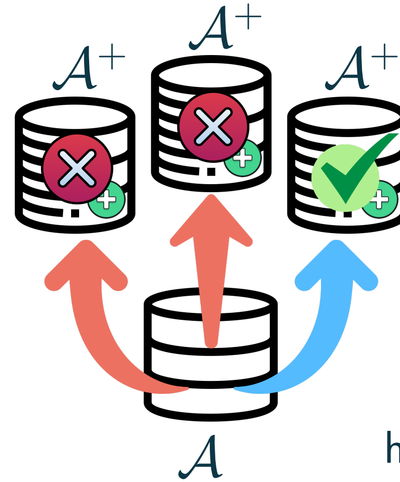


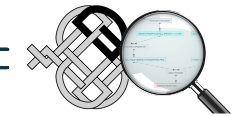
Reasoning Problems & Our Results

1. Satisfiability (consistency) problem.

> **IN**: ABox $\mathcal{A}$ (DB) + TBox $\mathcal{T}$ (Knowledge)
>
> **OUT**: Is there an extension of $\mathcal{A}$ satisfying $\mathcal{T}$?

- Decidability of full $\mathcal{ZOIQ}$ is unknown!
- Focus: max decidable fragments $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.
- For uniformity: we study quasi-forest satisfiability of $\mathcal{ZOIQ}$.



$\models$  $\mathcal{T}$

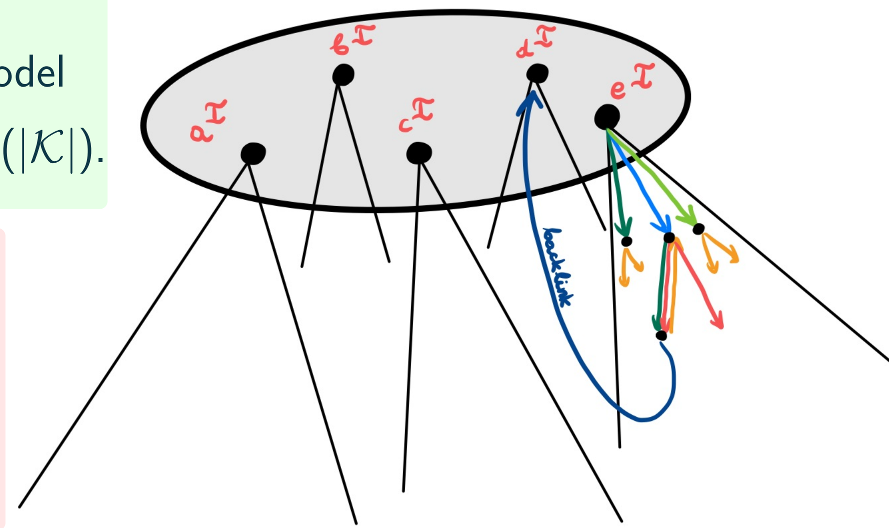
decidable

in $\exp(|\mathcal{A}| + |\mathcal{T}|)$ via

hardcore automata th.

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|)$ + nondet-poly($|\mathcal{K}|$) + $\exp(|\mathcal{T}|)\cdot$poly($|\mathcal{K}|$).



Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

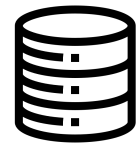
OUT: Is there an extension of $\mathcal{A}$ satisfying $\mathcal{T}$?

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

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

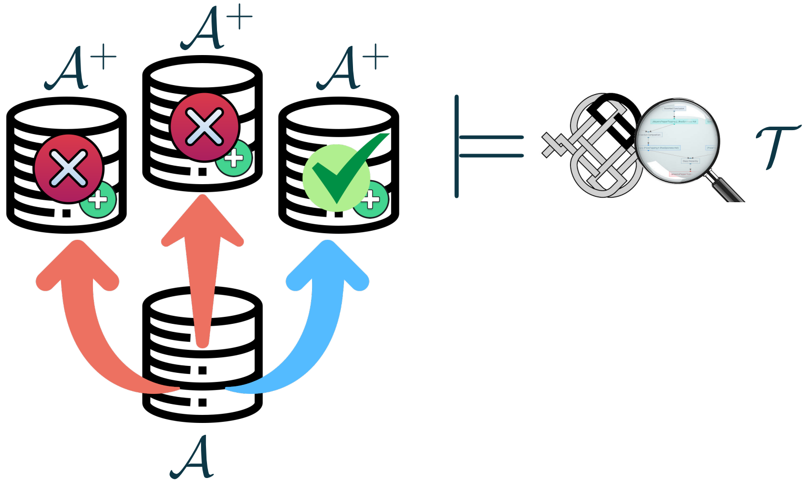


$\mathcal{A}$

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

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

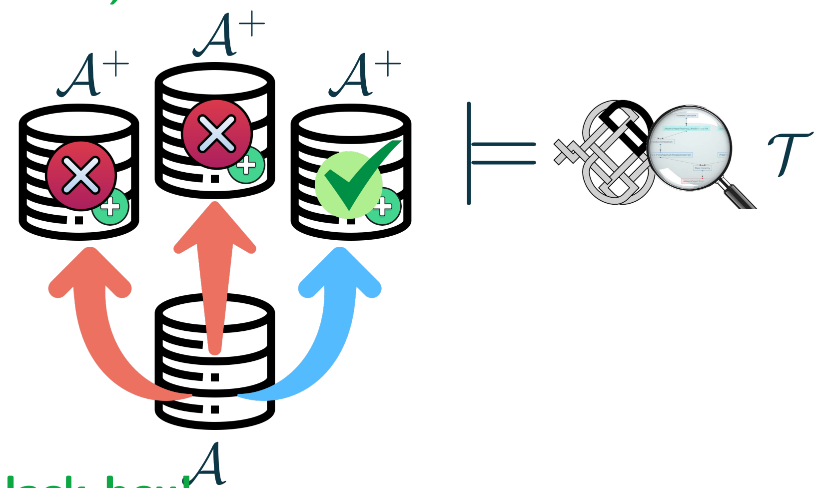


Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

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



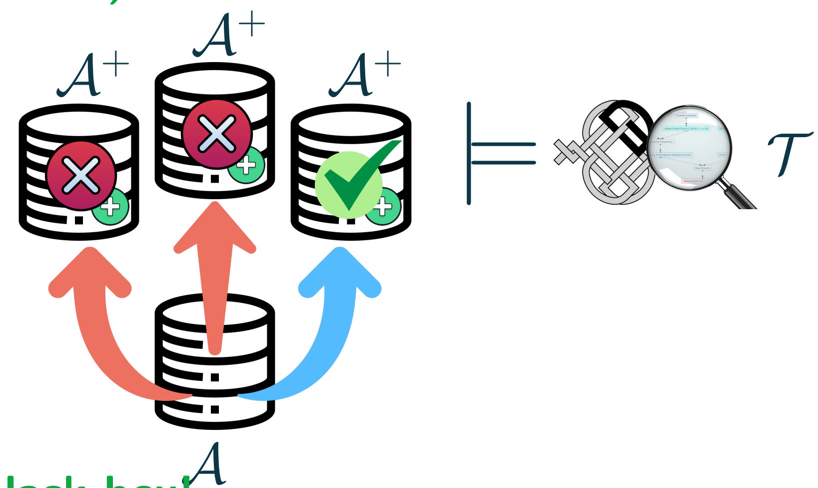
Proof plan: Use SAT as a black box!

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

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



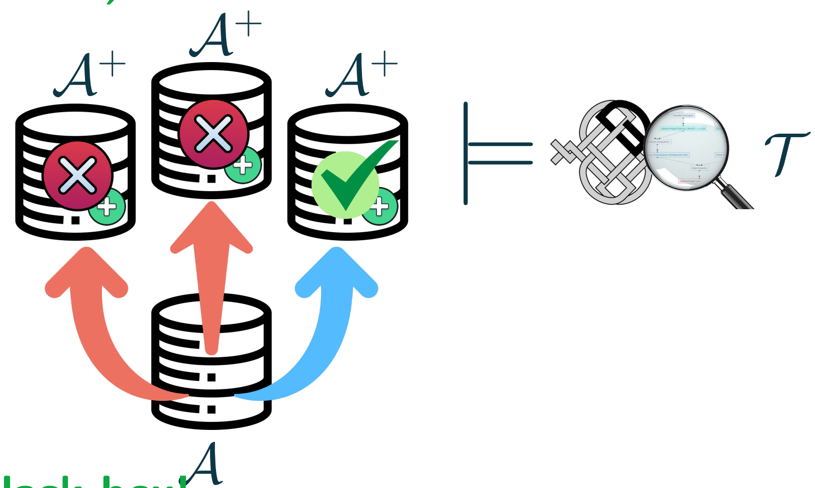
Proof plan: Use SAT as a black box!

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

IN: ABox $\mathcal{A}$ (Data)

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



Proof plan: Use SAT as a black box!

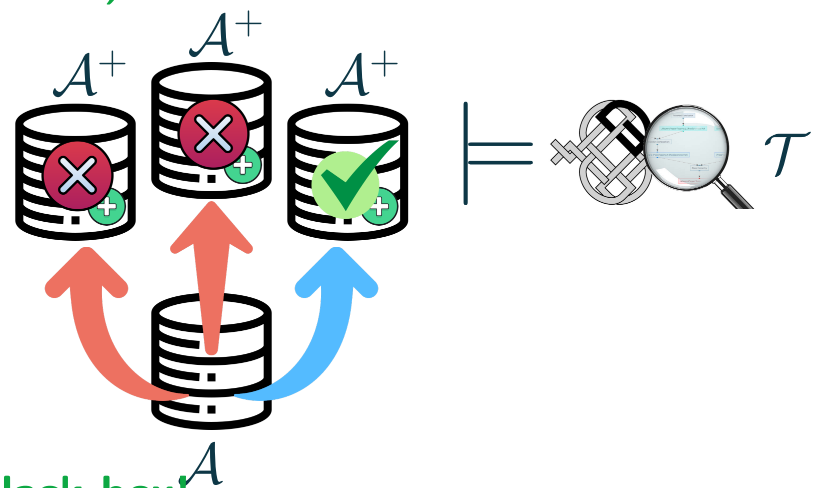
TBox: $\mathrm{A} \equiv \{\mathrm{o}\}$, $\mathrm{A} \equiv \neg\mathrm{B}$, $\mathrm{A} \equiv \mathrm{B} \sqcup \mathrm{B}'$, $\mathrm{A} \equiv \exists s.\mathsf{Self}$, $s = s'$, $\mathrm{A} \equiv (\geq n\ s).\top$, $\mathrm{A} \equiv \exists \mathbb{A}.\top$

Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

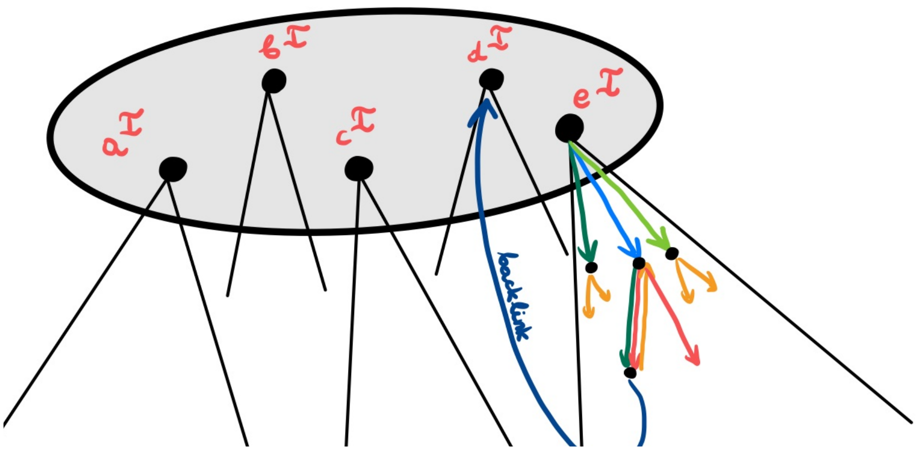
IN: ABox $\mathcal{A}$ (Data)

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



Proof plan: Use SAT as a black box!

TBox: $\mathrm{A} \equiv \{\mathrm{o}\}$, $\mathrm{A} \equiv \neg\mathrm{B}$, $\mathrm{A} \equiv \mathrm{B} \sqcup \mathrm{B}'$, $\mathrm{A} \equiv \exists s.\mathsf{Self}$, $s = s'$, $\mathrm{A} \equiv (\geq n\ s).\top$, $\mathrm{A} \equiv \exists \mathbb{A}.\top$

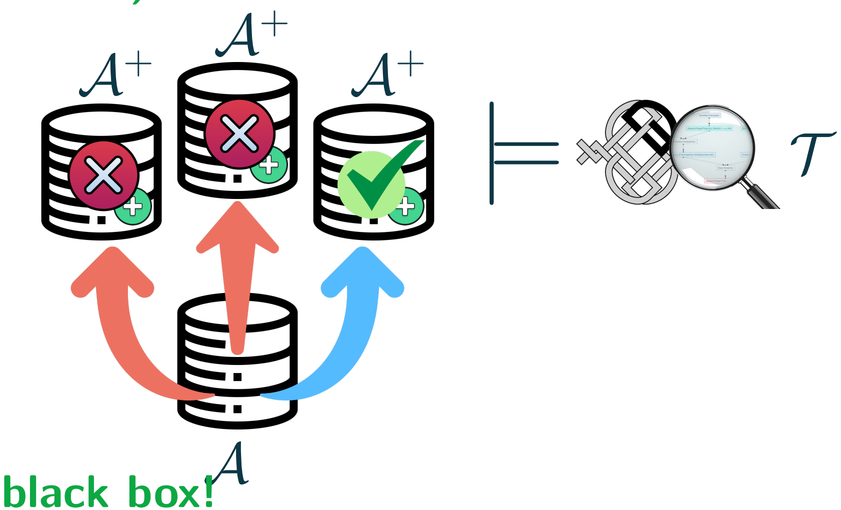


Example proofs. Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part I

Parameter: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ (Knowledge)

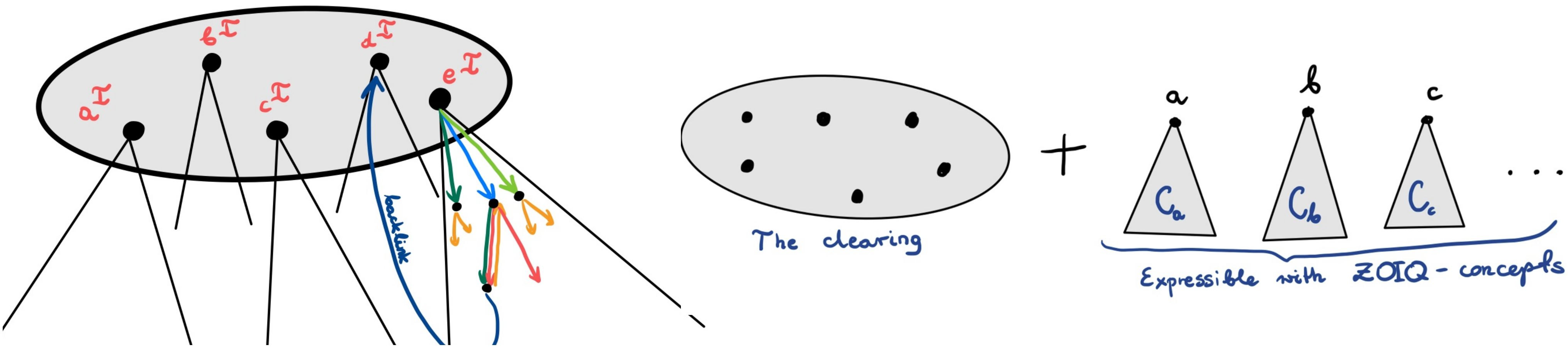
IN: ABox $\mathcal{A}$ (Data)

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



Proof plan: Use SAT as a black box!

TBox: $\mathrm{A} \equiv \{\mathrm{o}\}$, $\quad \mathrm{A} \equiv \neg\mathrm{B}$, $\quad \mathrm{A} \equiv \mathrm{B} \sqcup \mathrm{B}'$, $\quad \mathrm{A} \equiv \exists s.\mathsf{Self}$, $\quad s = s'$, $\quad \mathrm{A} \equiv (\geq n\ s).\top$, $\quad \mathrm{A} \equiv \exists \mathbb{A}.\top$



Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

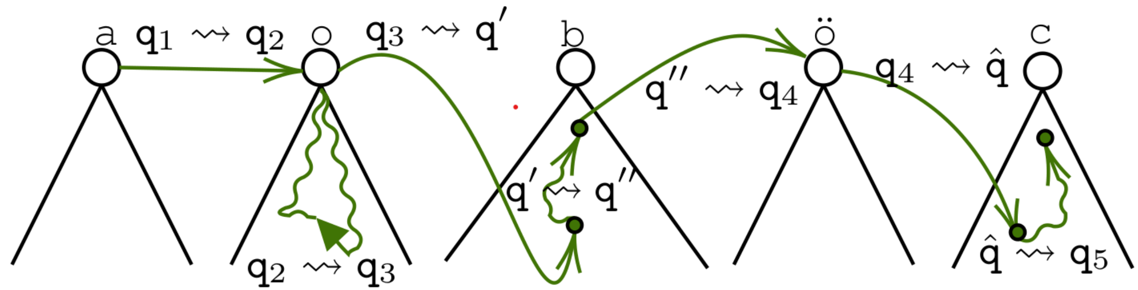
- Main issue: regular path constraints are not local.

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

- Main issue: regular path constraints are not local. Consider the following run of $\mathbb{A}$:

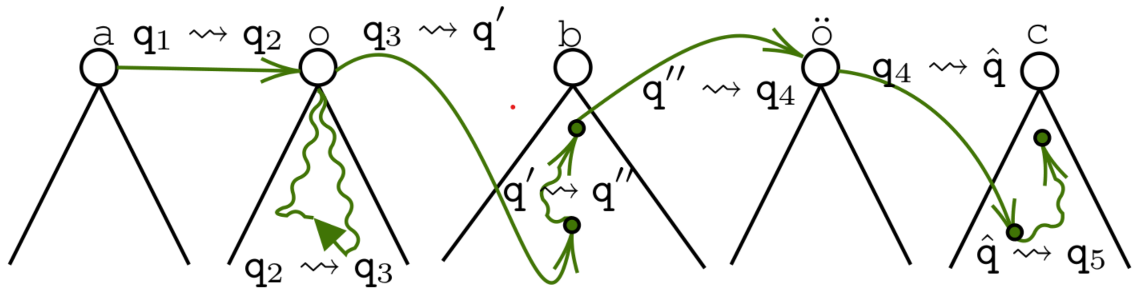
Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

- Main issue: regular path constraints are not local. Consider the following run of $\mathbb{A}$:



Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

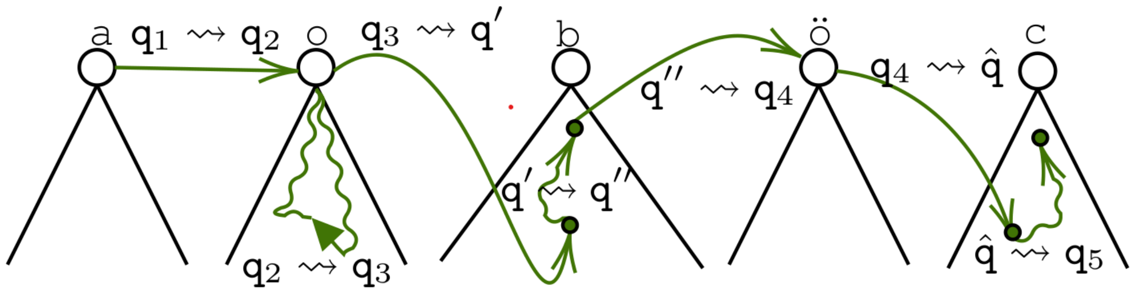
- Main issue: regular path constraints are not local.Consider the following run of $\mathbb{A}$:



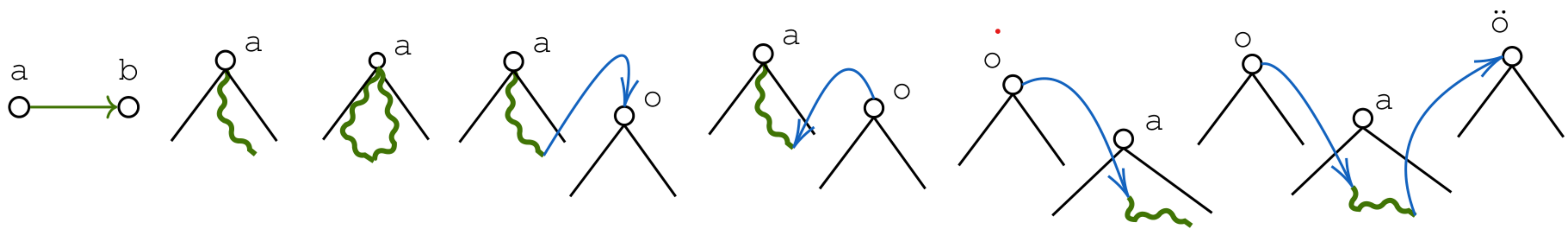
- Solution: Organize paths into interesting pieces and decorate the clearing with "shortcuts".

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

- Main issue: regular path constraints are not local.Consider the following run of $\mathbb{A}$:

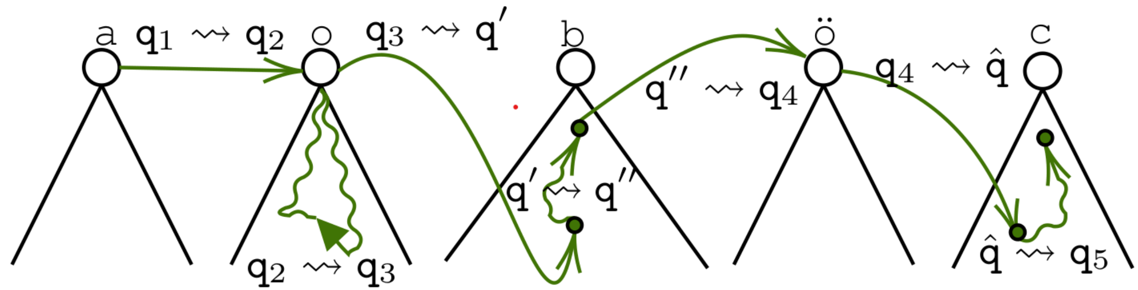


- Solution: Organize paths into interesting pieces and decorate the clearing with "shortcuts".



Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

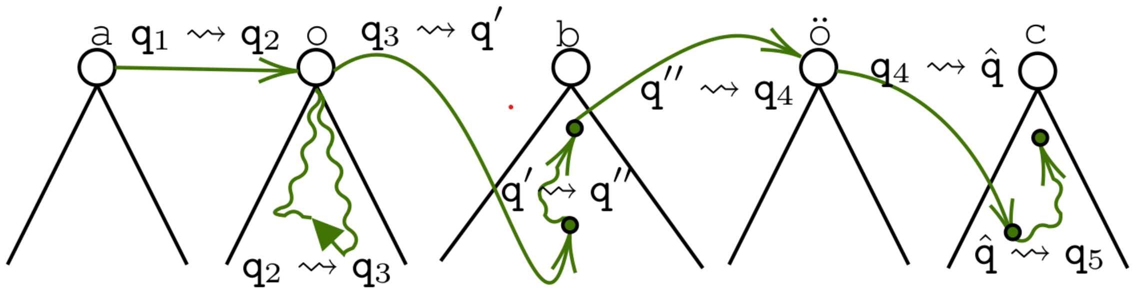
- Main issue: regular path constraints are not local. Consider the following run of $\mathbb{A}$:



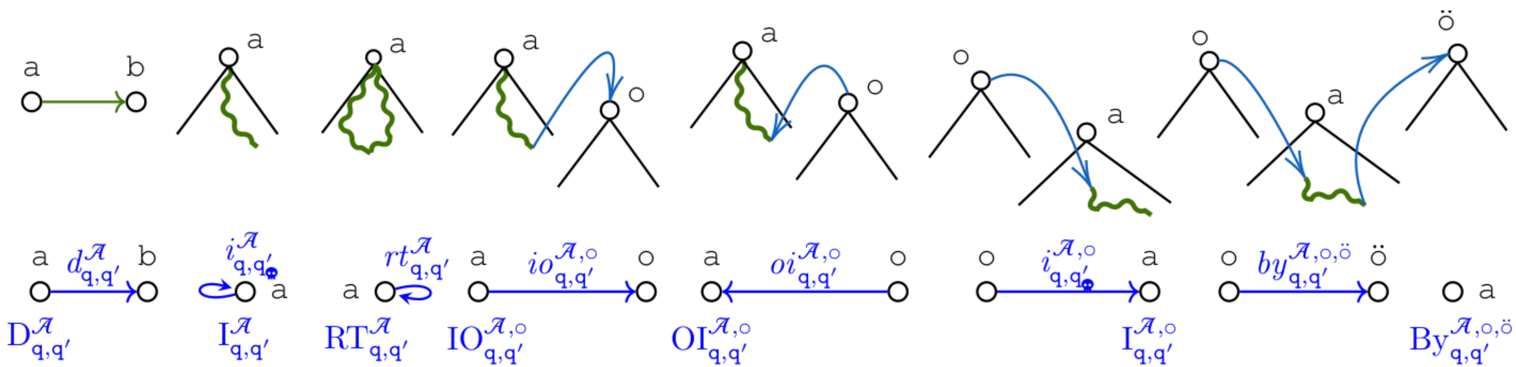
- Solution: Organize paths into interesting pieces and decorate the clearing with "shortcuts".

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

- Main issue: regular path constraints are not local. Consider the following run of $\mathbb{A}$:

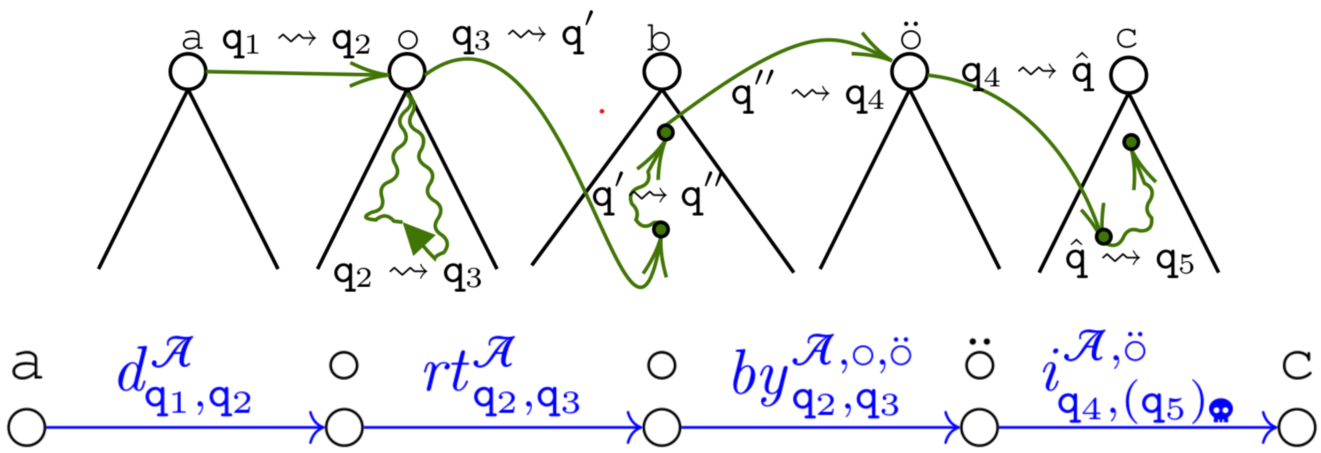


- Solution: Organize paths into interesting pieces and decorate the clearing with "shortcuts".

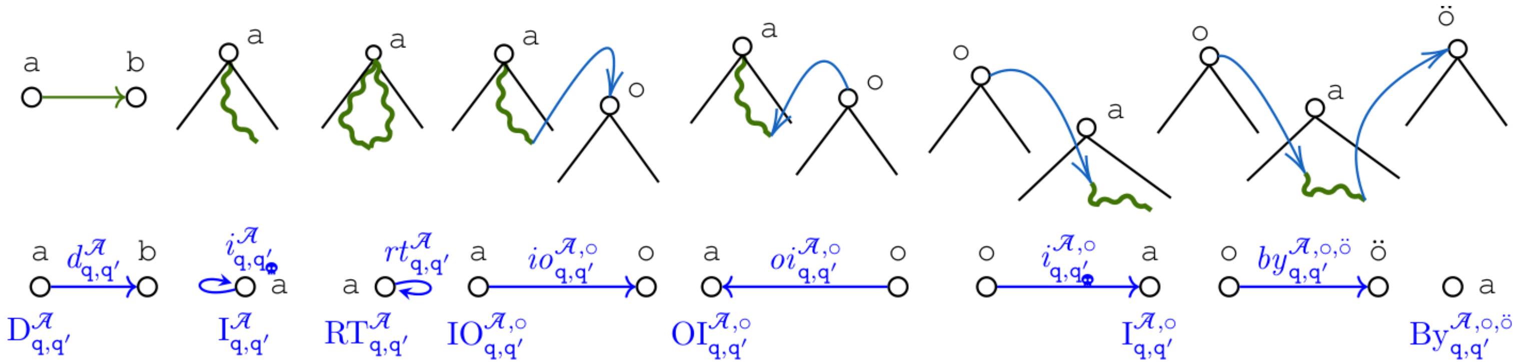


Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part II

- Main issue: regular path constraints are not local. Consider the following run of $\mathbb{A}$:



- Solution: Organize paths into interesting pieces and decorate the clearing with "shortcuts".



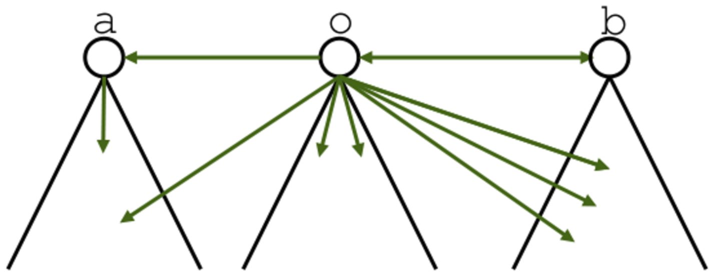
Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

- Second issue: counting is not local.

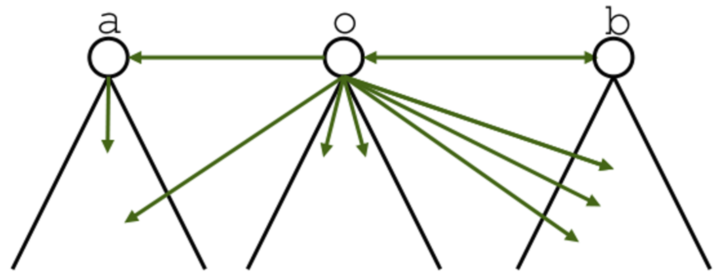
Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

- Second issue: counting is not local.



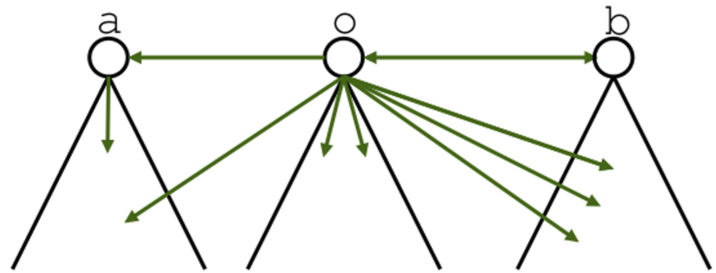
Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

- Second issue: counting is not local. Consider a concept $(\geq 2r).\top$.



Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

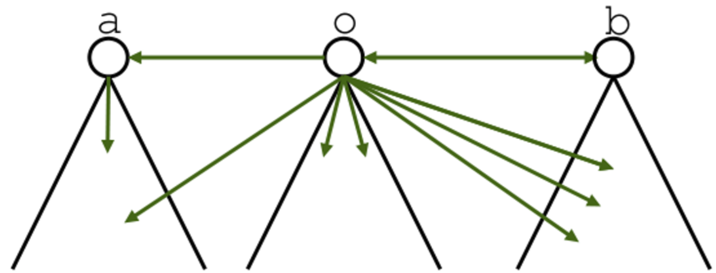
- Second issue: counting is not local. Consider a concept $(\geq 2r).\top$.



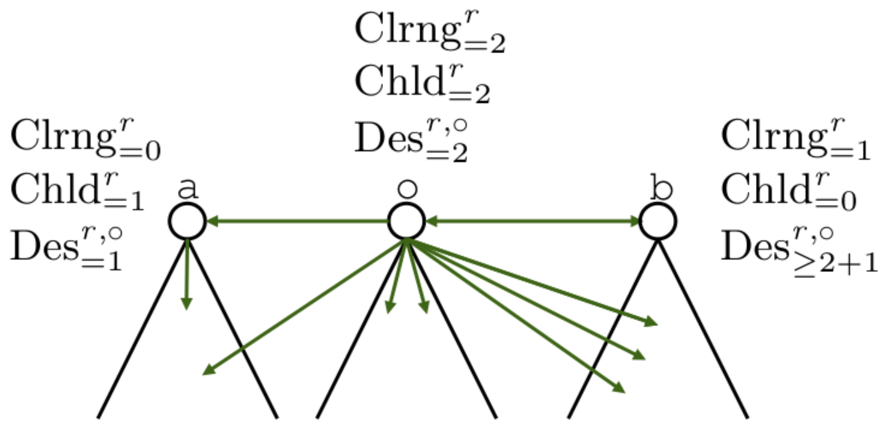
- Count the number of r-sucessors (i) in the clearing, (ii) among children, (iii) that a linked to a nominal.

Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

- Second issue: counting is not local. Consider a concept $(\geq 2r).\top$.

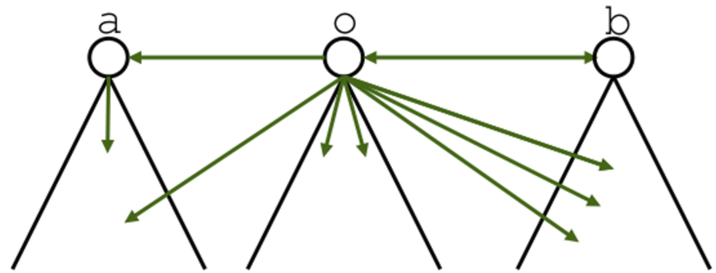


- Count the number of r-sucessors (i) in the clearing, (ii) among children, (iii) that a linked to a nominal.

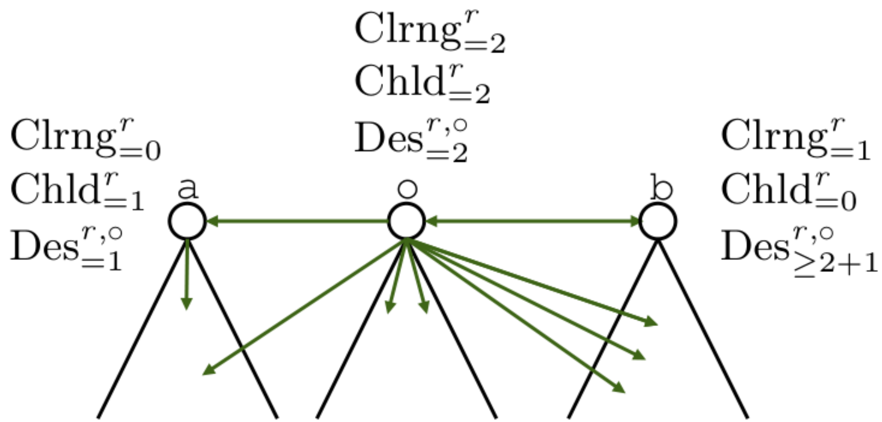


Data-Complexity of (tamed) $\mathcal{ZOIQ}$: Part III

- Second issue: counting is not local. Consider a concept $(\geq 2r).\top$.



- Count the number of r-sucessors (i) in the clearing, (ii) among children, (iii) that a linked to a nominal.



$\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ have NP-complete SAT (w.r.t. the data complexity).

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.
2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad\qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad\qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad\qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders

$\rightsquigarrow$

SAT $>$ FINSAT

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders $\rightsquigarrow$ SAT $>$ FINSAT

- $\mathcal{ZOIQ}$ with fixed-points or intersection of regular expressions are undecidable.

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders $\rightsquigarrow$ SAT $>$ FINSAT

- $\mathcal{ZOIQ}$ with fixed-points or intersection of regular expressions are undecidable.
- [Jean+Carsten+T.Zeume, KR'20] $\mathcal{ZOIQ}$ "is undecidable" $\rightsquigarrow$ incorrect proof.

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1 \; edge).\top \qquad \top \sqsubseteq (\leq 1 \; edge^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists (edge^{-})^{*}.\{\mathrm{start}\} \sqcap \exists edge^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders $\rightsquigarrow$ SAT $>$ FINSAT

- $\mathcal{ZOIQ}$ with fixed-points or intersection of regular expressions are undecidable.
- [Jean+Carsten+T.Zeume, KR'20] $\mathcal{ZOIQ}$ "is undecidable" $\rightsquigarrow$ incorrect proof.
- Partial decidability results by Jung et al.

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists (\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders $\rightsquigarrow$ SAT $>$ FINSAT

- $\mathcal{ZOIQ}$ with fixed-points or intersection of regular expressions are undecidable.
- [Jean+Carsten+T.Zeume, KR'20] $\mathcal{ZOIQ}$ "is undecidable" $\rightsquigarrow$ incorrect proof.
- Partial decidability results by Jung et al.

Close to undecidability

How about $\mathcal{ZOIQ}$? Notoriously difficult open problem!

1. Let start an end be individual names.

2. Let *edge* be a role name, interpreted as functional and backward functional relation.

$$\top \sqsubseteq (\leq 1\ \mathit{edge}).\top \qquad \top \sqsubseteq (\leq 1\ \mathit{edge}^{-}).\top$$

3. Every element can reach end and backwardly reach start.

$$\top \sqsubseteq \exists(\mathit{edge}^{-})^{*}.\{\mathrm{start}\} \sqcap \exists \mathit{edge}^{*}.\{\mathrm{start}\}$$

Defines Finite Linear Orders $\rightsquigarrow$ SAT $>$ FINSAT

- $\mathcal{ZOIQ}$ with fixed-points or intersection of regular expressions are undecidable.
- [Jean+Carsten+T.Zeume, KR'20] $\mathcal{ZOIQ}$ "is undecidable" $\rightsquigarrow$ incorrect proof.
- Partial decidability results by Jung et al.

Close to undecidability

Completely new approach is needed (in contrast to $\mathcal{ALCOIQ}$ and beyond)!

Summary

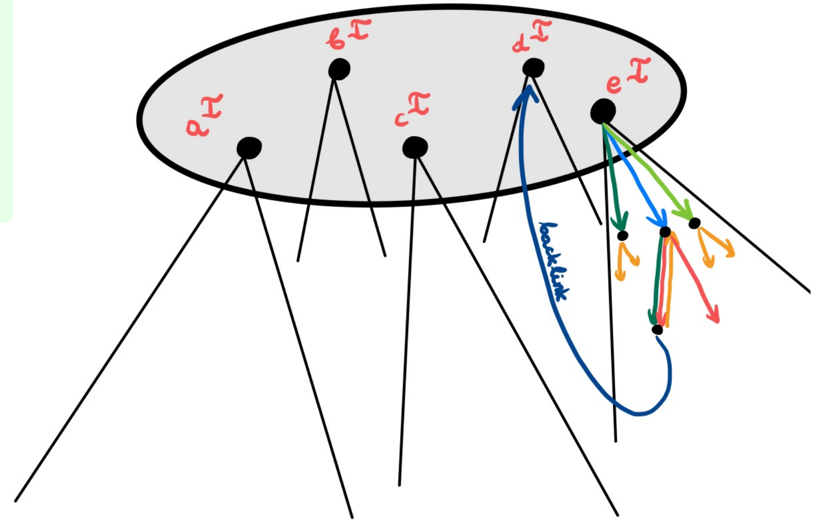
Summary

The Main Result

Summary

The Main Result

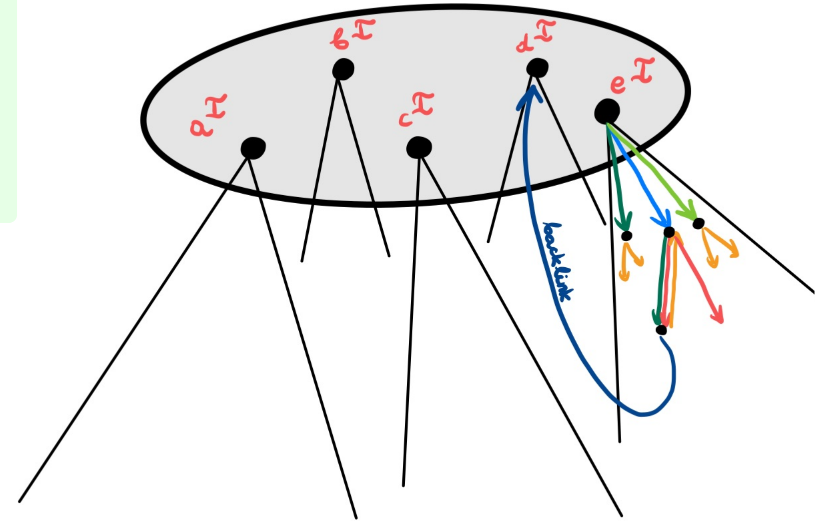
Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model



Summary

The Main Result

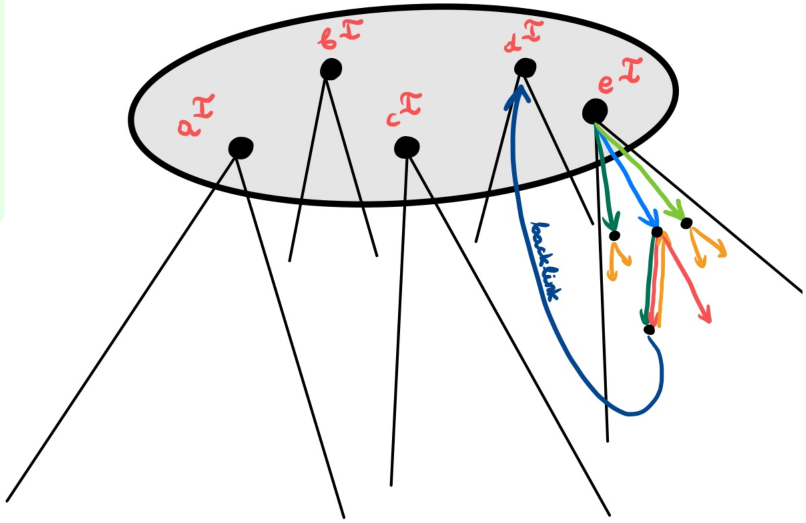
Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in



Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.

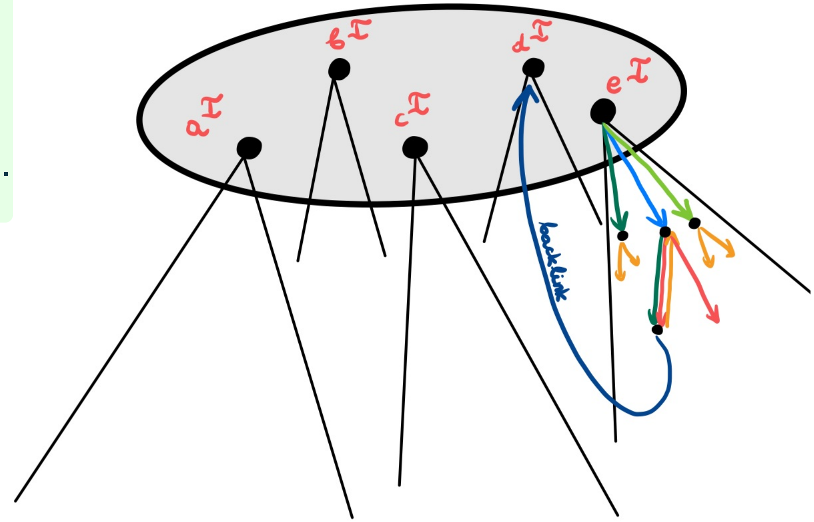


Summary

The Main Result

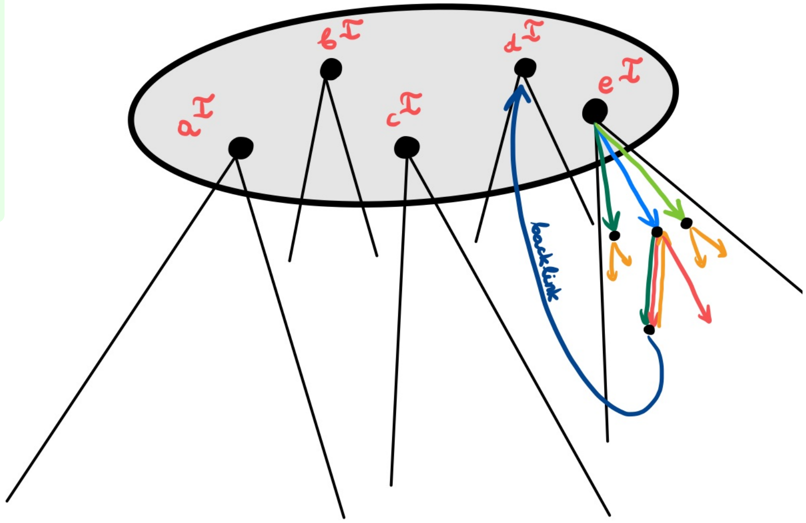
Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) +$ nondet-poly$(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot$poly$(|\mathcal{K}|)$.

Corollaries



The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) +$ nondet-poly$(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot$poly$(|\mathcal{K}|)$.



Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.

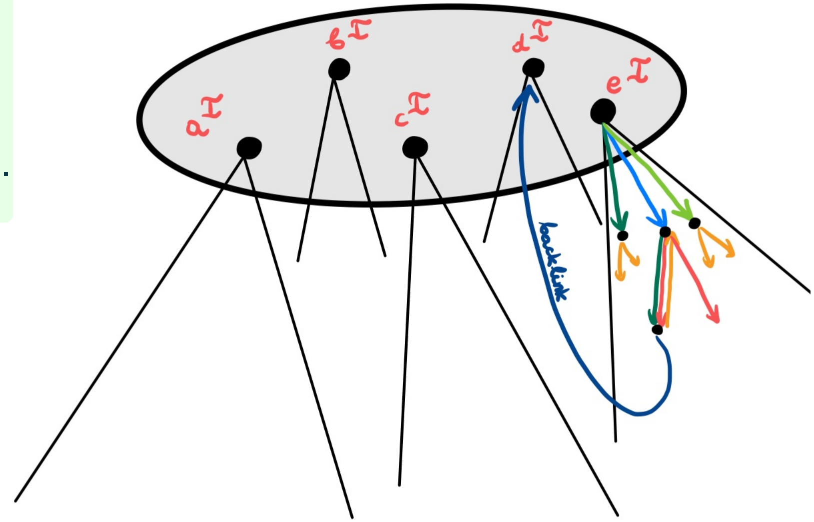
Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in exp($|\mathcal{T}|$) + nondet-poly($|\mathcal{K}|$) + exp($|\mathcal{T}|$)·poly($|\mathcal{K}|$).

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.



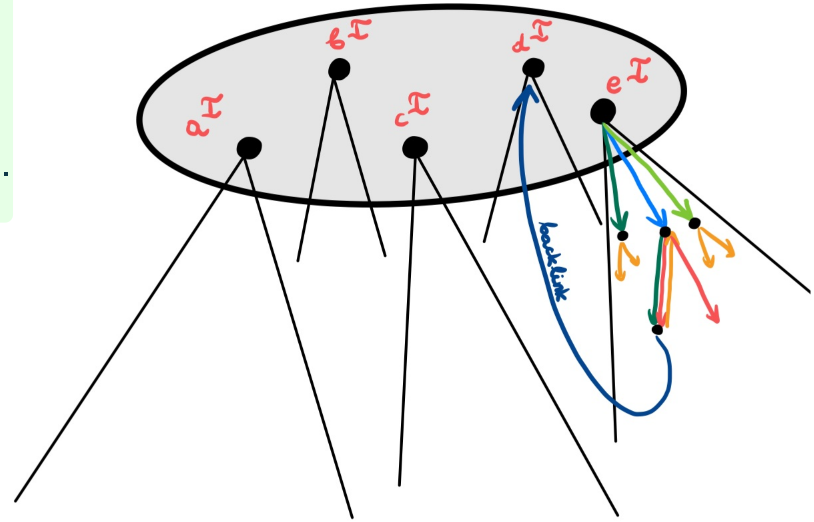
Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.



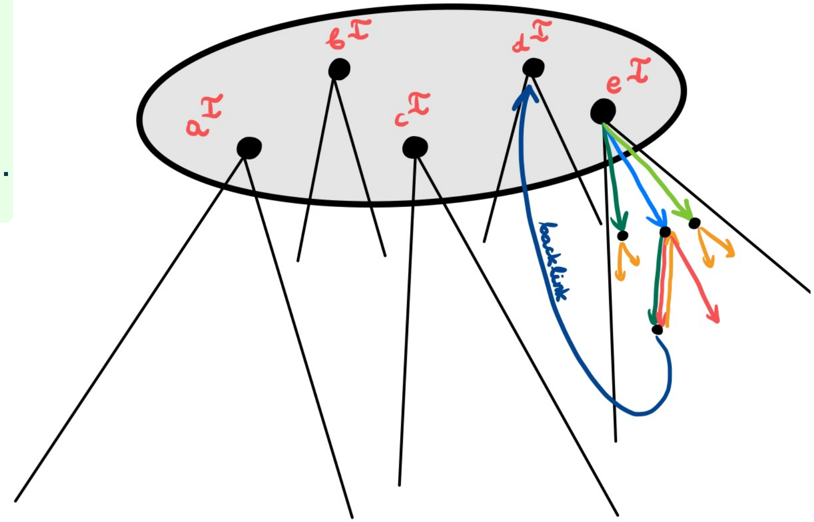
Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.



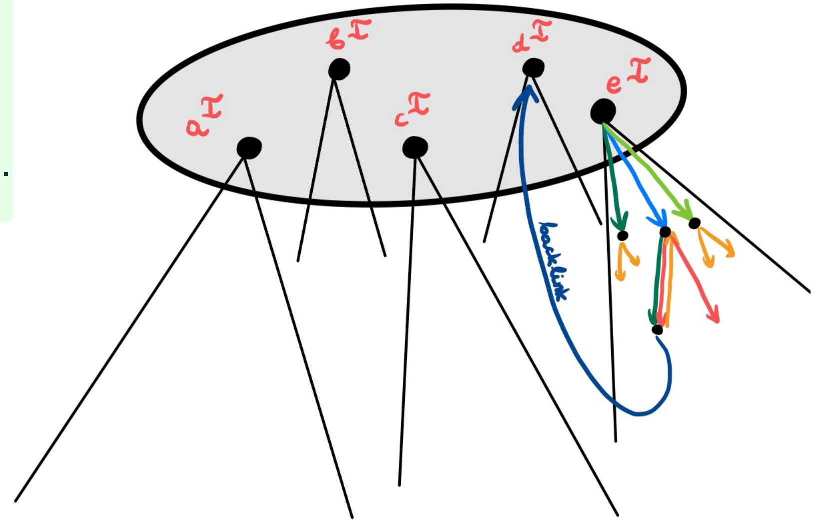
Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.



PhD Defense: Database-Inspired Reasoning Problems in DLs With Path Expressions

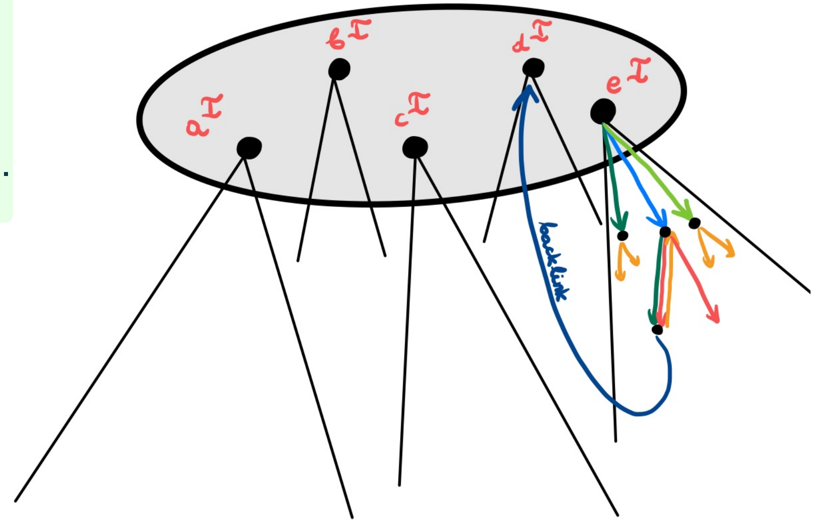
Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.

Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.



PhD Defense: Database-Inspired Reasoning Problems in DLs With Path Expressions

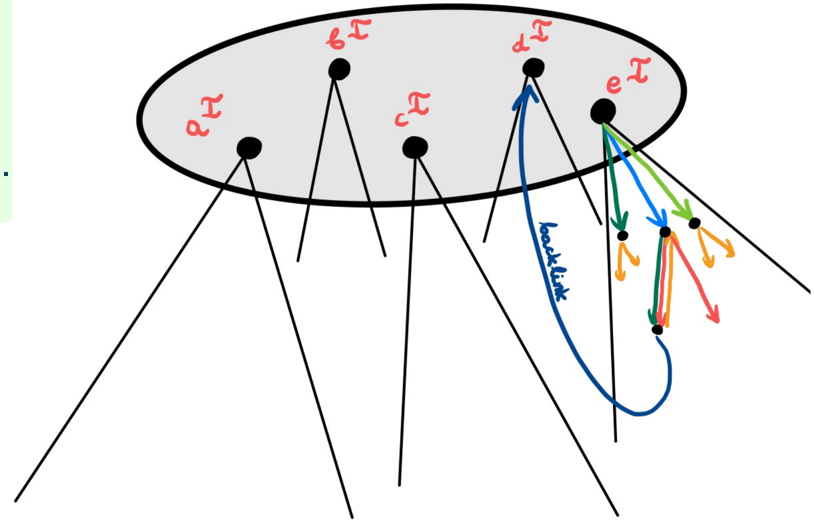
European Research Council
Established by the European Commission

Sebastian Rudolph (TU Dresden) Emanuel Kieroński (Univ. of Wrocław)

Summary

The Main Result

Checking if a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model can be done in $\exp(|\mathcal{T}|) + \text{nondet-poly}(|\mathcal{K}|) + \exp(|\mathcal{T}|) \cdot \text{poly}(|\mathcal{K}|)$.



Corollaries

- NP-c. of SAT (data-c) for $\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}$.
- coNP-c. of RPQ Ent. (data-c) for $\mathcal{Z}$- and $\mathcal{SR}$-families.
- coNEXP-c. of rooted (U)CQ Entailment for $\mathcal{ZIQ}$.

PhD Defense: Database-Inspired Reasoning Problems in DLs With Path Expressions

DeciGUT

European Research Council

Established by the European Commission

Tue 25.06.24 13:30 ICCL TU Dresden

Sebastian Rudolph (TU Dresden)

Emanuel Kieroński (Univ. of Wrocław)