A Framework for Reasoning about Dynamic Axioms

in Description Logics

Bartosz Bednarczyk, Stéphane Demri, Alessio Mansutti

TU Dresden & University of Wrocław, CNRS & ENS Paris-Saclay

TECHNISCHE UNIVERSITÄT DRESDEN

Uniwersytet Wrocławski

DeciGUT

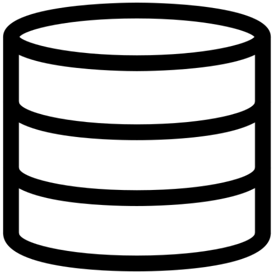
European Research Council

Established by the European Commission

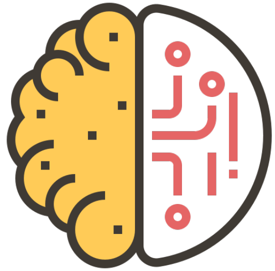
Running example: basketball teams and (possibly injured) players

Running example: basketball teams and (possibly injured) players

Database (ABox)

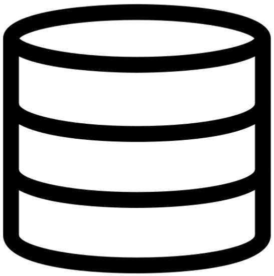


Knowledge (TBox)



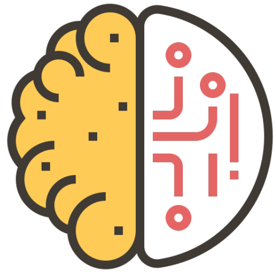
Running example: basketball teams and (possibly injured) players

Database (ABox)



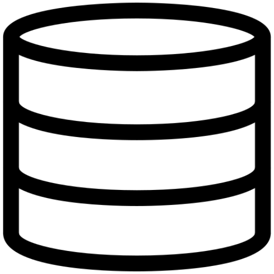
isDrafted($\texttt{Zion}, \texttt{Pelicans}$)

Knowledge (TBox)

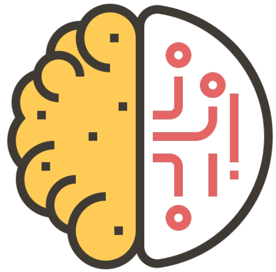


Running example: basketball teams and (possibly injured) players

Database (ABox)



Knowledge (TBox)

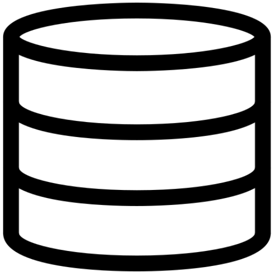


$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Running example: basketball teams and (possibly injured) players

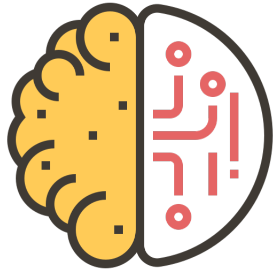
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

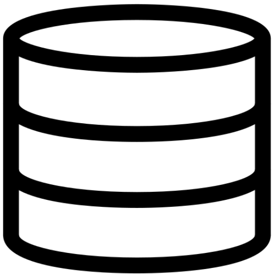
Knowledge (TBox)



$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$

Running example: basketball teams and (possibly injured) players

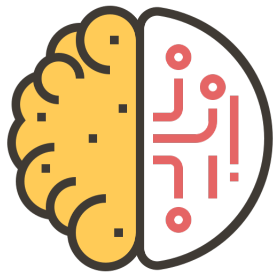
Database (ABox)



$\mathit{isDrafted}(\texttt{Zion}, \texttt{Pelicans})$

$\vdots$

Knowledge (TBox)

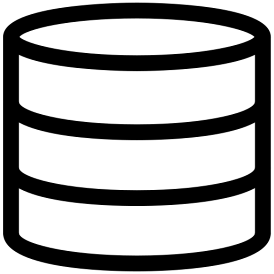


$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$

$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$

Running example: basketball teams and (possibly injured) players

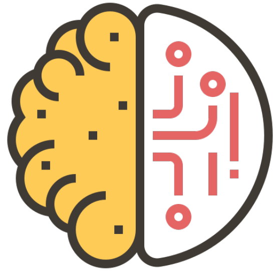
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Knowledge (TBox)



$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$

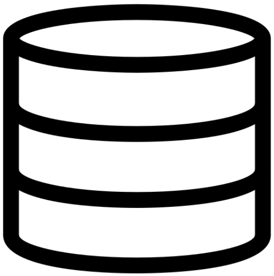
$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$

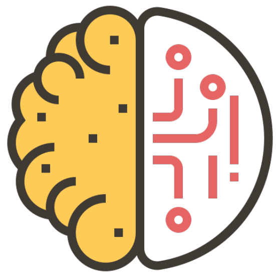
$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$

Running example: basketball teams and (possibly injured) players

Database (ABox)

Knowledge (TBox)





$$isDrafted(\texttt{Zion}, \texttt{Pelicans})$$

$$\vdots$$

$$\exists hasInjury.Injury \sqsubseteq Player$$

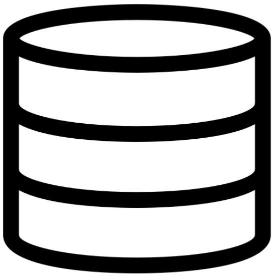
$$\exists isDrafted.Team \sqsubseteq Player$$

$$\exists hasInjury.Injury \sqcap \exists isDrafted.Team \sqsubseteq \bot$$

It essentially states that no injured player can be drafted by a team.

Running example: basketball teams and (possibly injured) players

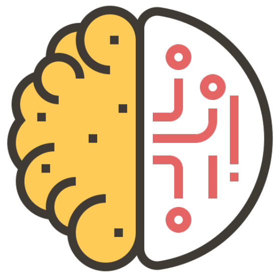
Database (ABox)



$isDrafted(\texttt{Zion}, \texttt{Pelicans})$

$\vdots$

Knowledge (TBox)



$$\exists hasInjury.Injury \sqsubseteq Player$$

$$\exists isDrafted.Team \sqsubseteq Player$$

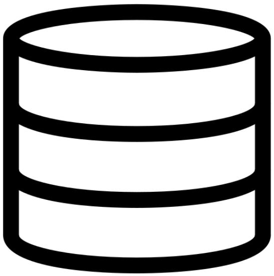
$$\exists hasInjury.Injury \sqcap \exists isDrafted.Team \sqsubseteq \bot$$

It essentially states that no injured player can be drafted by a team.

Usually knowledge bases are static.

Running example: basketball teams and (possibly injured) players

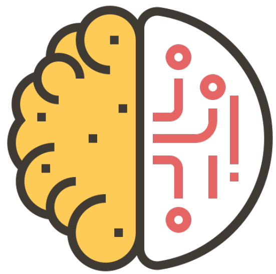
Database (ABox)



$$isDrafted(\texttt{Zion}, \texttt{Pelicans})$$

$$\vdots$$

Knowledge (TBox)



$$\exists hasInjury.Injury \sqsubseteq Player$$

$$\exists isDrafted.Team \sqsubseteq Player$$

$$\exists hasInjury.Injury \sqcap \exists isDrafted.Team \sqsubseteq \bot$$

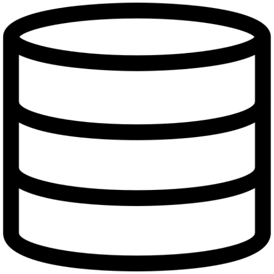
It essentially states that no injured player can be drafted by a team.

Usually knowledge bases are static.

How to update them? And how to reason about updates?

Running example: basketball teams and (possibly injured) players

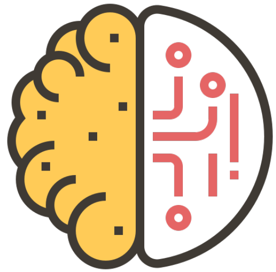
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Knowledge (TBox)



$$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$$

It essentially states that no injured player can be drafted by a team.

Usually knowledge bases are static.

How to update them? And how to reason about updates?

Description logics and updates

Description logics and updates

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Focus was mostly on updating ABoxes!

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Focus was mostly on updating ABoxes!

What if interpretations changes too?

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Focus was mostly on updating ABoxes!

What if interpretations changes too?

Our main goal: how to specify the evolution of ABoxes and TBoxes

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Focus was mostly on updating ABoxes!

What if interpretations changes too?

Our main goal: how to specify the evolution of ABoxes and TBoxes

when the current interpretation is updated?

Description logics and updates

- Liu et al. Updating Description Logic ABoxes, KR'06
- De Giacomo et al. On the update of DL ontologies at the instance level, AAAI'06
- Drescher et al. Putting ABox updates into action, FroCoS'09

Focus was mostly on updating ABoxes!

What if interpretations changes too?

Our main goal: how to specify the evolution of ABoxes and TBoxes

when the current interpretation is updated?

We propose a new framework based on separation logics!

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

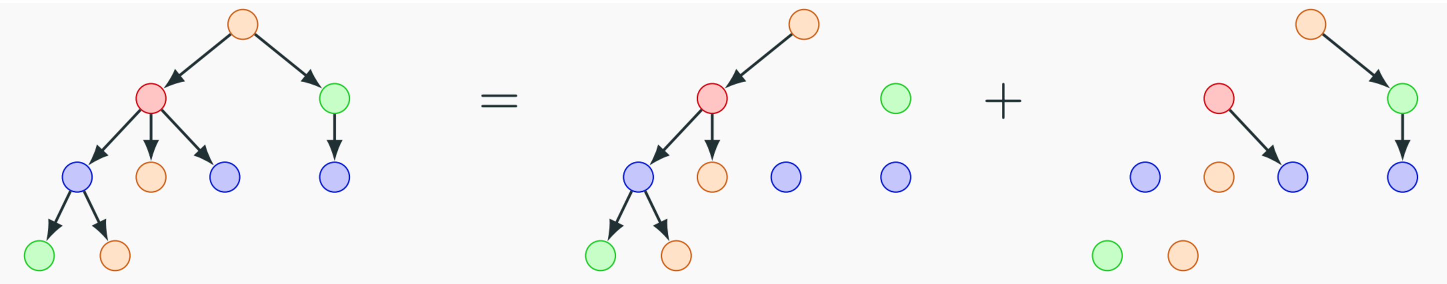
In our scenario $\oplus$ decomposes the roles of interpretations as follows:

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

In our scenario $\oplus$ decomposes the roles of interpretations as follows:

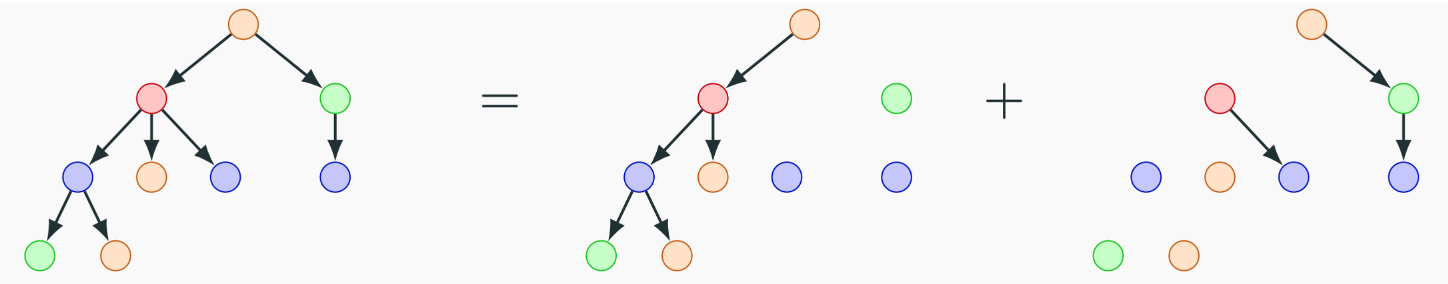


A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

In our scenario $\oplus$ decomposes the roles of interpretations as follows:



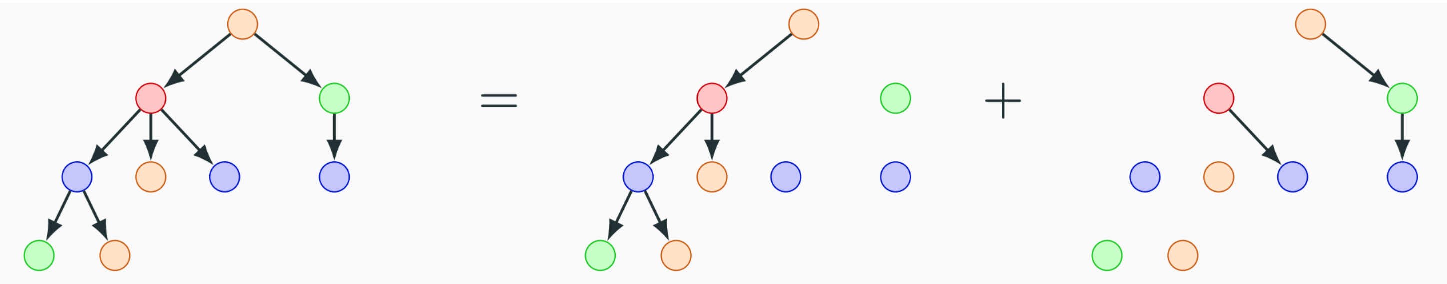
Now we are ready to introduce Dynamic Axioms:

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

In our scenario $\oplus$ decomposes the roles of interpretations as follows:



Now we are ready to introduce Dynamic Axioms:

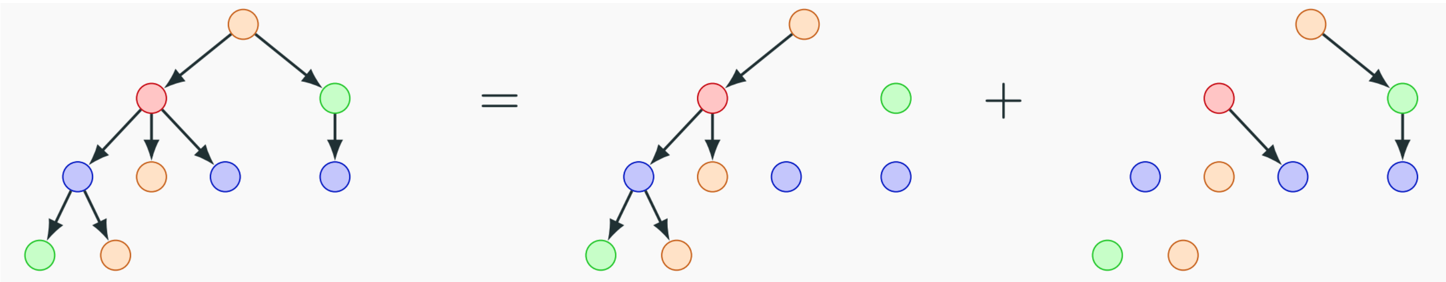
$$\mathbb{U}, \mathbb{V} ::= \underbrace{\top \mid C(\mathrm{a}) \mid r(\mathrm{a}, \mathrm{b}) \mid C \sqsubseteq D}_{\text{standard DL axioms}} \mid \mathbb{U} * \mathbb{V} \mid \mathbb{U} \multimap\!\circledast\, \mathbb{V} \quad \underbrace{\mid \neg \mathbb{U} \mid \mathbb{U} \sqcap \mathbb{V}}_{\text{boolean operations on axioms}}$$

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

In our scenario $\oplus$ decomposes the roles of interpretations as follows:



Now we are ready to introduce Dynamic Axioms:

$$\mathbb{U}, \mathbb{V} ::= \underbrace{\top \mid C(\mathrm{a}) \mid r(\mathrm{a}, \mathrm{b}) \mid C \sqsubseteq D}_{\text{standard DL axioms}} \mid \mathbb{U} * \mathbb{V} \mid \mathbb{U} \mathrel{-\!\!\circledast} \mathbb{V} \quad \underbrace{\mid \neg\mathbb{U} \mid \mathbb{U} \sqcap \mathbb{V}}_{\text{boolean operations on axioms}}$$

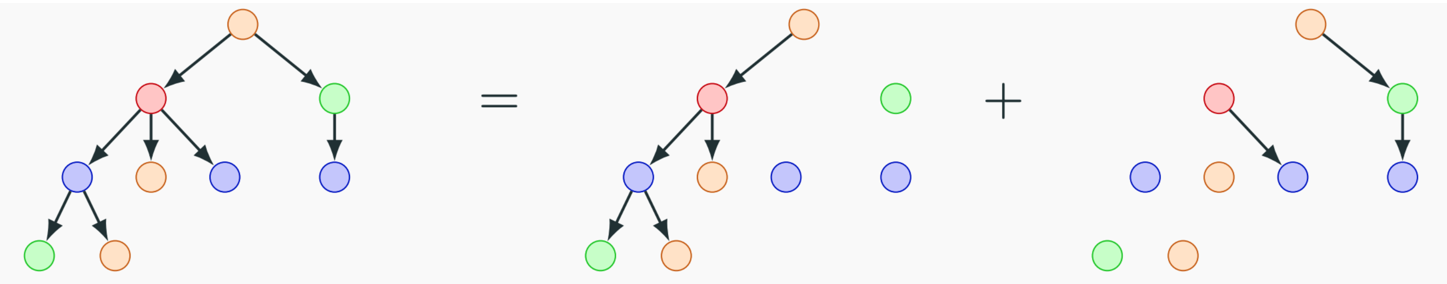
- $\mathcal{I} \models \mathbb{U}_1 * \mathbb{U}_2$ iff there are $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I} = \mathcal{I}_1 \oplus \mathcal{I}_2$ s.t. $\mathcal{I}_i \models \mathbb{U}_i$

A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$

Let $\mathbb{I}$ be the class of all interpretations.

We take a composition operator $\oplus : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ to be any AC operator.

In our scenario $\oplus$ decomposes the roles of interpretations as follows:



Now we are ready to introduce Dynamic Axioms:

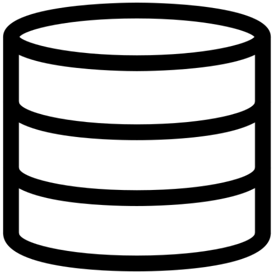
$$\mathbb{U}, \mathbb{V} ::= \underbrace{\top \mid C(\mathrm{a}) \mid r(\mathrm{a}, \mathrm{b}) \mid C \sqsubseteq D}_{\text{standard DL axioms}} \mid \mathbb{U} * \mathbb{V} \mid \mathbb{U} \mathrel{-\!\!\circledast} \mathbb{V} \quad \underbrace{\mid \neg\mathbb{U} \mid \mathbb{U} \sqcap \mathbb{V}}_{\text{boolean operations on axioms}}$$

- $\mathcal{I} \models \mathbb{U}_1 * \mathbb{U}_2$ iff there are $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I} = \mathcal{I}_1 \oplus \mathcal{I}_2$ s.t. $\mathcal{I}_i \models \mathbb{U}_i$
- $\mathcal{I} \models \mathbb{U}_1 \mathrel{-\!\!\circledast} \mathbb{U}_2$ iff there is $\mathcal{J}$ such that $\mathcal{J} \models \mathbb{U}_1$ and $\mathcal{I} \oplus \mathcal{J} \models \mathbb{U}_2$

Running example: recall $\mathcal{K}$

Running example: recall $\mathcal{K}$

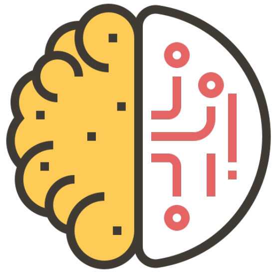
Database (ABox)



$isDrafted(\texttt{Zion}, \texttt{Pelicans})$

$\vdots$

Knowledge (TBox)



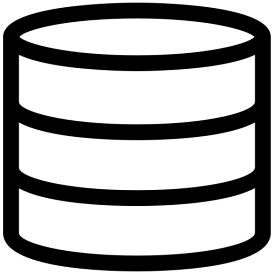
$\exists hasInjury.Injury \sqsubseteq Player$

$\exists isDrafted.Team \sqsubseteq Player$

$\exists hasInjury.Injury \sqcap \exists isDrafted.Team \sqsubseteq \bot$

Running example: recall $\mathcal{K}$

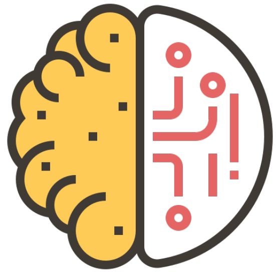
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Knowledge (TBox)



$$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$$

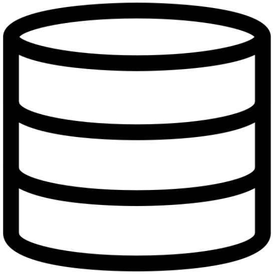
$$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$$

Consider the dynamic axiom:

Running example: recall $\mathcal{K}$

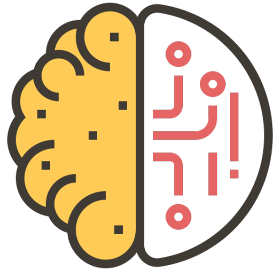
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Knowledge (TBox)



$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$

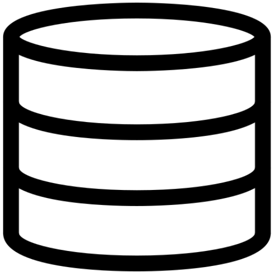
$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$

$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$

Consider the dynamic axiom: $\mathbb{U} = \top * (\top \mathbin{-\!\!\circledast} \mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans}))$

Running example: recall $\mathcal{K}$

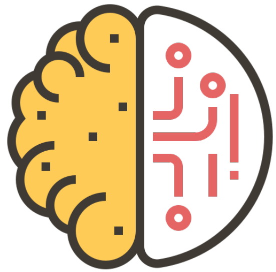
Database (ABox)



$\mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans})$

$\vdots$

Knowledge (TBox)



$$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$$

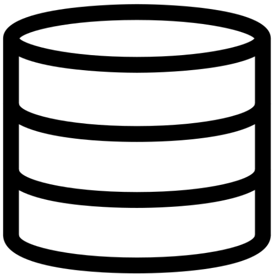
$$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$$

Consider the dynamic axiom: $\mathbb{U} = \top * (\top \mathbin{-\!\!\circledast} \mathit{isDrafted}(\mathtt{Zion}, \mathtt{Pelicans}))$

Running example: recall $\mathcal{K}$

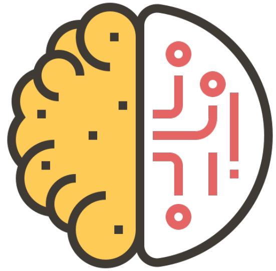
Database (ABox)



$\mathit{isDrafted}(\texttt{Zion}, \texttt{Pelicans})$

$\vdots$

Knowledge (TBox)



$$\exists \mathit{hasInjury}.\mathit{Injury} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \mathit{Player}$$

$$\exists \mathit{hasInjury}.\mathit{Injury} \sqcap \exists \mathit{isDrafted}.\mathit{Team} \sqsubseteq \bot$$

Consider the dynamic axiom: $\mathbb{U} = \top * (\top \mathrel{-\!\circledast} \mathit{isDrafted}(\texttt{Zion}, \texttt{Pelicans}))$

$\mathcal{K} \cup \mathbb{U}$ is satisfiable iff there is an evolution where $\texttt{Zion}$ is drafted by $\texttt{Pelicans}$.

Our results

Our results

- In this work we focused on the consistency problem only

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

pos-$\mathcal{EL}$	*pos*-$\mathcal{ALC}$	$\mathcal{EL}$ $\mathcal{ALC}$

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

pos-$\mathcal{EL}$	pos-$\mathcal{ALC}$	$\mathcal{EL}$
in PTIME proof system		$\mathcal{ALC}$

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

pos-$\mathcal{EL}$	*pos*-$\mathcal{ALC}$	$\mathcal{EL}$ $\mathcal{ALC}$
in PTIME proof system	EXPTIME-compl translation to $\mathcal{ALCO}b$	

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

pos-$\mathcal{EL}$	pos-$\mathcal{ALC}$	$\mathcal{EL}$ $\mathcal{ALC}$
in PTIME proof system	EXPTIME-compl translation to $\mathcal{ALCO}b$	UNDECIDABLE reduction via $\mathcal{ALC}+$ $r_1 \circ r_2 \circ \ldots \circ r_n \sqsubseteq s$

Our results

- In this work we focused on the consistency problem only
- We consider two logics: $\mathcal{ALC}$ and $\mathcal{EL}$
- We distinguish the cases of dynamic axioms (DAs) and negation-free DA

pos-$\mathcal{EL}$	pos-$\mathcal{ALC}$	$\mathcal{EL}$ $\mathcal{ALC}$
in PTIME proof system	EXPTIME-compl translation to $\mathcal{ALCO}b$	UNDECIDABLE reduction via $\mathcal{ALC}+$ $r_1 \circ r_2 \circ \ldots \circ r_n \sqsubseteq s$

Check the paper for more details!