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
Running example: basketball teams and (possibly injured) players

- Database (ABox)
- Knowledge (TBox)

Icons downloaded from icon-icons.com by

⃣ Rena Xiao and ⃣ Eucalyp Studio (both under CC BY 4.0). No changes have been made.

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)

Knowledge (TBox)

It essentially states that no injured player can be drafted by a team.
Running example: basketball teams and (possibly injured) players

Database (ABox)

Knowledge (TBox)

isDrafted(Zion, Pelicans)

It essentially states that no injured player can be drafted by a team.
Running example: basketball teams and (possibly injured) players

Database (ABox)

Knowledge (TBox)

\[ isDrafted(Zion, Pelicans) : \]

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)

```
isDrafted(Zion, Pelicans)
```

Knowledge (TBox)

```
∃ hasInjury. Injury ⊑ Player
```

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)

Knowledge (TBox)

\[ isDrafted(Zion, Pelicans) \]

\[ \exists hasInjury. Injury \sqsubseteq Player \]

\[ \exists isDrafted. Team \sqsubseteq Player \]

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?

B. Bednarczyk, S. Demri, A. Mansutti

A Framework for Reasoning about Dynamic Axioms in DLs
Running example: basketball teams and (possibly injured) players

Database (ABox)

\[
isDrafted(Zion, Pelicans) \quad : \quad \exists hasInjury . Injury \sqsubseteq Player
\]

Knowledge (TBox)

\[
\exists isDrafted . Team \sqsubseteq Player
\]

\[
\exists hasInjury . Injury \sqcap \exists isDrafted . Team \sqsubseteq ⊥
\]
Running example: basketball teams and (possibly injured) players

Database (ABox)

Knowledge (TBox)

\[ isDrafted(Zion, Pelicans) : \]

\[ \exists hasInjury. Injury \sqsubseteq Player \]

\[ \exists isDrafted. Team \sqsubseteq Player \]

\[ \exists hasInjury. Injury \sqcap \exists isDrafted. Team \sqsubseteq \bot \]

It essentially states that no injured player can be drafted by a team.
Running example: basketball teams and (possibly injured) players

Database (ABox)

\[ \text{isDrafted}(Zion, \text{Pelicans}) \]

Knowledge (TBox)

\[ \exists \text{hasInjury}. \text{Injury} \sqsubseteq \text{Player} \]
\[ \exists \text{isDrafted}. \text{Team} \sqsubseteq \text{Player} \]
\[ \exists \text{hasInjury}. \text{Injury} \sqcap \exists \text{isDrafted}. \text{Team} \sqsubseteq \bot \]

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)

Knowledge (TBox)

\[ \exists \text{isDrafted} . (\text{Team} \sqsubseteq \text{Player}) \]

\[ \exists \text{hasInjury} . (\text{Injury} \sqsubseteq \text{Player}) \]

\[ \exists \text{isDrafted} . (\text{Team} \sqsubseteq \bot) \]

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)

Knowledge (TBox)

\( \text{isDrafted}(Zion, \text{Pelicans}) \)

\( \exists \text{hasInjury} \cdot \text{Injury} \sqsubseteq \text{Player} \)

\( \exists \text{isDrafted} \cdot \text{Team} \sqsubseteq \text{Player} \)

\( \exists \text{hasInjury} \cdot \text{Injury} \sqcap \exists \text{isDrafted} \cdot \text{Team} \sqsubseteq \bot \)

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

Focus was mostly on updating ABoxes!

What if interpretations change 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!
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!
Description logics and updates

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

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!
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

Focus was mostly on updating ABoxes!

What if interpretations change 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!
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 $\oplus : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$
A partial composition operator $\oplus : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}$

Let $\mathcal{I}$ be the class of all interpretations.
A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$

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

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

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

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

In our scenario $\oplus$ decomposes the roles of interpretations as follows:
A partial composition operator $\oplus : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$

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

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

In our scenario $\oplus$ decomposes the roles of interpretations as follows:
A partial composition operator $\oplus : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}$

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

We take a composition operator $\oplus : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}$ 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 $\oplus : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$

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

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

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

Now we are ready to introduce Dynamic Axioms:

$\mathbb{U}, \mathbb{V} ::= \top | C(a) | r(a, b) | C \sqsubseteq D | \mathbb{U} \ast \mathbb{V} | \mathbb{U} \oplus \mathbb{V} | \neg \mathbb{U} | \mathbb{U} \cap \mathbb{V}$

- standard DL axioms
- boolean operations on axioms
A partial composition operator $\oplus : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}$

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

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

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

Now we are ready to introduce Dynamic Axioms:

\[ U, V ::= \top \mid C(a) \mid r(a, b) \mid C \sqsubseteq D \mid U \ast V \mid U \ominus V \mid \neg U \mid U \cap V \]

- standard DL axioms
- boolean operations on axioms

- $\mathcal{I} \models U_1 \ast U_2$ iff there are $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I} = \mathcal{I}_1 \oplus \mathcal{I}_2$ s.t. $\mathcal{I}_i \models U_i$
A partial composition operator $\oplus : \mathcal{I} \times \mathcal{I} \to \mathcal{I}$

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

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

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

Now we are ready to introduce Dynamic Axioms:

$$\begin{align*}
\mathcal{U}, \mathcal{V} &::= \top \mid C(a) \mid r(a, b) \mid C \sqsubseteq D \mid \mathcal{U} * \mathcal{V} \mid \mathcal{U} \ominus \mathcal{V} \mid \neg \mathcal{U} \mid \mathcal{U} \cap \mathcal{V} \\
\text{standard DL axioms} &\quad \text{boolean operations on axioms}
\end{align*}$$

- $\mathcal{I} \models \mathcal{U} * \mathcal{U}_2$ iff there are $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I} = \mathcal{I}_1 \oplus \mathcal{I}_2$ s.t. $\mathcal{I}_i \models \mathcal{U}_i$
- $\mathcal{I} \models \mathcal{U}_1 \ominus \mathcal{U}_2$ iff there is $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{U}_1$ and $\mathcal{I} \oplus \mathcal{J} \models \mathcal{U}_2$
Running example: recall $\mathcal{K}$
Running example: recall $\mathcal{K}$

Database ($\text{ABox}$)

Knowledge ($\text{TBox}$)

\[
\begin{align*}
\text{isDrafted}(\text{Zion}, \text{Pelicans}) & : \\
\exists \text{hasInjury}.\text{Injury} & \sqsubseteq \text{Player} \\
\exists \text{isDrafted}.\text{Team} & \sqsubseteq \text{Player} \\
\exists \text{hasInjury}.\text{Injury} \sqcap \exists \text{isDrafted}.\text{Team} & \sqsubseteq \bot
\end{align*}
\]
Running example: recall $\mathcal{K}$

Consider the dynamic axiom:
Consider the dynamic axiom: \( U = \top \ast (\top \oplus isDrafted(Zion, Pelicans)) \)
Running example: recall $\mathcal{K}$

Database (ABox)

Knowledge (TBox)

$isDrafted(Zion, Pelicans)$

$\exists \text{hasInjury.Injury} \subseteq \text{Player}$

$\exists \text{isDrafted.Team} \subseteq \text{Player}$

$\exists \text{hasInjury.Injury} \cap \exists \text{isDrafted.Team} \subseteq \bot$

Consider the dynamic axiom: $\mathbb{U} = \top * (\top \oplus isDrafted(Zion, Pelicans))$
Running example: recall $\mathcal{K}$

Database (ABox)

Knowledge (TBox)

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

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

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

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

Consider the dynamic axiom: $\mathcal{U} = \top \ast (\top \oplus \text{isDrafted}(\text{Zion}, \text{Pelicans}))$

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

In this work we focused on the consistency problem only.

We consider two logics: ALC and EL.

We distinguish the cases of dynamic axioms (DAs) and negation-free DA.

\[
\text{ALC} \overset{\text{ExpTime}}{\models} \text{EL} \\
\text{PTime} \downarrow \text{proof system} \\
\text{ExpTime} \downarrow \text{translation to} \\
\text{ALC} \downarrow \text{undecidable} \\
\text{reduction via} \\
\text{ALC} + r_1 \circ r_2 \circ \ldots \circ r_n \downarrow s.
\]

Check the paper for more details!
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: \textit{ALC} and \textit{EL}

Check the paper for more details!
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

Check the paper for more details!
Our results

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

```
<table>
<thead>
<tr>
<th>pos-EL</th>
<th>pos-ALC</th>
<th>EL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ALC</td>
</tr>
</tbody>
</table>
```

Check the paper for more details!
Our results

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

\begin{align*}
\text{pos-EL} & \quad \text{in PTIME proof system} \\
\text{pos-ALC} & \\
\text{EL} & \quad \text{ALC}
\end{align*}
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

<table>
<thead>
<tr>
<th>pos-$\mathcal{EL}$</th>
<th>pos-$\mathcal{ALC}$</th>
<th>$\mathcal{EL}$ $\mathcal{ALC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>in PTime</td>
<td>ExpTime-compl</td>
<td></td>
</tr>
<tr>
<td>proof system</td>
<td>translation to $\mathcal{ALCOb}$</td>
<td></td>
</tr>
</tbody>
</table>

Check the paper for more details!
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

$\text{pos-EL}$

in PTime

proof system

$\text{pos-ALC}$

ExpTime-compl

translation to $\mathcal{ALCOb}$

$\mathcal{EL}$

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

\[
pos-\mathcal{EL} \quad \text{in PTime proof system} \quad \quad pos-\mathcal{ALC} \quad \text{ExpTime-compl translation to } \mathcal{ALC}Ob \quad \quad \mathcal{EL} \quad \mathcal{ALC} \quad \text{Undecidable reduction via } \mathcal{ALC}+ \quad r_1 \circ r_2 \circ \ldots \circ r_n \sqsubseteq s
\]

Check the paper for more details!