Reasoning with Attributed Description Logics

Markus Krötzsch    Maximilian Marx    Ana Ozaki    Veronika Thost

Center for Advancing Electronics Dresden (cfaed), TU Dresden

30th International Workshop on Description Logics

Full paper: https://iccl.inf.tu-dresden.de/web/Inproceedings3154
Why Attributed DLs?

- Liz Burton & Richard Taylor in Wikidata:

  - Spouse edges may occur with multiple distinct annotations.
  - Annotations: finite attribute-value sets, attached to concept & role names.
  - E.g., spouse is symmetric, so inverses should coincide on start & end:

```
X ⌋ (spouse @ X ⊑ spouse − @ ⌋ start: X.start, end: X.end)
```

Krötzsch, Marx, Ozaki, Thost (TU Dresden)

Reasoning with Attributed Description Logics

DL 2017 2 / 10
Why Attributed DLs?

- Liz Burton & Richard Taylor in Wikidata:
  
  ![Diagram of Liz Burton and Richard Taylor]
  
  - start : 1964, end : 1974
  - start : 1975, end : 1976

- edges may occur with multiple distinct annotations
- annotations: finite attribute–value sets, attached to concept & role names
Why Attributed DLs?

- Liz Burton & Richard Taylor in Wikidata:
  
  ![Diagram](image.png)

- edges may occur with multiple distinct annotations
- annotations: finite attribute–value sets, attached to concept & role names
- e.g., spouse is symmetric, so inverses should coincide on start & end

\[ X : [ ] \quad (\text{spouse} @ X \sqsubseteq \text{spouse}^- @ [\text{start} : X.\text{start}, \text{end} : X.\text{end}]) \]
Specifiers: constraining annotations

- two flavours of annotations: open & closed specifiers

  consider  spouse(taylor, burton)@[start : 1964, end : 1974]

  \[
  \begin{align*}
  & [ ] \checkmark \quad [ ] \times \\
  & [\text{start} : 1964] \checkmark \quad [\text{start} : 1964] \times \\
  & [\text{start} : 1964, \text{end} : 1974] \checkmark \quad [\text{start} : 1964, \text{end} : 1974] \checkmark \\
  & [\text{start} : 1964, \text{loc} : \text{Montreal}] \times \quad [\text{start} : 1964, \text{end} : 1974, \text{loc} : *] \checkmark \\
  & [\text{start} : 1964, \text{end} : +] \checkmark \quad [\text{start} : 1964, \text{end} : 1974, \text{loc} : +] \times
  \end{align*}
  \]

- simplification: instead of \( C@[ ] \), write \( C \)
Attributed DL axioms

Axioms may use variables in annotation positions:

- all variables are universally quantified

\[ \text{spouse}@X \sqsubseteq \text{spouse}^{-}@X \]
Attributed DL axioms

Axioms may use variables in annotation positions:

- all variables are universally quantified

\[ \text{spouse} @ X \sqsubseteq \text{spouse}^{-} @ X \]

- variables may be constrained by using a specifier

\[ X : [\text{start} : 1964] \quad (\text{spouse} @ X \sqsubseteq \text{spouse}^{-} @ X) \]
Attributed DL axioms

Axioms may use variables in annotation positions:

- all variables are universally quantified

\[
\text{spouse}@X \sqsubseteq \text{spouse}^-@X
\]

- variables may be constrained by using a specifier

\[
X : [\text{start} : 1964] \quad (\text{spouse}@X \sqsubseteq \text{spouse}^-@X)
\]

- annotations may refer to assignments in other annotations

\[
X : [\text{start} : 1964], Y : [\text{start} : X\.\text{start}, \text{end} : Y\.\text{end}] \\
(\text{spouse}@X \sqsubseteq \text{spouse}^-@X)
\]
Attributed DL axioms

Axioms may use variables in annotation positions:

- all variables are universally quantified

\[
\text{spouse}@X \sqsubseteq \text{spouse}^-@X
\]

- variables may be constrained by using a specifier

\[
X: [\text{start} : 1964] \quad (\text{spouse}@X \sqsubseteq \text{spouse}^-@X)
\]

- annotations may refer to assignments in other annotations

\[
X: [\text{start} : 1964], Y: [\text{start} : X.\text{start}, \text{end} : Y.\text{end}] \\
(\text{spouse}@X \sqsubseteq \text{spouse}^-@X)
\]

- note: cyclic references are allowed
## Complexity of Reasoning in Attributed DLs

<table>
<thead>
<tr>
<th>DL</th>
<th>ground</th>
<th>restricted</th>
<th>unrestricted</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{EL}@$</td>
<td>PTime</td>
<td>PTime/PSpace-hard*</td>
<td>ExpTime</td>
</tr>
<tr>
<td>$\mathcal{ALCH}@$</td>
<td>ExpTime</td>
<td>ExpTime</td>
<td>2ExpTime</td>
</tr>
<tr>
<td>$\mathcal{SROIQ}@$</td>
<td>N2ExpTime</td>
<td>N2ExpTime</td>
<td>N2ExpTime</td>
</tr>
<tr>
<td>$\mathcal{EL}@+$</td>
<td>PTime</td>
<td>ExpTime</td>
<td>undecidable</td>
</tr>
<tr>
<td>$\mathcal{ALCH}@+$</td>
<td>ExpTime</td>
<td>2ExpTime</td>
<td>undecidable</td>
</tr>
</tbody>
</table>

- except for PSpace-hardness, bounds are tight
- Nominals require special handling (bounds on domain size)
Reasoning for ground KBs

Introduce fresh concept/role names for each annotated concept/role

- yields polynomially larger KB in underlying, classical DL:

\[
\text{spouse}(\text{taylor}, \text{burton}) \circlearrowleft_{\text{start: 1964, end: 1974}} \quad (1)
\]
\[
\text{spouse} \circlearrowleft_{\text{start: 1964}} \subseteq \text{spouse}^{-} \circlearrowleft_{\text{start: 1964}} \quad (2)
\]

\[
\leadsto \quad \text{spouse}_{\text{start:1964, end:1974}}(\text{taylor}, \text{burton})\] 
\[
\text{spouse}_{\text{start:1964}} \subseteq \text{spouse}_{\text{start:1964}}^{-}
\]
Reasoning for ground KBs

Introduce fresh concept/role names for each annotated concept/role

- yields polynomially larger KB in underlying, classical DL:
  \[
  \text{spouse}(\text{taylor, burton}) @ [\text{start} : 1964, \text{end} : 1974] \tag{1}
  \]
  \[
  \text{spouse} @ [\text{start} : 1964] \sqsubseteq \text{spouse}^- @ [\text{start} : 1964] \tag{2}
  \]
  \[
  \implies \text{spouse}_{[\text{start:1964,end:1974}]}(\text{taylor, burton}) \subseteq \text{spouse}^-_{[\text{start:1964}]} \tag{3}
  \]

- interactions between open & closed specifiers: (1), (2) entails
  \[
  \text{spouse}(\text{burton, taylor}) @ [\text{start} : 1964],
  \]

but we do not get \[
\text{spouse}_{[\text{start:1964}]}(\text{burton, taylor})
\]
Reasoning in Attributed DLs

Reasoning for ground KBs

Introduce fresh concept/role names for each annotated concept/role

- yields polynomially larger KB in underlying, classical DL:

\[
\text{spouse}(taylor, burton) @ [\text{start} : 1964, \text{end} : 1974] \quad (1) \\
\text{spouse} @ [\text{start} : 1964] \sqsubseteq \text{spouse}^- @ [\text{start} : 1964] \quad (2)
\]

\[\leadsto\]

\[
\text{spouse}_{[\text{start}:1964, \text{end}:1974]}(taylor, burton) \\
\text{spouse}_{[\text{start}:1964]} \sqsubseteq \text{spouse}_{[\text{start}:1964]}
\]

- interactions between open & closed specifiers: (1), (2) entails

\[
\text{spouse}(burton, taylor) @ [\text{start} : 1964],
\]

but we do not get \(\text{spouse}_{[\text{start}:1964]}(burton, taylor)\)

- axiomatise these inclusions: \(\text{spouse}_{[\text{start}:1964, \text{end}:1974]} \sqsubseteq \text{spouse}_{[\text{start}:1964]}\)
Dealing with non-ground KBs

Transform KB into a ground KB:
- instantiate each axiom for every possible annotation

\[
\text{spouse}(taylor, burton)@[start : 1964] \quad \text{spouse}@X \sqsubseteq \text{spouse}^-@X
\]
\[
\text{spouse}(taylor, burton)@[start : 1974]
\]

\[
\rightsquigarrow \text{spouse}_{[start:1964]}(taylor, burton) \quad \text{spouse}_{[start:1974]}(taylor, burton)
\]
\[
\text{spouse}_{[start:1964]} \sqsubseteq \text{spouse}^-_{[start:1964]} \\
\text{spouse}_{[start:1974]} \sqsubseteq \text{spouse}^-_{[start:1974]}
\]

unfortunately, the grounding is exponential in the size of KB:
Dealing with non-ground KBs

Transform KB into a ground KB:

- instantiate each axiom for every possible annotation

\[ \text{spouse}(\text{taylor}, \text{burton}) @ [\text{start} : 1964] \quad \text{spouse} @ X \sqsubseteq \text{spouse}^\sim @ X \]
\[ \text{spouse}(\text{taylor}, \text{burton}) @ [\text{start} : 1974] \]

\[ \leadsto \text{spouse}_{[\text{start}:1964]}(\text{taylor}, \text{burton}) \quad \text{spouse}_{[\text{start}:1974]}(\text{taylor}, \text{burton}) \]
\[ \text{spouse}_{[\text{start}:1964]} \sqsubseteq \text{spouse}^\sim_{[\text{start}:1964]} \]
\[ \text{spouse}_{[\text{start}:1974]} \sqsubseteq \text{spouse}^\sim_{[\text{start}:1974]} \]

- unfortunately, the grounding is exponential in the size of KB:

\[ C(a) @ [ ] \quad C(a) @ [b : b] \]
\[ C @ X \cap C @ Y \cap C @ Z \sqsubseteq C @ X \]

- syntactic restrictions ensure a polynomial grounding
Regaining Tractability for $\mathcal{EL}@$

Sufficient conditions for polynomial grounding:

(A) number of variables per axiom is bounded,

(B) number of ‘dots’ $X.a$ is bounded, and

(C) no merging with ‘dots’: if $a : X.b$ occurs in some annotation $S$, then there is no further assignment for $a$ in $S$. 

Violating (C) results in $\text{PSPACE}$-hardness reasoning for $\mathcal{ALCH}@$ $\text{KBs}$ satisfying the conditions is $\text{EXPSPACE}$-complete.
Regaining Tractability for $\mathcal{EL}@$

Sufficient conditions for polynomial grounding:

(A) number of variables per axiom is bounded,

(B) number of ‘dots’ $X.a$ is bounded, and

(C) no merging with ‘dots’: if $a : X.b$ occurs in some annotation $S$, then there is no further assignment for $a$ in $S$

- violating any condition yields intractability for $\mathcal{EL}@$
- violating (C) results in PSPACE-hardness
Regaining Tractability for $\mathcal{EL}@$

Sufficient conditions for polynomial grounding:

(A) number of variables per axiom is bounded,

(B) number of ‘dots’ $X.a$ is bounded, and

(C) no merging with ‘dots’: if $a : X.b$ occurs in some annotation $S$, then there is no further assignment for $a$ in $S$

- violating any condition yields intractability for $\mathcal{EL}@$
- violating (C) results in PSPACE-hardness
- reasoning for $\mathcal{ALCH}@$ KBs satisfying the conditions is ExpTime-complete
An Undecidable Case

Without restrictions, Attributed DLs with + are undecidable:

- interaction of $X.a$ and + admits an encoding of Existential Rules in quantifier-free attributed $\mathcal{EL}$
- forbidding either $X.a$ or + is sufficient to recover decidability
- practically, $X.a$ is more relevant
An Undecidable Case

Without restrictions, Attributed DLs with + are undecidable:

- interaction of $X.a$ and + admits an encoding of Existential Rules in quantifier-free attributed $\mathcal{EL}$
- forbidding either $X.a$ or + is sufficient to recover decidability
- practically, $X.a$ is more relevant, but + adds expressive power:
  
  $\text{educatedAt}@\left[\text{degree} : +\right] \sqsubseteq \text{obtainedDegreeFrom}$

- decidability results for Existential Rules suggest that a weaker condition may suffice for decidability
An Undecidable Case

Without restrictions, Attributed DLs with + are undecidable:

- interaction of $X.a$ and + admits an encoding of Existential Rules in quantifier-free attributed $\mathcal{EL}$
- forbidding either $X.a$ or + is sufficient to recover decidability
- practically, $X.a$ is more relevant, but + adds expressive power:

$$\text{educatedAt}@[\text{degree} : +] \subseteq \text{obtainedDegreeFrom}$$

- decidability results for Existential Rules suggest that a weaker condition may suffice for decidability
- Corollary: Attributed DLs (without +) capture Datalog
Summary & Outlook

Summary:
- we add annotations (sets of attribute–value pairs) to concept and role names
- specifiers allow to constrain variables in axioms
- ‘ground and rename’ reasoning approach
- attributed reasoning is exponentially harder
- syntactic conditions ensure that we avoid this blowup
Summary & Outlook

Summary:
- we add annotations (sets of attribute–value pairs) to concept and role names
- specifiers allow to constrain variables in axioms
- ‘ground and rename’ reasoning approach
- attributed reasoning is exponentially harder
- syntactic conditions ensure that we avoid this blowup

Future Work:
- data complexities
- extension to further DL constructs ($\mathcal{EL}^{++?}$)
- annotation-aware reasoning algorithms