Towards intended models: accounting for minimized and fixed predicates in DLs

Magdalena Ortiz
magdalena.ortiz@cs.umu.se

July 4, 2023
Family of logics well-suited for describing structured knowledge through concepts (classes) and roles (relationships)

Decidable fragments of first order logic in a funny syntax

\[
\begin{align*}
\text{EUCountry} & \sqsubseteq \text{Country} \\
\text{Person} & \sqsubseteq \exists \text{homeCountry.Country} \\
\text{City} & \sqsubseteq \exists \text{inCountry.Country} \\
\text{City} & \sqsubseteq \leq 1 \text{inCountry.T} \\
\text{EUCity} & \equiv \text{City} \sqcap \exists \text{inCountry.EUCountry} \\
\text{EUCitizen} & \equiv \text{Person} \sqcap \exists \text{homeCountry.EUCountry} \\
\end{align*}
\]

Person(Alice), Country(UK), EUCountry(Sweden), inCountry(Umeå,Sweden), homeCountry(Bob,UK) ...
Family of logics well-suited for describing structured knowledge through concepts (classes) and roles (relationships)

Decidable fragments of first order logic in a funny syntax

\[
\begin{align*}
\text{EUCountry} & \sqsubseteq \text{Country} \\
\text{Person} & \sqsubseteq \exists \text{homeCountry}.\text{Country} \\
\text{City} & \sqsubseteq \exists \text{inCountry}.\text{Country} \\
\text{City} & \sqsubseteq \leq 1 \text{inCountry}.\top \\
\text{EUCity} & \equiv \text{City} \sqcap \exists \text{inCountry}.\text{EUCountry} \\
\text{EUCitizen} & \equiv \text{Person} \sqcap \exists \text{homeCountry}.\text{EUCountry} \\
\end{align*}
\]

Person(Alice), Country(UK), EUCountry(Sweden), inCountry(Umeå, Sweden), homeCountry(Bob, UK) ...

DL knowledge base = TBox (axioms) + ABox (data)
guarded quantification, no explicit variables, function-free fragments
mostly contained in $\text{FO}^2$ or $\text{C}^2$

KBs are theories with two types of formulas:

- **ABox**: data, facts $B(c), B(d), r(c, d), r_2(d, e) \ldots$
- **TBox**: axioms, universal formulas

\[
\begin{align*}
A \sqcap B & \sqsubseteq C & \forall x \ (A(x) \land B(x) \rightarrow C(x)) \\
A & \sqsubseteq \exists r. B & \forall x \ A(x) \rightarrow \exists y \ r(x, y) \land B(y) \\
C & \sqsubseteq \forall r. \{a\} & \forall x, y \ C(x) \land r(x, y) \rightarrow y = a \\
r & \sqsubseteq s & \forall x, y \ r(x, y) \rightarrow s(x, y)
\end{align*}
\]
**DLs are a toolbox:**

- **different constructors** combine into many DLs
- **decidable fragments** of standard FOL
- **choice of the right logic**
- **taking into account the computational cost**

<table>
<thead>
<tr>
<th>Concept</th>
<th>DL-Lite</th>
<th>EL</th>
<th>ALC</th>
<th>SHIQ</th>
<th>SHOIQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUCountry ⊑ Country</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Country ⊑ EUCount □ 3rdCount</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EUCity ⊑ ∃inCountry.EUCount</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>∃citizOf.EUCount ⊑ EUCitiz</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EUCountry ⊑ {DE, AT, SE, IT, FR,...}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>inCountry ⊑ locatedIn</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>trans(locatedIn)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
The DL ALCHOIF (in normal form)

N_C: set of concept names  N_R: set of role names  N_I: set of constants

Nominals: \{a\}, for \(a \in N_I\)  Roles: \(r, r^-, \) for \(r \in N_R\)

TBox:  \(B_1 \sqcap \cdots \sqcap B_{k-1} \subseteq B_k \sqcup \cdots \sqcup B_m\)
       \(B_1 \subseteq \exists r.B_2\)
       \(B_1 \subseteq \forall r.B_2\)
       \(r_1 \subseteq r_2\)
       (func \(r\))

\(B_i\): concept name or a nominal, \(r_j\): role.

ABox:  \((-)B(a)\)
        \((-)r(a, b)\)

KB:  \((\mathcal{T}, \mathcal{A})\)  \(\mathcal{T}\) TBox, \(\mathcal{A}\) ABox

Models: FO interpretation \(\mathcal{I}\)  s.t. \(\mathcal{I}\) satisfies \(\mathcal{T}\) and \(\mathcal{A}\)
The DL $\text{ALCHOIF}$ (in normal form)

$N_C$: set of concept names   \hspace{1cm} $N_R$: set of role names   \hspace{1cm} $N_I$: set of constants

**Nominals:** $\{a\}$, for $a \in N_I$  \hspace{1cm} **Roles:** $r, r^-$, for $r \in N_R$

**TBox:**

- $B_1(x) \land \cdots \land B_{k-1}(x) \rightarrow B_k(x) \lor \cdots \lor B_m(x)$
- $B_1(x) \rightarrow \exists y. B_2(y) \land r(x, y)$
- $B_1(x) \land r(x, y) \rightarrow B_2(y)$
- $r_1(x, y) \rightarrow r_2(x, y)$
- $r(x, y) \land r(x, z) \rightarrow y = z$

$B_i$: concept name or a nominal, $r_j$: role.

**ABox:**

- $(\neg)B(a)$
- $(\neg)r(a, b)$

**KB:** $(\mathcal{T}, \mathcal{A}) \models \text{TBox}, \mathcal{A} \models \text{ABox}$

**Models:** FO interpretation $\mathcal{I}$ s.t. $\mathcal{I}$ satisfies $\mathcal{T}$ and $\mathcal{A}$
Reasoning services: concept subsumption, instance checking, concept satisfiability

Classical services reduce to satisfiability
Reasoning services: concept subsumption, instance checking, concept satisfiability

Classical services reduce to satisfiability

DLs have shed light on the decidability & complexity of FO fragments
DL Reasoning

- **Reasoning services**: concept subsumption, instance checking, concept satisfiability
- **Classical services** reduce to satisfiability

DLs have shed light on the decidability & complexity of FO fragments

Here, three types of DLs:

1. **Lightweight DLs**: tractable
   - DL-Lite
   - EL

2. **$ALC$ and ‘forest-like’ extensions**: ExpTime complete
   - $ALCHI$
   - $SHIQ$
   - $SHOI$

3. **The hard DLs**: nominals + inverses + counting: NExpTime complete
   - $ALCHOIQ$
   - $SHOIQ$
A model is an FO interpretation $I$ that satisfies $T$ and $A$. Sometimes we want:

- finite models
- minimal models
- models where some predicates are finite
- models where some predicates are minimized
- ...
A model is an FO interpretation $I$ that satisfies $T$ and $A$

But in KR, these are not always intended models

Sometimes we want:
- finite models
- minimal models
A **model** is an FO interpretation $\mathcal{I}$ that satisfies $\mathcal{T}$ and $\mathcal{A}$

But in KR, these are not always **intended models**

Sometimes we want:
- finite models
- minimal models
- models where some predicates are finite
A model is an FO interpretation $I$ that satisfies $T$ and $A$.

But in KR, these are not always intended models.

Sometimes we want:
- finite models
- minimal models
- models where some predicates are finite
- models where some predicates are minimized
A model is an FO interpretation $I$ that satisfies $T$ and $A$.

But in KR, these are not always intended models.

Sometimes we want:

- finite models
- minimal models
- models where some predicates are finite
- models where some predicates are minimized
- …
Adopted by many formalisms (logic programming, rule languages)

Good for common-sense reasoning

Example

```
Pizza(margh), Vegetarian(tomt), Vegetarian(mozz), 
hasIngredient(margh, tmt), hasIngredient(margh, mozz), 

Pizza ⊓ ∀hasIngredient. Vegetarian ⊑ VegetarianPizza
```

We can conclude VegetarianPizza(margh)
Adopted by many formalisms (logic programming, rule languages)

Good for common-sense reasoning

Example

Pizza(margh), Vegetarian(tomt), Vegetarian(mozz),
hasIngredient(margh, tomt), hasIngredient(margh, mozz),

\[
\text{Pizza} \sqcap \forall \text{hasIngredient}. \text{Vegetarian} \sqsubseteq \text{VegetarianPizza}
\]

We can conclude VegetarianPizza(margh)

Inference is non-monotonic hasIngredient(margh, prosc)
Useful for drawing defeasible inferences.

Example

\[
\begin{align*}
\text{Bird} & \sqcap \neg \text{Flies} \subseteq \text{Ab}_\text{bird} \\
\text{Student} & \sqcap \exists \text{hasJob} \subseteq \text{Ab}_\text{stud} \\
\text{Mammal} & \sqcap \exists \text{laysEggs} \subseteq \text{Ab}_\text{mamm}
\end{align*}
\]

- students usually don’t have a job
- students usually don’t have a job
- mammals usually don’t lay eggs
Closed Predicates

- force the closed-world assumption in specific predicates
- allow to combine incomplete and complete information
- applications in KR, privacy, data quality

**KB:** $(\mathcal{T}, \mathcal{A}, \Sigma)$

$\Sigma$ is a set of predicates (concepts and roles)

- ScandinavCount(Norway), ScandinavCount(Sweden), ScandinavCount(Denmark)
- hasIngred(margh, tomt), hasIngred(margh, mozz)

**Models:** FO interpretation $\mathcal{I}$ s.t.

- $\mathcal{I}$ satisfies $\mathcal{T}$ and $\mathcal{A}$
- for all $p \in \Sigma$, $p^\mathcal{I} = \{\bar{c} : p(\bar{c}) \in \mathcal{A}\}$

**Standard Name Assumption:** $a^\mathcal{I} = a$, for $a \in N_\mathcal{I}$
**Example**

*ALCHOIF* with closed predicates can express e.g., the even query

\[
A \equiv B_1 \cup B_2 \quad B_1 \cap B_2 \sqsubseteq \bot \quad B_1 \sqsubseteq 1.r.B_2 \quad B_2 \sqsubseteq 1.r.B_1 \quad T \sqsubseteq 1.r.T \quad T \sqsubseteq 1.r^-T
\]
A **circumscription pattern** $\mathcal{P}$ is a partition of the predicates into three sets $M$ minimized, $V$ varying, $F$ fixed.


Definition

Let $\mathcal{P} = (M, V, F)$ interpretations and $\mathcal{I}, \mathcal{J}$ interpretations. $\mathcal{I} \preceq \mathcal{J}$ if:

- $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $a^\mathcal{I} = a^\mathcal{J}$ for all individuals $a$,
- $Q^\mathcal{I} \subseteq Q^\mathcal{J}$ for all $Q \in M$, and
- $Q^\mathcal{I} = Q^\mathcal{J}$ for all $Q \in F$.

$\mathcal{I} \prec \mathcal{J}$, if $\mathcal{I} \preceq \mathcal{J}$ and $Q^\mathcal{I} \subset Q^\mathcal{J}$ for some $Q \in M$.

$\mathcal{I} \models \text{Circ}_\mathcal{P}(\mathcal{K})$ if $\mathcal{I} \models \mathcal{K}$ and there is no $\mathcal{J}$ s.t. $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \prec \mathcal{I}$.
Circumscription is very hard!

<table>
<thead>
<tr>
<th>Concept circ.</th>
<th>ALC</th>
<th>ALCQO</th>
<th>ALCI</th>
<th>ALCIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>#M ≤ n, #F ≤ n</td>
<td></td>
<td>NP^{NExp}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(unrestricted)</td>
<td>NExp^{NP}</td>
<td>even if &lt;= ∅, and either TBox=∅ or ABox=∅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minim. roles</td>
<td>TBox=∅</td>
<td>NExp^{NP}</td>
<td>even if #M ≤ 1, #F ≤ 0</td>
<td>Undecidable</td>
</tr>
<tr>
<td></td>
<td>TBox≠ ∅</td>
<td></td>
<td>Undecidable</td>
<td></td>
</tr>
<tr>
<td>Fixed roles</td>
<td></td>
<td></td>
<td>Highly undecidable</td>
<td>even if TBox=∅, &lt;= ∅</td>
</tr>
</tbody>
</table>

The Complexity of Circumscription in Description Logic (Bonatti, Lutz & Wolter, 2009)
Minimized and Closed Predicates

- Minimal model reasoning largely neglected
- Closed predicates are also hard

<table>
<thead>
<tr>
<th></th>
<th>Without closed predicates</th>
<th>With closed predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-Lite</td>
<td>NL</td>
<td>NP</td>
</tr>
<tr>
<td>$\mathcal{EL}$</td>
<td>P</td>
<td>Exp</td>
</tr>
<tr>
<td>$\mathcal{ALCO} \ldots$</td>
<td>Exp</td>
<td>Exp</td>
</tr>
</tbody>
</table>

- Closed predicates cause **NP hardness in data complexity** in most DLs
- For expressive DLs few upper bounds
Some recent work

- **tight NP upper bound for data complexity of** \textit{ALCHOIF} \textit{with closed predicates}
- **reduction to an exponentially large system of linear inequalities**
  - tiles describe local configurations in models
  - a full mosaic assigns them a multiplicity in \( \mathbb{N}^* = \mathbb{N} \cup \{\aleph_0\} \), where \( \aleph_0 \) represents infinity
  - a variable for each tile needed
  - \textit{NexpTime} in combined complexity
- **used for a polynomial reduction into Datalog**
**Type** = concept names that a domain element participates in + (possibly) a nominal (i.e. constant) it represents

**Tile (for a KB)** = a compact description of some domain element $d$ and its relevant neighborhood

- **type** for $d$ + a **set of neighbors** for $d$ described by their types and roles that connect them to $d$
- satisfies conditions that ensure descriptions are consistent with the given KB
- a building block for constructing models
Tiles ctd.

TBox:
\[ A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists s. C \]
\[ B \sqsubseteq \exists r. D \quad A \sqsubseteq \forall r. B \]
\[ s \sqsubseteq r \quad (\text{func } s) \]

ABox:
\[ C(a) \quad C(b) \quad s(a, b) \quad r(a, b) \]
\[ \Sigma = \{ C, s \} \]

\[ \{ \{ s, r \}, \{ \top, B, C, \{ b \} \}, \{ r, \{ \top, B, D \} \} \} \]

\[ \{ \{ \top, A, B, C, \{ a \} \}, \{ \{ s, r \}, \{ \top, B, C, \{ b \} \}, \{ r, \{ \top, B, D \} \} \} \} \]

\( \Sigma = \{ C, s \} \)
Tiles ctd.

\[\{\top, A, B, C\}\setminus\{a\}\]

\[\{s, r\},\{\top, B, C, \{b\}\}\]

\[\{r\},\{\top, B, D\}\}\]

**TBox:**
- \(A \sqcap B \sqsubseteq C\)
- \(A \sqsubseteq \exists s . C\)
- \(B \sqsubseteq \exists r . D\)
- \(A \sqsubseteq \forall r . B\)
- \(s \sqsubseteq r\)
- \(\text{(func } s)\)

**ABox:**
- \(C(a)\)
- \(C(b)\)
- \(s(a, b)\)
- \(r(a, b)\)

\(\Sigma = \{C, s\}\)
Tiles ctd.

TBox:

\[
A \cap B \subseteq C \quad A \subseteq \exists s.C \\
B \subseteq \exists r.D \quad A \subseteq \forall r.B \\
s \subseteq r \\
(funct \ s)
\]

ABox:

\[
C(a) \quad C(b) \quad s(a, b) \quad r(a, b) \\
\Sigma = \{C, s\}
\]

\[
(\{\top, A, B, C, \{a\}\}, \{\{s, r\}, \{\top, B, C, \{b\}\}\}, \{\{r\}, \{\top, B, D\}\}))
\]

\[
\text{type} \quad \text{neighbor} \quad \text{neighbor}
\]
Tiles ctd.

\[
\begin{align*}
&\text{TBox:} \\
&\quad A \sqcap B \subseteq C \quad A \sqsubseteq \exists s.C \\
&\quad B \subseteq \exists r.D \quad A \sqsubseteq \forall r.B \\
&\quad s \subseteq r \\
&\text{(func } s) \\
&\text{ABox:} \\
&\quad C(a) \quad C(b) \quad s(a, b) \quad r(a, b) \\
&\Sigma = \{C, s\}
\end{align*}
\]
Tiles ctd.

\[\begin{align*}
\text{TBox:} & \quad A \cap B \sqsubseteq C \\
& \quad A \sqsubseteq \exists s.C \\
& \quad B \sqsubseteq \exists r.D \\
& \quad A \sqsubseteq \forall r.B \\
& \quad s \sqsubseteq r \\
& \quad (\text{func } s) \\
\text{ABox:} & \quad C(a) \quad C(b) \quad s(a, b) \quad r(a, b) \\
& \quad \Sigma = \{C, s\}
\end{align*}\]

\[\begin{align*}
\{\top, A, B, C\} & , \{(s, r)\} \\
\{\top, B, C\} & , \{\top, B, D\} \\
\{\top, A, B, C\} & , \{(s, r)\}
\end{align*}\]

\begin{itemize}
\item type
\item neighbor
\item neighbor
\end{itemize}
Tiles ctd.

\[
\begin{align*}
TBox: & \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists s.C \\
& \quad B \sqsubseteq \exists r.D \quad A \sqsubseteq \forall r.B \\
& \quad s \sqsubseteq r \\
& \quad (\text{func } s)
\end{align*}
\]

\[
\begin{align*}
ABox: & \quad C(a) \quad C(b) \quad s(a, b) \quad r(a, b) \\
\Sigma = & \{C, s\}
\end{align*}
\]

\[
\begin{align*}
\{\top, A, B, C\}, \{\{s, r\}, \{\top, B, C\}, \{b\}\}, \{\{r\}, \{\top, B, D\}\}\)
\end{align*}
\]

- type
- neighbor
- neighbor
In order to keep our tiles small and data-independent, we keep track only of the neighbors that are used for satisfying existential axioms.

TBox:
\[
\begin{align*}
A \sqcap B & \sqsubseteq C \\
B & \sqsubseteq \exists r.D \\
s & \sqsubseteq r
\end{align*}
\]

ABox:
\[
\begin{align*}
C(a) & \quad C(b) \\
s(a, b) & \quad r(a, b)
\end{align*}
\]

\[\Sigma = \{C, s\}\]
Tiles ctd.

TBox:
\[ A \cap B \subseteq C \quad A \subseteq \exists s.C \]
\[ B \subseteq \exists r.D \quad A \subseteq \forall r.B \]
\[ s \subseteq r \quad (\text{func } s) \]

ABox:
\[ C(a) \quad C(b) \quad s(a,b) \quad r(a,b) \]
\[ \Sigma = \{C, s\} \]
Tiles ctd.

\[
\begin{align*}
\text{TBox:} & \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists s.C \\
& \quad B \sqsubseteq \exists r.D \quad A \sqsubseteq \forall r.B \\
& \quad s \sqsubseteq r \quad (\text{func } s)
\end{align*}
\]

\[
\begin{align*}
\text{ABox:} & \quad C(a) \quad C(b) \quad s(a,b) \quad r(a,b) \\
\Sigma & = \{C, s\}
\end{align*}
\]

\[
\begin{align*}
& \{(\top, A, B, C, \{a\}), (\{s, r\}, \{\top, B, C, \{b\}\}), (\{r\}, \{\top, B, D\})\} \\
& \quad \text{type} \quad \text{neighbor} \quad \text{neighbor}
\end{align*}
\]
Tiles ctd.

TBox:
\[
\begin{align*}
A \cap B & \subseteq C & A \subseteq \exists s. C \\
B & \subseteq \exists r. D & A \subseteq \forall r. B \\
s & \subseteq r & (\text{func } s)
\end{align*}
\]

ABox:
\[
\begin{align*}
C(a) & \quad C(b) & s(a, b) & \quad r(a, b) \\
\Sigma = \{C, s\}
\end{align*}
\]

\[
\{\{\top, A, B, C, \{a\}\}, \{(s, r), \{\top, B, C, \{b\}\}\}, \{(r), \{\top, B, D\}\}\} \quad \text{type neighbor neighbor}
\]

In order to keep our tiles small and data-independent, we keep track only of the neighbors that are used for satisfying existential axioms.
in order to keep our tiles small and data-independent, we keep track only of the neighbors that are used for satisfying existential axioms
Given a KB $\mathcal{K} = (\mathcal{T}, \Sigma, \mathcal{A})$, a tile for $\mathcal{K}$ is a tuple $\tau = (T, \rho)$, where $T \in \text{Types}(\mathcal{K})$ and $\rho$ is a set of pairs $(R, T')$, where $R \subseteq N^+_R(\mathcal{K})$, $T' \in \text{Types}(\mathcal{K})$, and the following conditions are satisfied:

1. $|\rho| \leq |T|$
2. $B_1 \cap \cdots \cap B_{k-1} \subseteq B_k \cup \cdots \cup B_m \in \mathcal{T}$ and $\{B_1, \ldots, B_{k-1}\} \subseteq T$ implies $\{B_k, \ldots, B_m\} \cap T \neq \emptyset$
3. $A \subseteq \exists r. B \in \mathcal{T}$ and $A \in T$ implies there exists $(R, T') \in \rho$ s.t. $r \in R$ and $B \in T'$
4. For all $(R, T') \in \rho$:
   1. $A \subseteq \forall r. B \in \mathcal{T}$, $A \in T$ and $r \in R$ implies $B \in T'$
   2. $A \subseteq \forall r. B \in \mathcal{T}$, $A \in T'$ and $r^{-} \in R$ implies $B \in T$
   3. $r \subseteq s \in \mathcal{T}$ and $r \in R$ implies $s \in R$
5. $(\text{func } r) \in T$ implies $|(\{R, T'\} \in \rho : r \in R)| \leq 1$
6. $A(b) \in \mathcal{A}$ and $\{b\} \in T$ implies $A \in T$
7. $\neg A(b) \in \mathcal{A}$ and $\{b\} \in T$ implies $A \notin T$
8. For all $(R, T') \in \rho$:
   1. $p(a, b) \in \mathcal{A}$, $\{p \subseteq r, (\text{func } r)\} \subseteq \mathcal{T}$, $\{a\} \in T$ and $r \in R$ implies $\{b\} \in T'$
   2. $p(a, b) \in \mathcal{A}$, $\{p \subseteq r, (\text{func } r^{-})\} \subseteq \mathcal{T}$, $\{b\} \in T$, and $r^{-} \in R$ implies $\{a\} \in T'$
   3. $\neg p(a, b) \in \mathcal{A}$, $r \subseteq p \in \mathcal{T}$, $\{a\} \in T$, and $r \in R$ implies $\{b\} \notin T'$
   4. $\neg p(a, b) \in \mathcal{A}$, $r \subseteq p^{-} \in \mathcal{T}$, $\{b\} \in T$, and $r \in R$ implies $\{a\} \notin T'$
9. $A \in \Sigma \cap N_C$ and $A \in T$ implies there exists $c \in N_I$ s.t. $\{c\} \in T$ and $A(c) \in \mathcal{A}$
10. If $r \in \Sigma \cap N_R$, then for all $(R, T') \in \rho$ with $r \in R$, there exist $c, d \in N_I$ such that $\{c\} \in T$, $\{d\} \in T'$ and $r(c, d) \in \mathcal{A}$.  

Tiles ctd.
Mosaics (for a KB) = functions that assign multiplicity (natural number of infinity) to each tile for this KB

They satisfy conditions to ensure that a model can be build by instantiating tiles according to their multiplicities:

• there is at least one domain element
Mosaics (for a KB) = functions that assign multiplicity (natural number of infinity) to each tile for this KB

They satisfy conditions to ensure that a model can be build by instantiating tiles according to their multiplicities:

- there is at least one domain element
- each nominal corresponds to exactly one domain element
Mosaics (for a KB) = functions that assign multiplicity (natural number of infinity) to each tile for this KB

They satisfy conditions to ensure that a model can be build by instantiating tiles according to their multiplicities:

- there is at least one domain element
- each nominal corresponds to exactly one domain element
- each domain element can find required neighbor s.t. functionality assertions and universal axioms are respected
A mosaic for $\mathcal{K} = (\mathcal{T}, \Sigma, \mathcal{A})$ is a function $N : \text{Tiles}(\mathcal{K}) \rightarrow \mathbb{N}^+$ such that:

1. For every $\{c\} \in \mathbb{N}_C^+(\mathcal{K})$:
   \[ \sum_{(T, \rho) \in \text{Tiles}(\mathcal{K}), \{c\} \in T} N((T, \rho)) = 1 \]

2. \[ \sum_{\tau \in \text{Tiles}(\mathcal{K})} N(\tau) \geq 1 \]

3. For $T, T' \in \text{Types}(\mathcal{K})$ and $R \subseteq \mathbb{N}_R^+(\mathcal{K})$ with $r \in R$ and $(\text{func } r^-) \in \mathcal{T}$:
   \[ \sum_{(T, \rho) \in \text{Tiles}(\mathcal{K}), (R, T') \in \rho} N((T, \rho)) \leq \sum_{(T', \rho') \in \text{Tiles}(\mathcal{K}), (R', T) \in \rho'} N((T', \rho')) \]

4. For all $(T, \rho) \in \text{Tiles}(\mathcal{K})$ and $(R, T') \in \rho$: $N((T, \rho)) > 0$ implies there exists $\rho'$ s.t. $(T', \rho') \in \text{Tiles}(\mathcal{K})$ and $N((T', \rho')) > 0$.

5. For all $\{a\}, \{b\} \in \mathbb{N}_C^+(\mathcal{K})$ and all $A, B \in \mathbb{N}_C(\mathcal{K})$, if there exist $p, r \in \mathbb{N}_R^+(\mathcal{K})$ s.t.:
   a. $p(a, b) \in \mathcal{A}$, $p \sqsubseteq r \in \mathcal{T}$ and $A \sqsubseteq \forall r.B \in \mathcal{T}$,
   b. $p(b, a) \in \mathcal{A}$, $p \sqsubseteq r^- \in \mathcal{T}$ and $A \sqsubseteq \forall r.B \in \mathcal{T}$,
   c. $p(a, b) \in \mathcal{A}$, $p \sqsubseteq r \in \mathcal{T}$ and $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $(\text{func } r) \in \mathcal{T}$, or
   d. $p(b, a) \in \mathcal{A}$, $p \sqsubseteq r^- \in \mathcal{T}$ and $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $(\text{func } r) \in \mathcal{T}$,

   then:
   \[ \sum_{(T, \rho) \in \text{Tiles}(\mathcal{K}), \{a\} \in T, A \in T} N((T, \rho)) > 0 \] implies \[ \sum_{(T', \rho') \in \text{Tiles}(\mathcal{K}), \{b\} \in T', B \in T'} N((T', \rho')) > 0 \]
Mosaics ctd.

\[ N(A) = N(B) = N(C) = N(D) = 1, \text{ otw. } N(\tau) = 0 \]
Mosaics ctd.

$\Sigma = \{C, s\}$

$N(A) = N(B) = N(C) = N(D) = 1$, \texttt{otw.} $N(\tau) = 0 \implies N \text{ is a mosaic}$
• **Enriched system** \((V, \mathcal{E}, I)\) of integer linear inequalities = 
  (variables, linear inequalities with integer coefficients, implications)

• Conditions placed on mosaics define an *enriched system* \(S_K\) s.t.:
  - tiles for \(K\) are variables in \(S_K\)
  - \(S_K\) is exponential in the size of \(\mathcal{T}\) and polynomial in the size of \(A\)
  - mosaics for \(K\) correspond to the solutions to \(S_K\) over \(\mathbb{N}^* = \mathbb{N} \cup \{\aleph_0\}\), where \(\aleph_0\) represents infinity

\[\Rightarrow K\text{ satisfiable iff } S_K\text{ has a solution}\]

Construct a Datalog\(^-\) program \(P_{\text{sat}}^{\mathcal{T}, \Sigma}\) s.t.
- \(P_{\text{sat}}^{\mathcal{T}, \Sigma} \cup \hat{A}\) has an answer set iff \((\mathcal{T}, \Sigma, A)\) is satisfiable, for all ABoxes \(A\) over the signature of \(\mathcal{T}\),
- \(P_{\text{sat}}^{\mathcal{T}, \Sigma}\) is polynomial in the size of \(\mathcal{T}\) and \(\Sigma\).
Some recent work

- tight NP upper bound for data complexity of $\text{ALCHOIF}$ with closed predicates
- reduction to an exponentially large system of linear inequalities

- $\text{NExpTime}$ upper bound for circumscribed $\text{ALCIO}$ when
  - no nesting of quantifiers
  - roles can be minimized or fixed
Some recent work

- tight NP upper bound for data complexity of $\text{ALCHOIF}$ with closed predicates
- reduction to an exponentially large system of linear inequalities

- $\text{NExpTime}$ upper bound for circumscribed $\text{ALCIO}$ when
  - no nesting of quantifiers
  - roles can be minimized or fixed
  - but using pointwise circumscription!
We let $\mathcal{I} \sim^{\cdot} \mathcal{J}$ if there exists $e \in \Delta^I$ such that:

- $A^I \setminus \{e\} = A^J \setminus \{e\}$ for all concept names $A$, and
- $r^I \cap (\Delta \times \Delta) = r^J \cap (\Delta \times \Delta)$ for all role names $r$, where $\Delta = \Delta^I \setminus \{e\}$.

We let $\mathcal{I} \preceq^{\cdot} \mathcal{J}$ if $\mathcal{I} \preceq^{\cdot} \mathcal{J}$ and $\mathcal{I} \sim^{\cdot} \mathcal{J}$. 
Definition

We let $\mathcal{I} \sim \cdot \mathcal{J}$ if there exists $e \in \Delta^\mathcal{I}$ such that:

1. $A^\mathcal{I} \setminus \{e\} = A^\mathcal{J} \setminus \{e\}$ for all concept names $A$, and
2. $r^\mathcal{I} \cap (\Delta \times \Delta) = r^\mathcal{J} \cap (\Delta \times \Delta)$ for all role names $r$, where $\Delta = \Delta^\mathcal{I} \setminus \{e\}$.

We let $\mathcal{I} \preceq \mathcal{J}$ if $\mathcal{I} \preceq \mathcal{J}$ and $\mathcal{I} \sim \cdot \mathcal{J}$.

Second-order quantification replaced by local FO checks
Sound approximation of Circ

\[ A \subseteq \exists R \quad \exists R^- \subseteq A \]
Sound approximation of Circ

\[ A \subseteq \exists R \quad \exists R^- \subseteq A \]
Limits of PW-Circumscription

Sound approximation of Circ

\[ A \subseteq \exists R \quad \exists R^{-} \subseteq A \]

- Point-wise minimization blind to some cyclic justifications
- They often coincide, especially for KR examples!
Proving the upper bound

Reduction to an exponentially large system of linear inequalities

- minimality at the individual tiles
- book-keeping needed to make sure that pw-minimal models are captured

About the restrictions

\[ B_1 \cap \cdots \cap B_{k-1} \subseteq B_k \cup \cdots \cup B_m \quad B_1 \subseteq \exists r. B_2 \quad B_1 \subseteq \forall r. B_2 \]

is not a normal form for circumscribed $\mathcal{ALCIO}!$
Reduction to an exponentially large system of linear inequalities

- minimality at the individual tiles
- book-keeping needed to make sure that pw-minimal models are captured

About the restrictions

\[ B_1 \cap \cdots \cap B_{k-1} \subseteq B_k \cup \cdots \cup B_m \quad B_1 \subseteq \exists r.B_2 \quad B_1 \subseteq \forall r.B_2 \]

is not a normal form for circumscribed \text{ALCIO}!

- undecidable for full \text{ALCIO}!
- NExpTime for pointwise circumscribed \text{ALCIO}_{d\leq 1} is tight!
Some results for DL-Lite

The same **mosaic technique** can give **NP bounds for DL-Lite**

- Satisfiability in $\text{DL-Lite}^{\text{HOF}}_{\text{Bool}}$ with a given set of **finite predicates**
- Satisfiability in **pointwise circumscribed DL-Lite**$_{\text{core}^-}$ where all predicates are varying

**Key ideas:**

- exponentially many variables
- only polynomially many inequations
- we can assume that only polynomially many variables take non-zero values
  - Eisenbrand and Shmonin 2006; Pratt-Hartmann 2008
- non-deterministically generate the inequation system and decide if a solution exists
### Global Circumscription

<table>
<thead>
<tr>
<th>DL-Lite(^H)</th>
<th>basic</th>
<th>varying roles</th>
<th>general</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-Lite(^H)</td>
<td>NL-c ≥ NP and $\leq \Sigma^p_2$</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>DL-Lite(^H)horn</td>
<td>≥ NP ≥ NP</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>DL-Lite(^H)bool</td>
<td>≥ $\Sigma^p_2$ ≥ $\Sigma^p_2$</td>
<td>undec.</td>
<td></td>
</tr>
</tbody>
</table>

### Pointwise Circumscription

<table>
<thead>
<tr>
<th>DL-Lite(^H)</th>
<th>basic</th>
<th>varying roles</th>
<th>general</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-Lite(^H)horn</td>
<td>≤ P NP-c*</td>
<td>≤ NExp(^\dagger)</td>
<td></td>
</tr>
<tr>
<td>DL-Lite(^H)bool</td>
<td>≥ NP ≥ NP</td>
<td>≤ NExp(^\dagger)</td>
<td></td>
</tr>
<tr>
<td>DL-Lite(^H)bool</td>
<td>≥ $\Sigma^p_2$ ≥ $\Sigma^p_2$</td>
<td>≤ NExp(^\dagger)</td>
<td></td>
</tr>
</tbody>
</table>

Basic Pattern: all predicates in $M \cup F$; Role-varying Patterns: all roles in $V$. 
Conclusions and Future Work

- Still working on (pointwise) circumscription
- Surprisingly little done on minimal model reasoning
- Minimality and mixing OWA and CWA are hard problems
- From intended models to preferred models
- The mosaic technique to reduce to integer equations is quite effective!
Conclusions and Future Work

- Still working on (pointwise) circumscription
- Surprisingly little done on minimal model reasoning
- Minimality and mixing OWA and CWA are hard problems
- From intended models to preferred models
- The mosaic technique to reduce to integer equations is quite effective!

New applications: SHACL, the new standard constraint language for webdata, is basically the DL $\mathcal{ALCOIQ}_{\text{reg}}$ with closed roles
Conclusions and Future Work

- Still working on (pointwise) circumscription
- Surprisingly little done on minimal model reasoning
- Minimality and mixing OWA and CWA are hard problems
- From intended models to preferred models
- The mosaic technique to reduce to integer equations is quite effective!

New applications: SHACL, the new standard constraint language for webdata, is basically the DL $\mathcal{ALCOIQ}_{reg}$ with closed roles

Combining SHACL + DLs/OWL needs minimal model reasoning
Conclusions and Future Work

- Still working on (pointwise) circumscription
- Surprisingly little done on minimal model reasoning
- **Minimality and mixing OWA and CWA are hard problems**
- From intended models to preferred models
- The mosaic technique to reduce to integer equations is quite effective!

New applications: SHACL, the new standard constraint language for webdata, is basically the DL $\mathcal{ALC}OIQ_{reg}$ with closed roles

Combining SHACL + DLs/OWL needs minimal model reasoning

The decidability of $\mathcal{ALC}OIQ_{reg}$ is still open!
Conclusions and Future Work

- Still working on (pointwise) circumscription
- Surprisingly little done on minimal model reasoning
- Minimality and mixing OWA and CWA are hard problems
- From intended models to preferred models
- The mosaic technique to reduce to integer equations is quite effective!

New applications: SHACL, the new standard constraint language for webdata, is basically the DL $\mathcal{ALCOIQ}_{\text{reg}}$ with closed roles

Combining SHACL + DLs/OWL needs minimal model reasoning

The decidability of $\mathcal{ALCOIQ}_{\text{reg}}$ is still open!

Thanks to all my co-authors and collaborators, and especially to Federica Di Stefano and Sanja Lukumbuzya. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
The recent works mentioned in the talk can be found in these papers:


For some pointers to SHACL: