An Ordered Theory Resolution
Calculus for First-order
Extensions of Description Logic

Scott Sanner (ssanner@s.toronto.edu)
Sheila McIlraith (sheila@s.toronto.edu)
University of Toronto
In a Nutshell

• **Description Logics (DLs):**
  - Decidable fragment of first-order logic (FOL)
  - Widely used for ontology modeling
  - **Caveat:** Some ontology-oriented applications need FOL expressiveness!

• **Problem: How to reason with DL+FOL?**
  - DL reasoners efficient, but limited
  - FOL theorem provers sound and complete, but inefficient for DLs (Tsarkov et al.)
  - **Can we combine both approaches?**
Outline

- **Background & Motivation**
  - State-of-the-art for DL & FOL reasoning

- **(Ordered) Theory Resolution 101**

- **Reasoning with DL-FOL**
  - Overview, difficulties with theory res.
  - (Partial) narrow theory res. & strategies
  - Soundness and completeness

- **Experimental Results** (proof-of-concept)

- **Conclusions and Future Work**
**DL/FOL Correspondence I**

- **DL is a concept-oriented logic**
  - Widely used for ontology modeling
  - Decidable fragment of FOL

<table>
<thead>
<tr>
<th>English</th>
<th>FOL</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>All CEOs are employees</td>
<td>$\forall x. \text{CEO}(x) \Rightarrow \text{Employee}(x)$</td>
<td>$\text{CEO} \sqsubseteq \text{Employee}$</td>
</tr>
<tr>
<td>An employee is a person who has a job that is a paid position</td>
<td>$\forall x. \text{Employee}(x) \equiv \text{Person}(x) \land \exists y. \text{hasJob}(x,y) \land \text{PaidPosition}(y)$</td>
<td>$\text{Employee} \equiv \text{Person} \sqcap \exists \text{hasJob}.\text{PaidPosition}$</td>
</tr>
</tbody>
</table>
• But not all ontological concepts or axioms are expressible in DL:

<table>
<thead>
<tr>
<th>English</th>
<th>FOL</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Competent-CEO is a CEO who has some skill required for their job</td>
<td>$\forall x. \text{CompetentCEO}(x) \equiv \text{CEO}(x) \land \exists y. \text{hasJob}(x,y)$ \land $\exists z. \text{requiresSkill}(y,z) \land \text{hasSkill}(x,z)$</td>
<td>Not obvious due to use of vars:</td>
</tr>
</tbody>
</table>

• How to augment DLs with FOL expr.?
Extensions of DL

- **Horn/Datalog Extensions of DL:**
  - **CARIN:** DL+Horn Rules (Halevy and Rousset)
  - **AL-LOG:** DL(ALC)+Datalog (Donini et al.)
  - **On Semantic Web:**
    - **Languages:** SWRL and RuleML
    - **Reasoners:** DL Programs (Grosos et al.), Production Rule Systems (Golbreich), DL(SHIQ)+Disj. Datalog (Motik et al.)
      ...

- **Full FOL Extensions of DL (DL-FOL)**
  - **On Semantic Web:**
    - **Languages:** FOL-SWRL, FOL-RuleML, OWL-S + SWSO
    - **Reasoners:** Theorem Proving? Hybrid?
Reasoning with DL-FOL

- **Why not use a theorem prover for DL-FOL?**
  - *Comparison of Vampire to FaCT++: (Tsarkov et al.)*
    - Vampire took more time & proved fewer queries
  - But DL reasoners alone cannot handle full FOL
  - Can we combine theorem proving and DL inf.?

- **Krypton: Augment FO res. with DL inference**
  - Given: $B \subseteq C, A(x) \lor B(x), \neg C(y) \lor D(y)$
  - Infer: $A(x) \lor D(x)$ (Why? b/c $B$ and $\neg C$ are unsat)

- **Drawback of Krypton:**
  - No conditions for removing theory axioms from KB
  - Important for efficiency, soundness/completeness
Generalizing to Theory Res.

- Theory Resolution generalized Krypton ideas for arbitrary theories
  - Any theory allowed: DL, (in)equality, intervals, ...
  - Allowed axioms of theory to be removed from KB
  - Gave conditions for soundness and completeness

- But no follow-on work appears to address theory resolution for an expressive DL:
  - What does it take to meet soundness and completeness conditions of theory resolution?
  - This is the question we want to answer.

- First, let’s review resolution and the (Ordered) Theory Resolution calculus...
First-order Resolution

• Binary Resolution Rule

Rule:
\[ C_1 \ldots C_2 \quad \sigma = \text{MGU}(L_1, L_2) \quad \text{for } L_1 \subseteq C_1, L_2 \subseteq C_2 \]
\[ \{C_1 \sigma - L_1 \sigma\} \cup \{C_2 \sigma - L_2 \sigma\} \]

Example application:
\[ P(3) \lor Q(f(x)) \quad R(y) \lor \neg Q(y) \]
\[ P(3) \lor R(f(x)) \]

• Factoring Rule

Rule:
\[ C \quad \sigma = \text{MGU}(L_1, \ldots, L_n) \quad \text{for } \{L_1, \ldots, L_n\} \subseteq C \]
\[ C_\sigma \]

Example application:
\[ P(z) \lor Q(3) \lor Q(z) \]
\[ P(3) \lor Q(3) \]
Theory Resolution

• **Theory Resolution (Stickel)**
  - **Resolve over sets of unsatisfiable subclauses, e.g.,**
    - **Given:** \( A \lor x \prec y, B \lor y \prec z, C \lor z \prec x \)
    - **Infer:** \( A \lor B \lor C \)
  - **Remove axioms of theory from KB and use theory-specific decision procedure to determine unsat!**

• **Two refinements of theory resolution:**
  - **Narrow:** Resolve over one literal per clause
  - **Partial:** Can resolve with residue “conditions”
    - **Given:** \( A \lor x \prec y, B \lor y \prec z \)
    - **Infer:** \( A \lor B \lor x \prec z \)
    - \( x \prec z \) is a valid residue if \( \{-(x \prec z), x \prec y, y \prec z\} \) is unsat
Ordered Theory Resolution

• Ordered Theory Resolution (Baumgartner)
  - Uses literal ordering restrictions to reduce search
  - Lifts from ground to non-ground case
  - How to refute non-ground literals?

• Theory Refuting Substitutions
  - With theory T, unifiers of literals L may not be unique
    • Let theory T = \{ \forall x. A(x) \implies B(x) , \forall x. A(f(g(x))) \implies B(x) \}
    • Let literals L = \{ A(w), \neg B(z) \}
    • Then CSR_T(L) = \{ \{w/z\}, \{w/f(g(z))\} \}
  - Generalize to complete set of T-refuters: CSR_T(L)
  - Require decision procedure for Find-CR_T(L)
  - If Find-CR_T(L) correct & complete (i.e., all found) for T then ordered theory res. is sound & complete
**Ordered Theory Res. Rules**

- **Ordered Factoring**
  
  Rule: \[
  \frac{C}{C_\sigma}
  \]
  
  If (1) \( \sigma \) is the most general syntactic unifier for some \( \{L_1, \ldots, L_n\} \subseteq C \), and
  
  (2) \( L_1 \sigma \) is maximal in \( C_\sigma \)

- **Ordered Narrow Theory Resolution**
  
  Rule: \[
  \frac{C_1 \ldots C_n}{\{C_1 \sigma \rightarrow L_1 \sigma\} \cup \ldots \cup \{C_n \sigma \rightarrow L_n \sigma\}}
  \]
  
  If (1) \( \sigma \in CSR_T(\{L_1, \ldots, L_n\}) \) for some \( L_1 \in C_1, \ldots, L_n \in C_n \), and
  
  (2) \( L_i \sigma \) is maximal in \( C_i \sigma \) (for \( i=1 \ldots n \))

*Note: “maximal” is w.r.t. literal ordering*
Outline

- **Background & Motivation**
  - State-of-the-art for DL & FOL reasoning

- **(Ordered) Theory Resolution 101**

- **Reasoning with DL-FOL**
  - Overview, difficulties with theory res.
  - (Partial) narrow theory res. & strategies
  - Soundness and completeness

- **Experimental Results** (proof-of-concept)

- **Conclusions and Future Work**
Theory Res. with DL-FOL

- Example DL(SHI)-FOL KB w/ Query $S(c,f(c))$

- Refutation $\bot$ found... query proved!
DL, DL’, and FOL Concepts

- DL-FOL KB axioms sorted into theories
- All recognizable SHI DL concepts and constructors sorted into DL/DL’ theories:

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL</th>
<th>DL’</th>
<th>FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic Concept</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Top Concept</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Bottom Concept</td>
<td>⊥</td>
<td>⊥</td>
<td></td>
</tr>
<tr>
<td>Negation</td>
<td>¬C</td>
<td>¬C</td>
<td></td>
</tr>
<tr>
<td>Conjunction</td>
<td>C ⋂ D</td>
<td>C ⋂ D</td>
<td></td>
</tr>
<tr>
<td>Disjunction</td>
<td>C ⋃ D</td>
<td>C ⋃ D</td>
<td></td>
</tr>
</tbody>
</table>
DL, DL’, and FOL Roles

- DL roles/restrictions redundant in DL, FOL
- $R^*/A^*$ are newly gen. role/concept names

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL</th>
<th>DL’</th>
<th>FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic Role</td>
<td>$R$</td>
<td>$R$</td>
<td>$\forall x, y. R(x,y) \equiv R^*(y,x)$</td>
</tr>
<tr>
<td>Inverse Role</td>
<td>$R^<em>, R^</em> \equiv \neg R$</td>
<td>$R^*$</td>
<td>$\forall x, y, z. R^<em>(x,y) \land R^</em>(y,z) \Rightarrow R^*(x,z)$</td>
</tr>
<tr>
<td>Transitive Role</td>
<td>$R^<em>, R^</em> \equiv R^+$</td>
<td>$R^*$</td>
<td>$\forall x, y, z. R^*(x,y)$</td>
</tr>
<tr>
<td>Exists Restriction</td>
<td>$A^<em>, A^</em> \equiv \exists R. C$</td>
<td>$A^*$</td>
<td>$\forall x. A^*(x) \equiv \exists y. R(x,y) \land C(y)$</td>
</tr>
<tr>
<td>Value Restriction</td>
<td>$A^<em>, A^</em> \equiv \forall R. C$</td>
<td>$A^*$</td>
<td>$\forall x. A^*(x) \equiv \forall y. R(x,y) \Rightarrow C(y)$</td>
</tr>
<tr>
<td>Role Filler Restriction</td>
<td>$A^*$</td>
<td>$A^*$</td>
<td>$\forall x. A^*(x) \equiv R(x,c)$</td>
</tr>
</tbody>
</table>
### DL, DL’, and FOL Axioms

- DL / FOL axioms go in respective theories
- Negated query always goes in FOL theory

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL</th>
<th>DL’</th>
<th>FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Inclusion</td>
<td>C ⊑ D</td>
<td>C ⊑ D</td>
<td></td>
</tr>
<tr>
<td>Concept Equivalence</td>
<td>C ≡ D</td>
<td>C ≡ D</td>
<td></td>
</tr>
<tr>
<td>Role Inclusion</td>
<td>R ⊑ S</td>
<td>R ⊑ S</td>
<td></td>
</tr>
<tr>
<td>Role Equivalence</td>
<td>R ≡ S</td>
<td>R ≡ S</td>
<td></td>
</tr>
<tr>
<td>Concept Assertion</td>
<td></td>
<td></td>
<td>C(a)</td>
</tr>
<tr>
<td>Role Assertion</td>
<td></td>
<td>R(a,b)</td>
<td></td>
</tr>
<tr>
<td>FOL Axiom ( \varphi )</td>
<td></td>
<td>( \varphi )</td>
<td></td>
</tr>
<tr>
<td>Query ( \varphi )</td>
<td></td>
<td></td>
<td>( \neg \varphi )</td>
</tr>
</tbody>
</table>
DL-FOL Theory Sorting Ex.

- **Given DL-FOL KB:**
  \[
  \{ \text{MSOD} \equiv \text{Male} \cap \exists \text{hasChild}^{-}.\text{Doctor} \}
  \]

- **DL Component:**
  \[
  \{ R^* \equiv \text{hasChild}^{-}, \ A^* \equiv \exists R^*.\text{Doctor}, \\
  \text{MSOD} \equiv \text{Male} \cap \ A^* \}
  \]

- **DL’ Component:**
  \[
  \{ \text{MSOD} \equiv \text{Male} \cap \ A^* \}
  \]

- **FOL Component:**
  \[
  \{ \forall x, y. \ R^*(x,y) \equiv \text{hasChild}(y,x), \\
  \forall x. \ A^*(x) \equiv \exists y. R^*(x,y) \land \text{Doctor}(y) \}
  \]
• Why have we defined a DL & DL’ theory?
• Let’s analyze theory T-refuters when T=DL
  – All Find-CSR_T(L) procedures must return same T-refuters... just use resolution here

• Example:
  – Given DL-FOL Components:
    DL: \{ \exists S. \forall R. A \sqsubseteq B \}  
    FOL: \{ \exists w. S(c,w) \land (\forall z. R(w,z) \Rightarrow A(z)), 
            \neg B(c) \} 
  – L= \{ \neg B(c), S(c,d), \neg R(d,z) \} 
  – CSR_T(L)= \{ \{z/f(c,d)\} \}  

• Theorem: Even if literal set L contains no function symbols, when T=DL, CSR_T(L) may contain arbitrarily large function symbols.
Algorithm for Find-$\text{CSR}_{DL'}(L)$

- **Solution:** Let $T=DL'$ instead of $T=DL$
  - $T$-refuters for $DL'$ are limited to standard MGUs of literals $L$
  - Why? Because source of function symbols has been removed from $DL$ and put in FOL.

- **This suggests a Find-$\text{CSR}_{DL'}(L)$ algorithm:**
  1. Return MGUs for all *syntactically complementary* literals
  2. Return MGUs of all *dyadic* literals that are unsatisfiable w.r.t. role hierarchy
  3. Return MGUs of all *monadic* literals that are unsatisfiable using the $DL$ reasoner
Soundness and Completeness

- **Find-CSR_{DL'}(L) clearly correct:**
  - Easy to verify all substitutions lead to unsat of L
- **Find-CSR_{DL'}(L) completeness a little harder:**
  - Can consider (1) monadic, (2) dyadic, and (3) standard non-theory syntactic complementary
  - No interaction b/w axioms of (1), (2)
  - DL handles (1), transitive closure of role hierarchy covers (2), and (3) is just standard res.

- **Proves soundness/completeness of Theory Resolution using Find-CSR_{DL'}(L) for DL'+FOL**
- **DL only adds redundancy to DL’, thereby retaining completeness for full DL+FOL**
Partial Narrow Theory Res.

• Problem with narrow (N) OTRC: non-binary resolution of \( k \) clauses is difficult
  – May have to select literals from all \( k \) clauses!
  – Combinatorially explosive number of resolutions
  – Must systematically try all combos for completeness

• Prefer to do binary resolution if possible...

• Suggests partial narrow (PN) OTRC

\[
\text{FOL Component}
\begin{align*}
\{L_{1,1}, L_{1,2}, L_{1,3}\} \\
\{L_{2,1}, L_{2,2}\} \\
\ldots \\
\{L_{k,1}, L_{k,2}, L_{k,3}\}
\end{align*}
\]
Partial Narrow OTRC

- **Partial Narrow Ordered Theory Resolution**

  Rule:

  \[
  \frac{C_1, C_2}{\{C_1 \sigma - L_1 \sigma\} \cup \{C_2 \sigma - L_2 \sigma\} \cup (L_1 \cap L_2) \sigma}
  \]

  If \((1) \sigma \in \text{MGU}({L_1, \ldots, L_2})\) (term-only MGU) for some \(L_1 \in C_1, L_2 \in C_2,\) and \((2) L_i \sigma\) is maximal in \(C_i \sigma\) (for \(i = 1, 2\))

- **Creates a compound residue literal**
  - If \(L_1 \cap L_2\) is unsat. then remove literal
  - Else \(L_1 \cap L_2\) may be refuted in another res.

- **PN-OTRC is sound and complete**
  - Have to give residue literals proper prec.
  - Then easy to show can simulate N-OTRC
• Problem: Lots of residue!
• Introduce age-weight strategy (Otter)
  – At every step, choose a clause to resolve with all others (incl. self)
  – Keep two clause queues
    • A FIFO queue that orders clauses by age
    • A priority queue that orders clauses by weight
  – For every $a + w$ clauses chosen, select $a$ from age queue and $w$ from weight queue
    • Complete for $a > 0$
• Assign clause weight corresponding to residue size – avoid large residue!
Ordering Heuristics

- Can also exploit DL taxonomy structure in literal ordering:
  - Prefer-Deep: Prioritize literals near bottom of DL taxonomy
  - Prefer-Deep focuses on hardest to refute literals first... should be more efficient

- Preference for Shallow
  - Prioritize literals near top of DL taxonomy

- Intuition: Deeper in taxonomy, fewer and more specific inferences
Outline

• Background & Motivation
  – State-of-the-art for DL & FOL reasoning

• (Ordered) Theory Resolution 101

• Reasoning with DL-FOL
  – Overview, difficulties with theory res.
  – (Partial) narrow theory res. & strategies
  – Soundness and completeness

• Experimental Results (proof-of-concept)

• Conclusions and Future Work
## Experimental Results

### Spatial Subset of OpenCyc KB

<table>
<thead>
<tr>
<th>Reasoner</th>
<th>#Successes</th>
<th>Avg. Clauses Generated</th>
<th>Avg. Resolution Proof Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vampire v8</td>
<td>25/25</td>
<td>137</td>
<td>10.5</td>
</tr>
<tr>
<td>Otter v3.3</td>
<td>25/25</td>
<td>603</td>
<td>9.6</td>
</tr>
<tr>
<td>SPASS v2.1</td>
<td>25/25</td>
<td>4763</td>
<td>9.4</td>
</tr>
<tr>
<td>DL-FOL (FOL Translation only)</td>
<td>5/25</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>DL-FOL (PN-OTR, Prefer-Shallow)</td>
<td>25/25</td>
<td>346</td>
<td>7.3</td>
</tr>
<tr>
<td>DL-FOL (PN-OTR, Prefer-Deep)</td>
<td>25/25</td>
<td>147</td>
<td>7.3</td>
</tr>
<tr>
<td>DL-FOL (Hyperres-OTR*, Prefer-Deep)</td>
<td>25/25</td>
<td>86</td>
<td>2.4</td>
</tr>
</tbody>
</table>

*Incomplete?*
Summary & Conclusions

• **Theory resolution for the DL(SHI) + FOL:**
  - Identified potential caveats and worked around them in the theory definition, inf. rules, & strategies
  - Proved soundness and completeness
  - Empirically: yielded shorter res. proof lengths than non-theory res.
  - Promising approach for further research

• **Pro:** Leverage efficiency of DL reasoners in first-order inference

• **Con:** Currently relies heavily on heuristics to guide search
Future Work

• **Augment factoring with theory implication to avoid retention of tautologies?**

• **Use decidable res. proc. for DL?** (Motik et al.)
  – Will yield complex T-refuters at no extra cost!
  – Allows full separation of DL, FOL KBs
  – Should extend to equality and SHIQ/SHOIQ (cardinality restrictions!)

• **Ordered theory resolution w/ selection?**
  – Selection functions are a powerful saturation technique – carefully select literals to prevent resolution inferences
  – Retains completeness with forward/backward subsumption deletion – generalize to theory res.?