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

Scott Sanner (ssanner@cs.toronto.edu)

Sheila McIlraith (sheila@cs.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

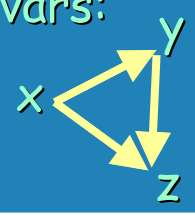
DL/FOL Correspondence I

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

English	FOL	DL
All CEOs are employees	$\forall x.\ CEO(x) \Rightarrow Employee(x)$	$CEO \sqsubseteq Employee$
An employee is a person who has a job that is a paid position	$\forall x.\ Employee(x) \equiv Person(x) \wedge \exists y.\ hasJob(x,y) \wedge PaidPosition(y)$	$Employee \equiv Person \sqcap \exists hasJob.PaidPosition$

DL/FOL Correspondence II

- But not all ontological concepts or axioms are expressible in DL:

English	FOL	DL
A Competent-CEO is a CEO who has some skill required for their job	$\forall x.\ CompetentCEO\ (x) \equiv CEO(x) \wedge \exists y.\ hasJob(x,y) \wedge \exists z.\ requiresSkill(y,z) \wedge hasSkill(x,z)$	Not obvious due to use of vars: 

- 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 (Grosof *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: $B \sqsubseteq C,\ A(x) \vee B(x),\ \neg C(y) \vee D(y)$
  - Infer: $A(x) \vee D(x)$ (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:

$$\frac{C_1 \; \dots \; C_2}{\{C_1\sigma - L_1\sigma\} \cup \{C_2\,\sigma - L_2\sigma\}} \quad \begin{array}{l} \sigma = MGU(L_1, L_2) \\ \text{for } L_1 \subseteq C_1, L_2 \subseteq C_2 \end{array}$$

Example application:

$$\frac{P(3) \vee Q(f(x)) \quad R(y) \vee \neg Q(y)}{P(3) \vee R(f(x))}$$

- Factoring Rule

Rule:

$$\frac{C}{C\sigma} \quad \begin{array}{l} \sigma = MGU(L_1, \dots, L_n) \\ \text{for } \{L_1, \dots, L_n\} \subseteq C \end{array}$$

Example application:

$$\frac{P(z) \vee Q(3) \vee Q(z)}{P(3) \vee Q(3)}$$

Theory Resolution

- Theory Resolution (Stickel)
  - Resolve over sets of unsatisfiable subclauses, e.g.,
    - Given: $\underline{A \vee x < y,\ B \vee y < z,\ C \vee z < x}$
    - Infer: $A \vee B \vee C$
  - 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: $\underline{A \vee x < y,\ B \vee y < z}$
    - Infer: $A \vee B \vee x < z$
  - $x < z$ is a valid residue if $\{\neg(x < z), x < y, y < z\}$ is unsat

Ordered Theory Resolution

- Ordered Theory Resolution (Baumgartner)
  - Uses literal ordering restrictions to reduce search
  - Lifts from ground to non-ground case
  - <u>How to refute non-ground literals?</u>
- Theory Refuting Substitutions
  - W/ theory T, unifiers of literals L may not be unique
    - Let theory T = $\{ \forall x.\ A(x) \Rightarrow B(x)\ ,\ \forall x.\ A(f(g(x))) \Rightarrow B(x) \}$
    - Let literals L = $\{ A(w), \neg B(z) \}$
    - Then $CSR_T(L) = \{ \{w/z\}, \{w/f(g(z))\} \}$
  - <u>Generalize to complete set of *T-refuters*: $CSR_T(L)$</u>
  - Require *decision procedure* for Find-$CSR_T(L)$
  - If Find-$CSR_T(L)$ 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) σ 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:

$$\frac{C_1 \ \ldots \ C_n}{\{C_1\sigma - L_1\sigma\} \cup \ldots \cup \{C_n\sigma - L_n\sigma\}}$$

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 $i=1\ldots n$)

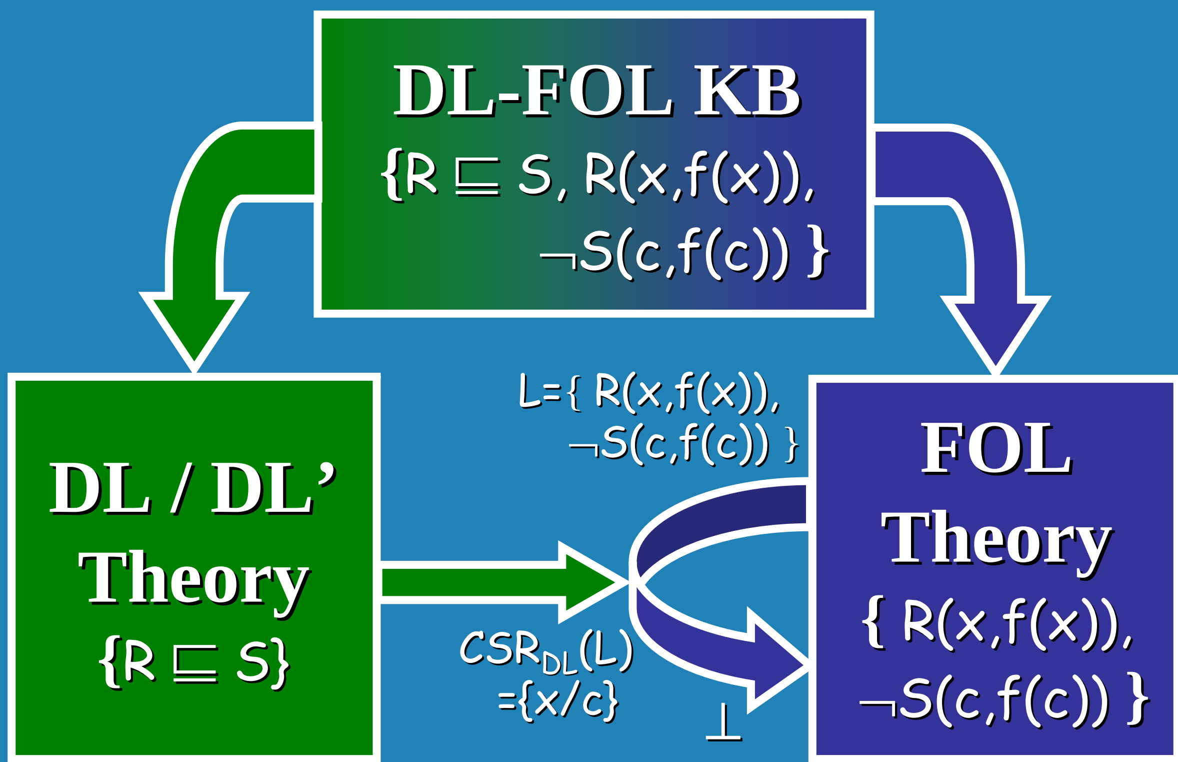
Note: "maximal" is w.r.t. literal ordering

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 ⊥ found… query proved!

DL, DL’, and FOL Concepts

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

Constructor	DL	DL’	FOL
Atomic Concept	A	A	
Top Concept	$\top$	$\top$	
Bottom Concept	$\bot$	$\bot$	
Negation	$\neg C$	$\neg C$	
Conjunction	$C \sqcap D$	$C \sqcap D$	
Disjunction	$C \sqcup D$	$C \sqcup D$	

DL, DL', and FOL Roles

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

Constructor	DL	DL'	FOL
Atomic Role	R	R	
Inverse Role	R^*, $R^* \equiv R^-$	R^*	$\forall x,y.\ R(x,y) \equiv R^*(y,x)$
Transitive Role	R^*, $R^* \equiv R^+$	R^*	$\forall x,y,z.\ R^*(x,y) \wedge R^*(y,z) \Rightarrow R^*(x,z)$
Exists Restriction	A^*, $A^* \equiv \exists R.C$	A^*	$\forall x.\ A^*(x) \equiv \exists y.R(x,y) \wedge C(y)$
Value Restriction	A^*, $A^* \equiv \forall R.C$	A^*	$\forall x.\ A^*(x) \equiv \forall y.R(x,y) \Rightarrow C(y)$
Role Filler Restriction	A^*	A^*	$\forall x.\ A^*(x) \equiv R(x,c)$

DL, DL', and FOL Axioms

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

Constructor	DL	DL'	FOL
Concept Inclusion	$C \sqsubseteq D$	$C \sqsubseteq D$	
Concept Equivalence	$C \equiv D$	$C \equiv D$	
Role Inclusion	$R \sqsubseteq S$	$R \sqsubseteq S$	
Role Equivalence	$R \equiv S$	$R \equiv S$	
Concept Assertion			$C(a)$
Role Assertion			$R(a,b)$
FOL Axiom φ			φ
Query φ			$\neg\varphi$

DL-FOL Theory Sorting Ex.

- Given DL-FOL KB:
  $\{ MSOD \equiv Male \sqcap \exists hasChild^{-}.Doctor \}$
- DL Component:
  $\{ R^* \equiv hasChild^{-}, A^* \equiv \exists R^*.Doctor,$
  $MSOD \equiv Male \sqcap A^* \}$
- DL' Component:
  $\{ MSOD \equiv Male \sqcap A^* \}$
- FOL Component:
  $\{ \forall x,y.\ R^*(x,y) \equiv hasChild(y,x),$
  $\forall x.\ A^*(x) \equiv \exists y.R^*(x,y) \wedge Doctor(y) \}$

Difficulties of Find-CSR_{DL}(L)

- 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: {$\exists w.\ S(c,w) \wedge (\forall z.\ R(w,z) \Rightarrow A(z))$, $\neg B(c)$ }
  - L= {$\neg B(c)$, $S(c,d)$, $\neg R(d,z)$ }
  - CSR_T(L)= { {z/f(c,d)} } ← **CSR_T(L) for T=DL contains fn symbol!**

- Theorem: Even if literal set L contains no function symbols, when <u>T=DL</u>, CSR_T(L) may contain arbitrarily large function symbols.

Algorithm for Find-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-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

FOL Component

$\{L_{1,1}, L_{1,2}, \mathbf{L_{1,3}}\}$

$\{\mathbf{L_{2,1}}, L_{2,2}\}$

...

$\{L_{k,1}, \mathbf{L_{k,2}}, L_{k,3}\}$

- Prefer to do binary resolution if possible…
- Suggests *partial* narrow (PN) OTRC

Partial Narrow OTRC

- Partial Narrow Ordered Theory Resolution

Rule:

$$\frac{C_1,\ C_2}{\{C_1\sigma - L_1\sigma\} \cup \{C_2\,\sigma - L_2\sigma\} \cup (L_1 \sqcap L_2)\,\sigma}$$

If (1) $\sigma \in MGU(\{L_1,\ldots,L_2\})$ (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 $L_1 \sqcap L_2$ is unsat. then remove literal
  - Else $L_1 \sqcap L_2$ 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

Age-Weight Strategy

- 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-Shallow: Prioritize literals near top of DL taxonomy
  - Prefer-Deep: Prioritize literals near bottom of DL taxonomy
- Intuition: Deeper in taxonomy, fewer and more specific inferences
  - Prefer-Deep focuses on hardest to refute literals first... should be more efficient

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

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

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.?