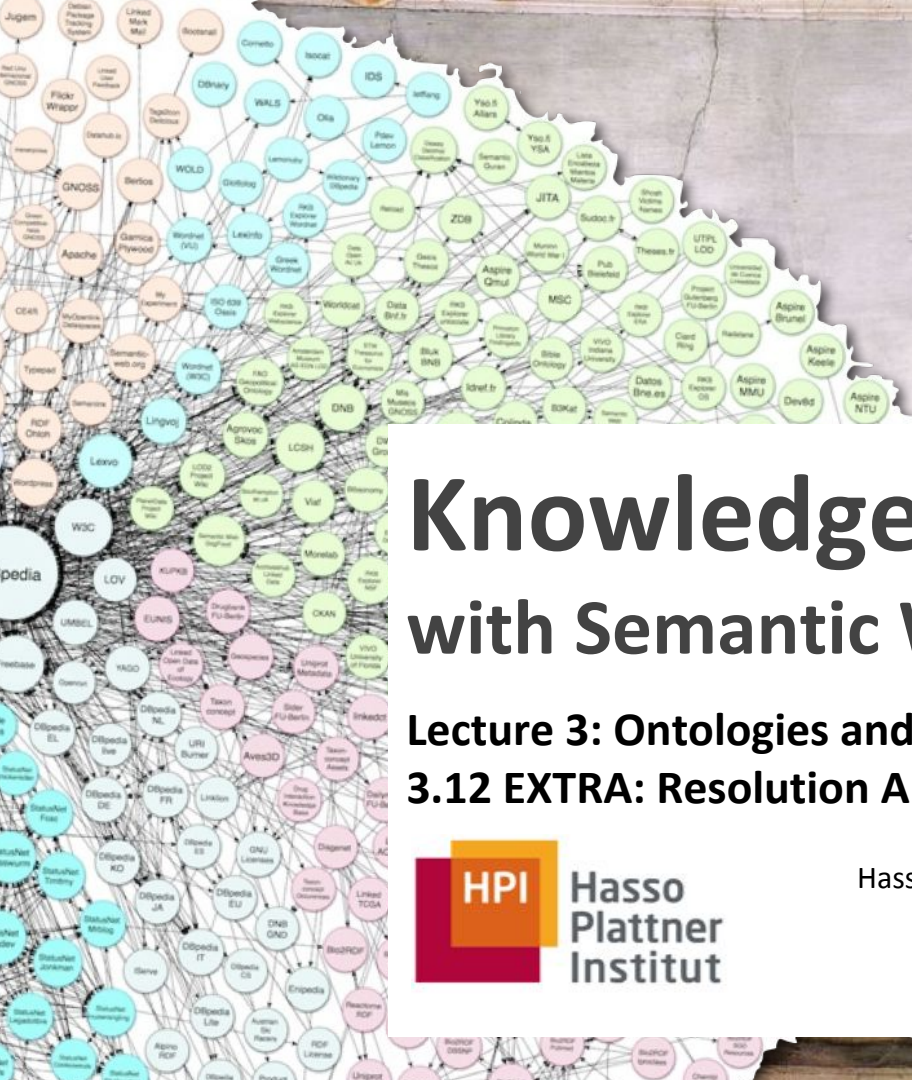




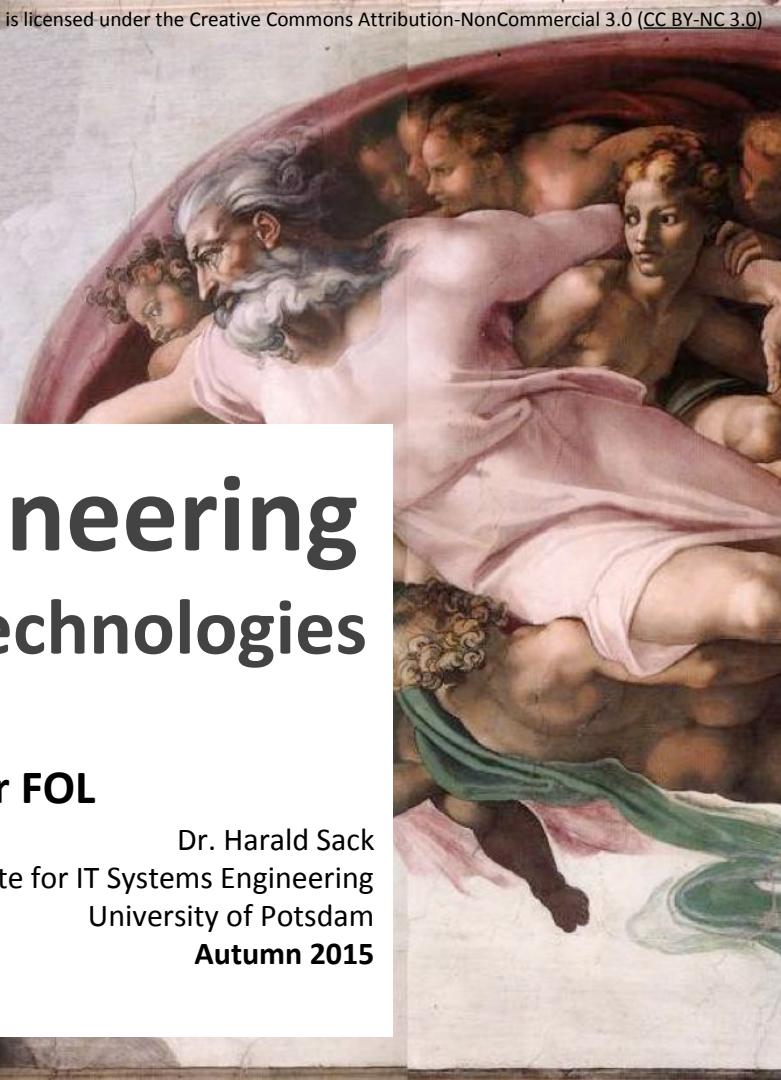
Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic

3.12 EXTRA: Resolution Algorithm for FOL



Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015



Resolution

- **Resolution** is based on simple deduction rules and is a special form of enumeration
- Instead of proving that a formula is a tautology, it deduces a logical contradiction from its negation

Theory $\{F_1, \dots, F_n\}$ with F_θ as logical Consequence: $\{F_1, \dots, F_n\} \models F_\theta$

$F_1 \wedge \dots \wedge F_n \rightarrow F_\theta$ is a tautology

$\neg(F_1 \wedge \dots \wedge F_n \rightarrow F_\theta)$ is a contradiction

$G_1 \wedge \dots \wedge G_k$ is a contradiction

- The resolution procedure allows the entailment of a contradiction from $G_1 \wedge \dots \wedge G_k$.

Resolution for PL

- If $(p_1 \vee \dots \vee p_k \vee \mathbf{p} \vee \neg q_1 \vee \dots \vee \neg q_l) \wedge (r_1 \vee \dots \vee r_m \vee \mathbf{\neg p} \vee \neg s_1 \vee \dots \vee \neg s_n)$ is true, then:
- One of both $\mathbf{p}$, $\mathbf{\neg p}$ has to be wrong.

- Therefore: One of the other Literals must be true, i.e.

$p_1 \vee \dots \vee p_k \vee \neg q_1 \vee \dots \vee \neg q_l \mathbf{\vee} r_1 \vee \dots \vee r_m \vee \neg s_1 \vee \dots \vee \neg s_n$
must be true.

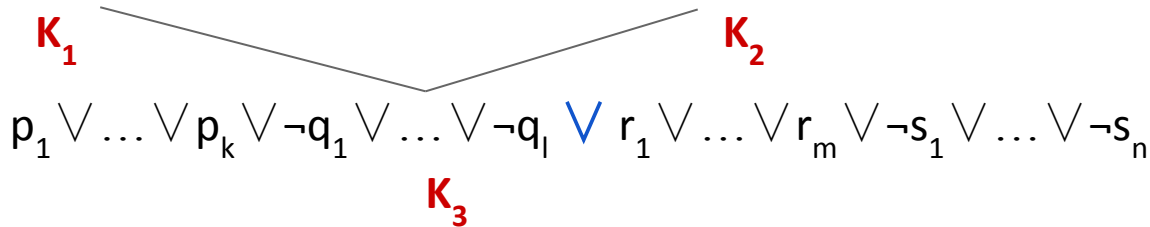
- Therefore:

If $p_1 \vee \dots \vee p_k \vee \neg q_1 \vee \dots \vee \neg q_l \vee r_1 \vee \dots \vee r_m \vee \neg s_1 \vee \dots \vee \neg s_n$ is a **contradiction**,
then

$(p_1 \vee \dots \vee p_k \vee \mathbf{p} \vee \neg q_1 \vee \dots \vee \neg q_l) \wedge (r_1 \vee \dots \vee r_m \vee \mathbf{\neg p} \vee \neg s_1 \vee \dots \vee \neg s_n)$ is **also a contradiction**.

Resolution for PL

- $(p_1 \vee \dots \vee p_k \vee \mathbf{p} \vee \neg q_1 \vee \dots \vee \neg q_l) \wedge (r_1 \vee \dots \vee r_m \vee \neg \mathbf{p} \vee \neg s_1 \vee \dots \vee \neg s_n) \#$



- two clauses $\mathbf{K}_1$ and $\mathbf{K}_2$ are transformed into a new one $\mathbf{K}_3$
- End of the resolution procedure:
 - If clauses are resolved that consist only of an atom and the negated atom, then a new „empty clause“ $\perp$ can be resolved.

Resolution for PL

How to deduce a contradiction from a set M of clauses:

1. Select two clauses from M and create a new clause K via a resolution step.
 2. If $K = \perp$, then a contradiction has been found.
 3. If $K \neq \perp$, then K is added to the set M , continue with step 1.
- the Resolution Calculus (for propositional logic) is correct and complete

Resolution for PL

- How to proof that a formula ψ is a logical consequence of a Knowledge Base (Theory) Φ of clauses, i.e. $\Phi \models \psi$?

1. Compute the negation of $\Phi \rightarrow \psi$, which is a contradiction: $\neg(\Phi \rightarrow \psi) \equiv \Phi \wedge \neg\psi$
2. Determine the **clausal form** of $\Phi \wedge \neg\psi$
3. Select two clauses from $\Phi \wedge \neg\psi$ and create a new clause K via a **resolution** step.
4. If $K = \perp$, then a contradiction has been found. It holds that $\Phi \models \psi$. q.e.d.
5. If $K \neq \perp$, K is added to the set $\Phi \wedge \neg\psi$, continue with step 3.

Resolution for PL - Example

- If it rains, Jane brings her umbrella $r \rightarrow u$
- If Jane has an umbrella, she doesn't get wet $u \rightarrow \neg w$
- If it doesn't rain, Jane doesn't get wet $\neg r \rightarrow \neg w$

knowledge
base
 Φ

- Now, prove that Jane doesn't get wet $\neg w$

formula
 ψ

Resolution for PL - Example

1. Transform knowledge base into clausal form

- $r \rightarrow u = \neg r \vee u = \{\neg r, u\}$
- $u \rightarrow \neg w = \neg u \vee \neg w = \{\neg u, \neg w\}$
- $\neg r \rightarrow \neg w = r \vee \neg w = \{r, \neg w\}$

knowledge
base
 Φ

2. Add negated formula to knowledge base

- $\neg \neg w = w$

formula
 ψ

1. $\{\neg r, u\}$
2. $\{\neg u, \neg w\}$
3. $\{r, \neg w\}$
4. $\{w\}$

new
knowledge
base
 $\Phi \wedge \neg \psi$

Resolution for PL - Example

3. Start resolution

1. $\{\neg r, u\}$
2. $\{\neg u, \neg w\}$
3. $\{r, \neg w\}$
4. $\{w\}$

new
knowledge
base
 $\Phi \wedge \neg \psi$

$$\begin{array}{c} (3,4) \quad \{r, \neg w\} \\ \quad \quad \{w\} \\ \hline 5. \quad \quad \{r\} \end{array}$$

Resolution for PL - Example

3. Continue resolution

1. $\{\neg r, u\}$
2. $\{\neg u, \neg w\}$
3. $\{r, \neg w\}$
4. $\{w\}$
5. $\{r\}$

new
knowledge
base
 $\Phi \wedge \neg \psi$

$$\begin{array}{c} (2,4) \quad \{\neg u, \neg w\} \\ \quad \{w\} \\ \hline 6. \quad \{\neg u\} \end{array}$$

Resolution for PL - Example

3. Continue resolution

1. $\{\neg r, u\}$
2. $\{\neg u, \neg w\}$
3. $\{r, \neg w\}$
4. $\{w\}$
5. $\{r\}$
6. $\{\neg u\}$

new
knowledge
base
 $\Phi \wedge \neg \psi$

$$\begin{array}{c} (1, 5) \quad \{\neg r, u\} \\ \quad \quad \{r\} \\ \hline 7. \quad \quad \{u\} \end{array}$$

Resolution for PL - Example

3. Continue resolution

1. $\{\neg r, u\}$
2. $\{\neg u, \neg w\}$
3. $\{r, \neg w\}$
4. $\{w\}$
5. $\{r\}$
6. $\{\neg u\}$
7. $\{u\}$

new
knowledge
base
 $\Phi \wedge \neg \psi$

$$\begin{array}{c} (6,7) \quad \{\neg u\} \\ \quad \quad \{u\} \\ \hline 7. \quad \quad \{\perp\} \end{array}$$

4. We have found a contradiction in
 $\Phi \wedge \neg \psi$,
therefore it holds that $\Phi \models \psi$, q.e.d.

Resolution for FOL

- For resolution in First Order Logic additional variable bindings have to be considered with the help of **Substitutions**
- e.g. $(p(X, f(Y)) \vee q(f(X), Y)) \quad (\neg p(a, Z) \vee r(Z))$

Resolution with $[X/a, Z/f(Y)]$ results in

$$(q(f(a), Y) \vee r(f(Y))).$$

Unification of Terms

- Given: Literals L_1, L_2
- Wanted: Variable substitution σ applied on L_1 and L_2 results in: $L_1\sigma = L_2\sigma$
- If there is such a variable substitution σ , then σ is called **Unifier of L_1 und L_2** .

Unification Algorithm

- Given: Literals L_1, L_2
 - Wanted: Unifier σ of L_1 and L_2
1. L_1 and L_2 are **Constants**: only unifiable, if $L_1 = L_2$.
 2. L_1 is a **Variable** and L_2 an arbitrary Term:
unifiable, if for Variable L_1 the Term L_2 can be substituted and
Variable L_1 does not occur in L_2 .
 3. L_1 and L_2 are **Predicates** or **Functions**
 $PL_1(s_1, \dots, s_m)$ and $PL_2(t_1, \dots, t_n)$:
unifiable, if
 - a. $PL_1 = PL_2$ or
 - b. $n=m$ and all terms s_i are unifiable with a term t_i

Examples for Unification

L_1	L_2	σ
$p(X, X)$	$p(a, a)$	$[X/a]$
$p(X, X)$	$p(a, b)$	n.a.
$p(X, Y)$	$p(a, b)$	$[X/a, Y/b]$
$p(X, Y)$	$p(a, a)$	$[X/a, Y/a]$
$p(f(X), b)$	$p(f(c), Z)$	$[X/c, Z/b]$
$p(X, f(X))$	$p(Y, Z)$	$[X/Y, Z/f(Y)]$
$p(X, f(X))$	$p(Y, Y)$	n.a.

Resolution for FOL - Example

Terminological Knowledge (TBox)

$$(\forall X) (\text{human}(X) \rightarrow (\exists Y) \text{parent_of}(Y, X))$$
$$(\forall X) (\text{orphan}(X) \leftrightarrow (\text{human}(X) \wedge \neg (\exists Y) (\text{parent_of}(Y, X) \wedge \text{alive}(Y))))$$

Assertional Knowledge (ABox)

$\text{orphan}(\text{harrypotter})$

$\text{parent_of}(\text{jamespotter}, \text{harrypotter})$

Can we deduce $\neg \text{alive}(\text{jamespotter})$?

Resolution for FOL - Example

We have to proof that:

```
(( $\forall X$ ) (human(X) $\rightarrow$ ( $\exists Y$ )parent_of(Y,X))  
   $\wedge$  ( $\forall X$ )(orphan(X) $\leftrightarrow$ (human(X) $\wedge \neg$ ( $\exists Y$ )(parent_of(Y,X) $\wedge$ alive  
(Y)))  
   $\wedge$  orphan(harrypotter)  
   $\wedge$  parent_of(jamespotter,harrypotter))  
   $\rightarrow$   $\neg$ alive(jamespotter))
```

...is a tautology

Resolution for FOL - Example

We have to proof that:

$$\begin{aligned} &\neg((\forall X) (\text{human}(X) \rightarrow (\exists Y) \text{parent_of}(Y, X)) \\ &\quad \wedge (\forall X) (\text{orphan}(X) \leftrightarrow (\text{human}(X) \wedge \neg(\exists Y) (\text{parent_of}(Y, X) \wedge \text{alive} \\ &\quad (Y)))) \\ &\quad \wedge \text{orphan}(\text{harrypotter}) \\ &\quad \wedge \text{parent_of}(\text{jamespotter}, \text{harrypotter})) \\ &\rightarrow \neg \text{alive}(\text{jamespotter})) \end{aligned}$$

...is a **contradiction**

Resolution for FOL - Example

Prenex Normalform

```
( $\forall X$ )( $\exists Y$ )( $\forall X1$ )( $\forall Y1$ )( $\forall X2$ )( $\exists Y2$ )  
(( $\neg$ human( $X$ )  $\vee$  parent_of( $Y$ , $X$ ))  
 $\wedge$  ( $\neg$ orphan( $X1$ )  $\vee$  (human( $X1$ )  $\wedge$  ( $\neg$ parent_of( $Y1$ , $X1$ )  $\vee$   $\neg$ alive( $Y1$ )))  
 $\wedge$  (orphan( $X2$ )  $\vee$  ( $\neg$ human( $X2$ )  $\vee$  (parent_of( $Y2$ , $X2$ )  $\wedge$  alive( $Y2$ )))  
 $\wedge$  orphan(harrypotter)  
 $\wedge$  parent_of(jamespotter,harrypotter))  
 $\wedge$  alive(jamespotter))
```

Resolution for FOL - Example

Conjunctive Normalform

```
(¬human(X) ∨ parent_of(f(X),X))  
∧ (¬orphan(X1) ∨ human(X1))  
∧ (¬orphan(X1) ∨ ¬parent_of(Y1,X1) ∨ ¬alive(Y1))  
∧ (orphan(X2) ∨ ¬human(X2) ∨ parent_of(g(X,X1,Y1,X2),X2))  
∧ (orphan(X2) ∨ ¬human(X2) ∨ alive(g(X,X1,Y1,X2)))  
∧ orphan(harrypotter)  
∧ parent_of(jamespotter,harrypotter)  
∧ alive(jamespotter)
```

Resolution for FOL - Example

Knowledge Base:

1. $\{\neg\text{human}(X), \text{parent_of}(f(X), X)\}$
2. $\{(\neg\text{orphan}(X1), \text{human}(X1))\}$
3. $\{\neg\text{orphan}(X1), \neg\text{parent_of}(Y1, X1), \neg\text{alive}(Y1)\}$
4. $\{(\text{orphan}(X2), \neg\text{human}(X2), \text{parent_of}(g(X, X1, Y1, X2), X2))\}$
5. $\{\text{orphan}(X2), \neg\text{human}(X2), \text{alive}(g(X, X1, Y1, X2))\}$
6. $\{\text{orphan}(\text{harrypotter})\}$
7. $\{\text{parent_of}(\text{jamespotter}, \text{harrypotter})\}$
8. $\{\text{alive}(\text{jamespotter})\}$

Entailed Clauses:

9. $\{\neg\text{orphan}(\text{harrypotter}), \neg\text{alive}(\text{jamespotter})\}$ (3,7) $[X1/\text{harrypotter}, Y1/\text{jamespotter}]$

Resolution for FOL - Example

Knowledge Base:

1. $\{\neg\text{human}(X), \text{parent_of}(f(X), X)\}$
2. $\{(\neg\text{orphan}(X1), \text{human}(X1))\}$
3. $\{\neg\text{orphan}(X1), \neg\text{parent_of}(Y1, X1), \neg\text{alive}(Y1))\}$
4. $\{(\text{orphan}(X2), \neg\text{human}(X2), \text{parent_of}(g(X, X1, Y1, X2), X2))\}$
5. $\{\text{orphan}(X2), \neg\text{human}(X2), \text{alive}(g(X, X1, Y1, X2))\}$
6. $\{\text{orphan}(\text{harrypotter})\}$
7. $\{\text{parent_of}(\text{jamespotter}, \text{harrypotter})\}$
8. $\{\text{alive}(\text{jamespotter})\}$

Entailed Clauses:

9. $\{\neg\text{orphan}(\text{harrypotter}), \neg\text{alive}(\text{jamespotter})\}$ (3,7) $[X1/\text{harrypotter}, Y1/\text{jamespotter}]$
10. $\{\neg\text{orphan}(\text{harrypotter})\}$ (8,9)

Resolution for FOL - Example

Knowledge Base:

1. $\{\neg\text{human}(X), \text{parent_of}(f(X), X)\}$
2. $\{(\neg\text{orphan}(X1), \text{human}(X1))\}$
3. $\{\neg\text{orphan}(X1), \neg\text{parent_of}(Y1, X1), \neg\text{alive}(Y1))\}$
4. $\{(\text{orphan}(X2), \neg\text{human}(X2), \text{parent_of}(g(X, X1, Y1, X2), X2))\}$
5. $\{\text{orphan}(X2), \neg\text{human}(X2), \text{alive}(g(X, X1, Y1, X2))\}$
6. **$\{\text{orphan}(\text{harrypotter})\}$**
7. $\{\text{parent_of}(\text{jamespotter}, \text{harrypotter})\}$
8. $\{\text{alive}(\text{jamespotter})\}$

Entailed Clauses:

9. $\{\neg\text{orphan}(\text{harrypotter}), \neg\text{alive}(\text{jamespotter})\}$ (3,7) $[X1/\text{harrypotter}, Y1/\text{jamespotter}]$
10. **$\{\neg\text{orphan}(\text{harrypotter})\}$** (8,9)
11. $\perp$ (6,10)

Completeness of Refutation

- If resolution is applied to a contradictory set of clauses, then there exists a **finite number of resolution steps** to detect the contradiction.
- The number n of necessary steps can be very large (not efficient)
- Resolution in FOL is **undecidable**
 - If the set of clauses is not contradictory, then the termination of the resolution is not guaranteed.



13: EXTRA - A Brief History of Ontology

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic

Some Things that FOL Can't Do...

- **Quantification over Properties**
 - *If John is content, then he has at least one thing in common with Paul.*
 - *Paul has all the properties of an optimist.*
- **Predicate adverbials**
 - *John is reading quickly*
- **Predicate adverbial modifiers**
 - *John is reading very quickly*
- **Relative adjective**
 - *John is a small student*
- **Relative adjective modifier**
 - *John is a terribly small student*
- **Modal operators**
 - *It is possible that John is a philosopher*
 - *It is necessary that John is a philosopher*