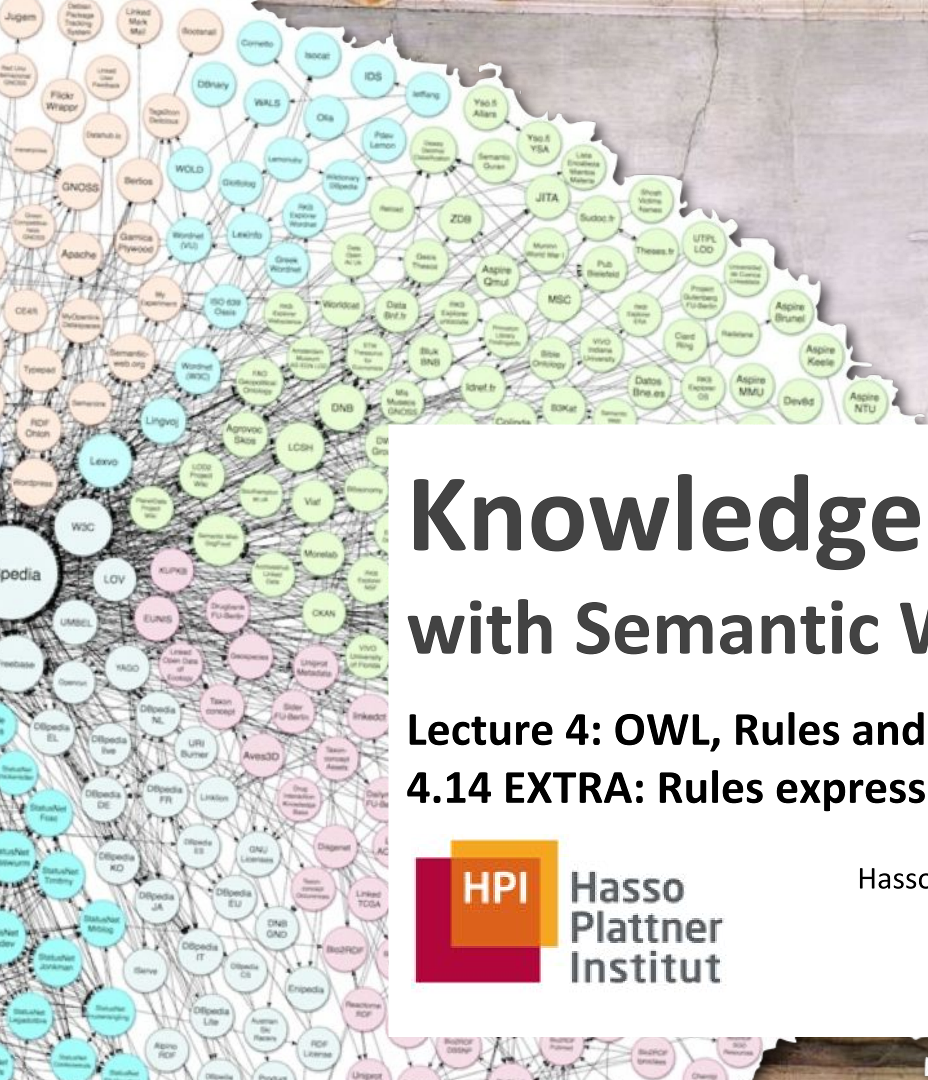
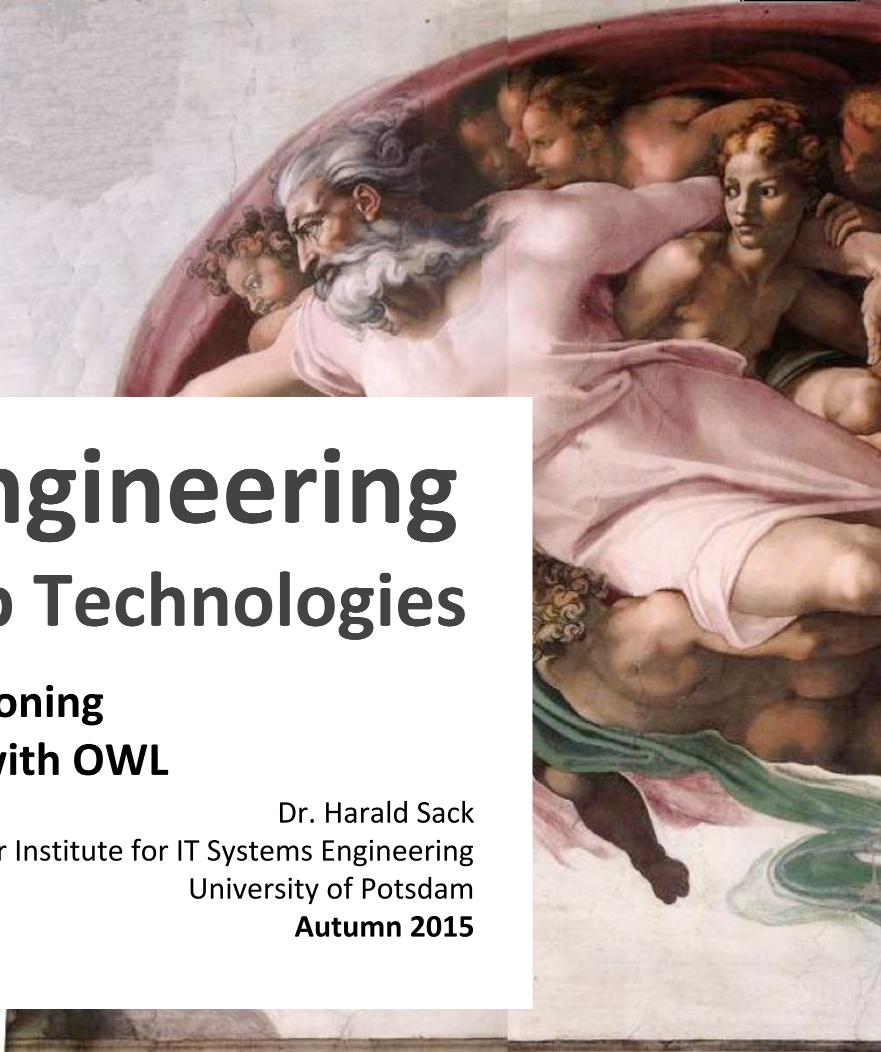




Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules and Reasoning

4.14 EXTRA: Rules expressible with OWL

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

Autumn 2015

Decidable Fragments of SWRL

- Which SWRL knowledge bases enable **complete inference algorithms**?
    - All SWRL knowledge bases that exist only of OWL / OWL2 axioms
    - All SWRL knowledge bases that exist only of sets of DATALOG programs
    - Every static finite class of SWRL knowledge bases

- Are there more SWRL fragments that are decidable?
    1. **Description Logics Rules**
       SWRL Rules that can already be expressed with OWL2
    2. **DL-safe Rules**
       Restriction of SWRL Rules via variable assignment constraints

(1) Description Logics Rules

- **General Idea**:
  Find out, which rules can be expressed via OWL2

- Simple Rules in OWL2

  Class1 $\sqsubseteq$ Class2

  Property1 $\sqsubseteq$ Property2

- become

```
Class1(x)          →  Class2(x)
Property1(x,y)     →  Property2(x,y)
```

Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. 2008. Description Logic Rules. ECAI 2008. IOS Press, Amsterdam, The Netherlands, The Netherlands, 80-84.

(1) Description Logics Rules

- Some classes can be fragmented into rules (1/2)

$A \sqcap B \sqsubseteq \bot$ becomes $A(x) \land B(x) \rightarrow$

$A \sqsubseteq \forall R.B$ becomes $A(x) \land R(x,y) \rightarrow B(y)$

$A \sqsubseteq \neg B \sqcup C$ becomes $A(x) \land B(x) \rightarrow C(x)$

$\top \sqsubseteq \leq 1 R.\top$ becomes $R(x,y) \land R(x,z) \rightarrow y=z$

$A \sqcap \exists R.\exists S.B \sqsubseteq C$
becomes
$A(x) \land R(x,y) \land S(y,z) \land B(z) \rightarrow C(x)$

(1) Description Logics Rules

- Some classes can be fragmented into rules (2/2)

$A \sqcap \exists R.\{b\} \sqsubseteq C$	becomes	$A(x) \wedge R(x,b) \rightarrow C(x)$
$\{a\} \equiv \{b\}$	becomes	$\rightarrow a=b$
$A \sqsubseteq B \sqcap C$	becomes	$A(x) \rightarrow B(x)$ and $A(x) \rightarrow C(x)$
$A \sqcup B \sqsubseteq C$	becomes	$A(x) \rightarrow C(x)$ or $B(x) \rightarrow C(x)$

(1) Description Logics Rules

- In general:
  - A **DL axiom α** can be translated into rules, if after translating **α** into a **FOL expression α'**, and after normalizing this expression into a **set of clauses M**, each formula in **M** is a **Horn clause** (i.e. a rule)

- What about complex properties (property chains)?

(1) Description Logics Rules

- Property Chains can be expressed as rules

$$\text{hasParent} \circ \text{hasBrother} \sqsubseteq \text{hasUncle}$$

- becomes

$$\text{hasParent}(x,y) \wedge \text{hasBrother}(y,z) \rightarrow \text{hasUncle}(x,z)$$

- in general:

$$R \circ S \sqsubseteq T \text{ becomes } R(x,y) \wedge S(y,z) \rightarrow T(x,z)$$

(1) Description Logics Rules

- Problems:
  - What if on both sides of the rule are **not only properties or not only classes** ?

    Example: $\texttt{Man(x)} \wedge \texttt{hasChild(x,y)} \rightarrow \texttt{fatherOf(x,y)}$

  - How to express this in OWL2?

(1) Description Logics Rules

- Example: $Man(x) \wedge hasChild(x,y) \rightarrow fatherOf(x,y)$
- How to express this in OWL2?

- Idea:
  - **substitute** Man(x) with a **property** to make a property chain
  - Apply „Self“ to **transform classes into properties**
  - Auxiliary property P_{Man}
  - Auxiliary axiom $Man \equiv \exists P_{Man}.Self$
  - Thus: $P_{Man} \circ hasChild \sqsubseteq fatherOf$

(1) Description Logics Rules

- Rolification of concepts/classes

$A(x) \wedge R(x,y) \rightarrow S(x,y)$ becomes $P_A \circ R \sqsubseteq S$

$A(y) \wedge R(x,y) \rightarrow S(x,y)$ becomes $R \circ P_A \sqsubseteq S$

$A(x) \wedge B(y) \wedge R(x,y) \rightarrow S(x,y)$
becomes
$P_A \circ R \circ P_B \sqsubseteq S$

(1) Description Logics Rules

- More Problems:
  - Example: $Vegetarian(x) \wedge Fishproduct(y) \rightarrow dislikes(x,y)$
- How to express this in OWL2?
- Idea:
  - Apply **universal property U** to connect classes
  - Auxiliary properties: $P_{Vegetarian}$ and $P_{Fishproduct}$
  - Auxiliary Axioms: $Vegetarian \equiv \exists P_{Vegetarian}.Self$
    $Fishproduct \equiv \exists P_{Fishproduct}.Self$
  - Thus: $P_{Vegetarian} \circ U \circ P_{Fishproduct} \sqsubseteq dislikes$

(1) Description Logics Rules

- More Problems:
  - Example: $\text{orderedDish}(x,y) \wedge \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x)$
- How to express this in OWL2?
- Idea:
  - Is it possible to eliminate variable y to form a class inclusion in OWL2?

$$\exists \text{orderedDish}.\top \sqcap \exists \text{dislikes}.\top \sqsubseteq \text{Unhappy} \ ??$$

$$\exists \text{orderedDish}.\exists \text{dislikes-}.\top \sqsubseteq \text{Unhappy} \ ??$$

Decidable Fragments of SWRL

- Which SWRL knowledge bases enable **complete inference algorithms**?
  - All SWRL knowledge bases that exist only of OWL / OWL2 axioms
  - All SWRL knowledge bases that exist only of sets of DATALOG programs
  - Every static finite class of SWRL knowledge bases

- Are there more SWRL fragments that are decidable?
  1. **Description Logics Rules**
     SWRL Rules that can already be expressed with OWL2
  2. **DL-safe Rules**
     Restriction of SWRL Rules via variable assignment constraints

(2) DL-Safe Rules

- **Idea:**
  - *Restrict SWRL to guarantee decidability via restricting the number of possible variable assignments*
- Arbitrary DATALOG rules are allowed with names of OWL classes and OWL properties
- Rules must be **DL-safe**: each variable must also be represented within an expression of the rule body that doesn‘t use an OWL class or OWL property *(=* DATALOG *Atom)*
- Semantics can be adopted from OWL+SWRL (FOL).

➤ **DL-safety** restricts the application of rules to known individuals

How to force DL-Safety?

- Example: hasFather(x,y) ∧ hasBrother(y,z) → hasUncle(x,z)
- **NOT DL-safe**, if „hasBrother" and „hasFather" are OWL properties
- Force DL-Safeness by **Restricting the rules to known individuals**

  O(x)∧O(y)∧O(z)∧ hasFather(x,y)∧hasBrother(y,z)→hasUncle(x,z)

- whereby the fact O(a) has to be asserted for all OWL individuals
- *Rule is only applicable to known OWL individuals (Named Individuals)*

Forcing DL-Safety

- Example: *„He, who hates his brother, is evil.“*

Knowledge Base:

hasFather(Kain, Adam)
hasFather(Abel, Adam)
$\exists$hasFather.hasFather⁻{Remus}(Romulus)
hates(Kain, Abel)
hates(Romulus, Remus)

hasFather(x,z)∧hasFather(y,z)∧hates(x,y) → evil(x)
O(x)∧O(y)∧O(z)∧hasFather(x,z)∧hasFather(y,z)∧hates(x,y) → $evil_{safe}$(x)

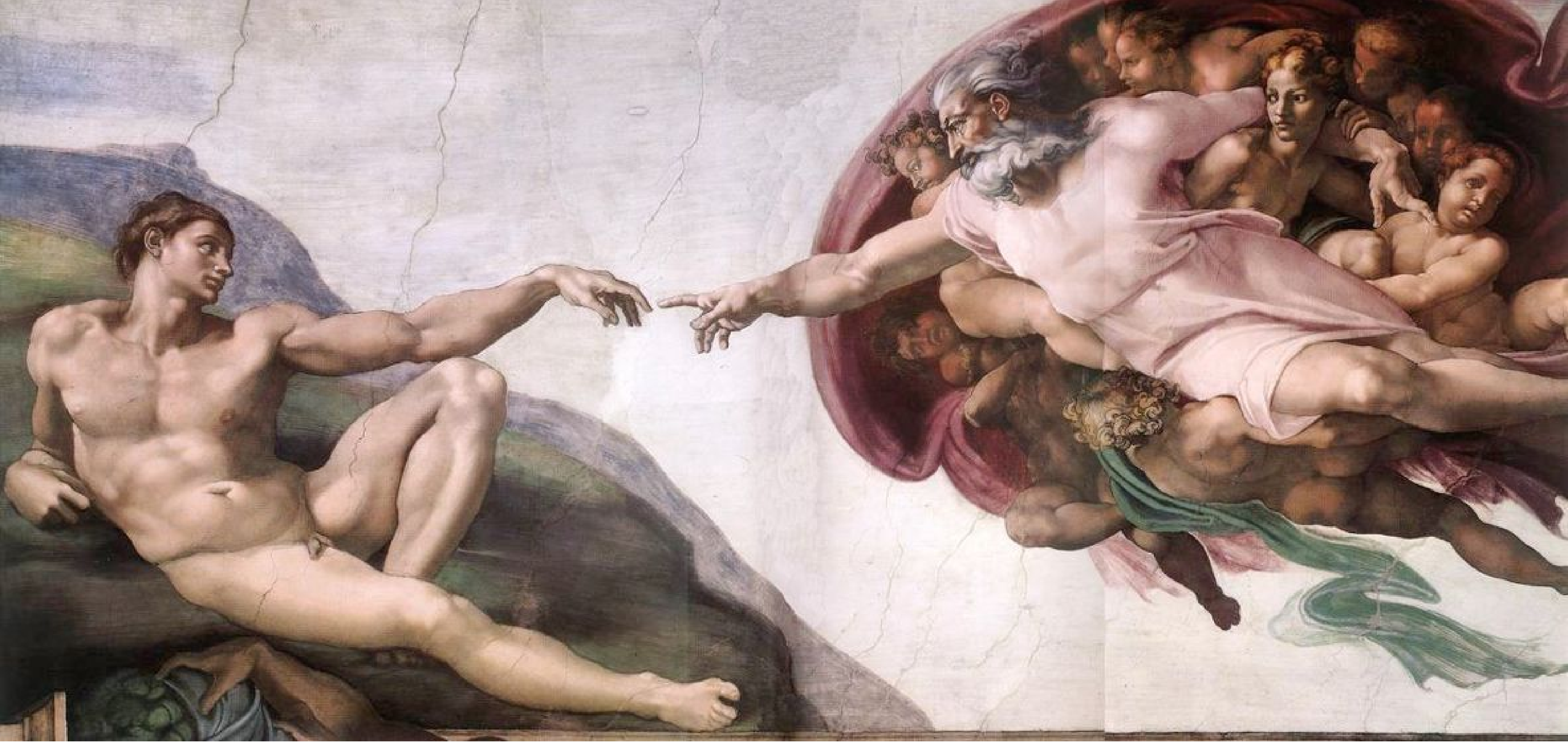
O(Kain), O(Abel), O(Adam), O(Romulus), O(Remus)

- this implies: evil(Kain), evil(Romulus)
- this implies (**safe**): $evil_{safe}$(Kain)

  BUT NOT: $evil_{safe}$(Romulus)

Rules & the Semantic Web - Summary

- SWRL („OWL + DATALOG“) is **not decidable**
- **Description Logic Rules**
  - SWRL fragment that can be expressed with OWL 2
  - indirectly supported by OWL 2 reasoners
- **DL-safe Rules**
  - SWRL fragment where variables only can be assigned to concrete values (constants)
  - supported by OWL reasoner
  - DL-safety can be forced
- Standard and Best-Practice for rules still does not exist



OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 5: Ontological Engineering