

## This detail from Sandro Botticelli's 'Birth of Venus' focuses on the central figure, Venus, as she reclines on a large, scallop-shaped shell. She is depicted with a fair complexion and long, flowing blonde hair, wearing a pinkish-red mantle that is partially draped over her. Her gaze is directed towards the viewer. To her left, a figure in a blue and white garment is partially visible, and to her right, another figure in a red garment is seen. The background features a landscape with a body of water and a distant shore. The overall style is characteristic of the Italian Renaissance, with soft lines and a focus on idealized beauty.



HPI Hasso Plattner Institut

Hasso Plattner Institute for IT Systems Engineering  
University of Potsdam

## Autumn 2015

- Reasoning with OWL is hard
  - OWL Full is **undecidable** with RDF-based semantics
  - OWL DL ( $\mathcal{SROIQ}(\mathcal{D})$ ) with some structural restrictions on regularity and simplicity is **decidable** in N2ExpTime
- **Idea:**
  - Identify maximal OWL sublanguages (fragments) for which reasoning is easier, i.e. that are decidable in **polynomial time**

- Main reason for super-polynomial runtime: **Non-Determinism**  
(requires guessing / backtracking)
  - Disjunction or Negation + Conjunction
  - Maximum number restrictions
  - Combination of existential and universal quantification in one superclass
  - Non-unary enumerated classes
  - ▶ therefore not allowed for OWL Profiles
- **Attention:** *many other features may also lead to non-determinism...*

# OWL 2 EL is based on the description logic $\mathcal{EL}^{++}$

## Description Logic $\mathcal{EL}^{++}$

- Conjunction  $C \sqcap D$ , existential Restriction  $\exists R.C$ ,  $\top$  and  $\perp$
- Nominals, restricted property ranges, self restriction
- General property inclusion (RBox), Transitivity, Reflexivity
- **NOT ALLOWED:**     *universal quantification, disjunction, complement, number restrictions, disjoint and inverted properties*

## ● Advantages:

- Polynomial complexity for standard entailment, i.e. decidability, class membership, etc.
- simple implementation
- supports important ontologies (e.g. SNOMED-CT)

# OWL 2 EL Examples

- $\exists \text{has.Sorrow} \sqsubseteq \exists \text{has.Liqueur}$
- $\exists \text{married}.\top \sqcap \text{CatholicPriest} \sqsubseteq \perp$
- $\text{German} \sqsubseteq \exists \text{knows}.\{\text{merkel}\}$
- $\exists \text{livesIn}.\{\text{europe}\} \sqsubseteq \text{European}$
- $\text{hasParent} \circ \text{hasParent} \sqsubseteq \text{hasGrandparent}$
- $\text{CEO} \sqsubseteq \exists \text{supervisedBy.Self}$

# OWL 2 QL is based on the description logic $\mathcal{DL}$ -Lite

## Description Logic $\mathcal{DL}$ -Lite

- Superclasses ( $R \sqsubseteq S$ ):  $\sqcap, \neg, \exists R.C$
- Subclasses ( $R \sqsubseteq S$ ):  $\exists R.\top$
- inverse properties, simple property hierarchies
- ABox like  $\mathcal{SROIQ}(\mathcal{D})$
- **NOT ALLOWED:** *universal quantification, enumerated classes, disjunction, self, functional and inverse functional properties, number restrictions, transitivity, general property inclusion, equality of individuals*

### ● Advantages:

- Sub-polynomial complexity (related to Relational Databases), Instance retrieval in LogSpace
- fast implementations available, scalable

# OWL 2 QL Examples

- $\exists \text{married}.\top \sqsubseteq \text{Lucky} \sqcap \exists \text{has.NoSorrows}$
- $\text{Doctor} \sqsubseteq \exists \text{treats.Patients}$

# OWL 2 RL is based on Horn-Rule fragment of OWL2

- Subclass axioms ( $R \sqsubseteq S$ ) can be interpreted as rules ( $R \rightarrow S$ )

## Horn-Rule fragment of OWL 2:

- Superclasses ( $R \sqsubseteq S$ ):  $\sqcap, \exists R.\{a\}, \forall R.C, \leq 1R.C$
- Subclasses ( $R \sqsubseteq S$ ):  $\sqcap, \sqcup, \exists R.C, \exists R.\{a\}$
- $\perp$
- **NOT ALLOWED:** *negated facts, reflexivity, ...*

- **Advantages:**

- Polynomial complexity (PTime-complete)
- Simple Implementation (OWL Axioms as Rules)
- Related to rule languages

- $\exists \text{parentOf} . \exists \text{parentOf} . \top \sqsubseteq \text{Grandparent}$   
as rule:  $\text{parentOf}(x,y) \wedge \text{parentOf}(y,z) \rightarrow \text{Grandparent}(x)$
- $\text{Orphan} \sqsubseteq \forall \text{hasParent} . \text{Dead}$   
as rule:  $\text{Orphan}(x) \wedge \text{hasParent}(x,y) \rightarrow \text{Dead}(y)$
- $\text{Monogamous} \sqsubseteq \leq 1 \text{ married} . \text{Alive}$   
as rule:  $\text{Monogamous}(x) \wedge \text{married}(x,y) \wedge \text{Alive}(y) \wedge \text{married}(x,z) \wedge \text{Alive}(z) \rightarrow y=z$
- $\text{childOf} \circ \text{childOf} \sqsubseteq \text{grandchildOf}$   
as rule:  $\text{childOf}(x,y) \wedge \text{childOf}(y,z) \rightarrow \text{grandchildOf}(x,z)$

# Do we need so many OWL Fragments?

- The union of any two of the OWL profiles is no longer light-weight!  
QL+RL, QL+EL, RL+EL all ExpTime-hard
- Restricting to fewer profiles = giving up potentially useful feature combinations
- **Rationale:**
  - profiles are “maximal” (well, not quite) well-behaved OWL 2 fragments
  - select a suitable feature set for your applications



## **13 EXTRA - SWRL - Semantic Web Rule Language**

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

**Lecture 4: OWL, Rules, and Reasoning**