CS 7810 - Knowledge Representation and Reasoning (for the Semantic Web)
07 - OWL and Description Logics

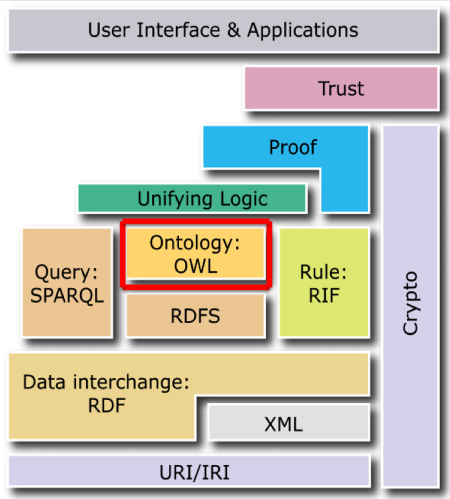
Adila Krisnadhi

Data Semantics Lab
Wright State University, Dayton, OH

November 1, 2016

Outline

DaSe Lab


User Interface & Applications
Trust
Proof
Unifying Logic
Query: SPARQL
Ontology: OWL
RDFS
Rule: RIF
Crypto
Data interchange: RDF
XML
URI/IRI

Acknowledgements

Materials in this presentation are adapted from:

- Markus Krötzsch, "OWL Syntax and Intuition", slides for Foundations of Semantic Web Technologies course, TU Dresden, May 14, 2014.
- Markus Krötzsch, "OWL & Description Logics", slides for Foundations of Semantic Web Technologies course, TU Dresden, May 16, 2014.
- Markus Krötzsch, "OWL Syntax and Semantics", slides for Foundations of Semantic Web Technologies course, TU Dresden, May 16, 2014.
- Axel Polleres, "Unit 5: OWL, OWL 2, SPARQL+OWL", slides for Semantic Web Technologies course, TU Vienna, June 06, 2012.

Outline

Description Logics (DLs)

- One of the most prominent KR paradigms.
- Significantly influenced standardization of Semantic Web languages.
  - OWL is essentially based on DLs.
- Numerous reasoners: Quonto, JFact, FaCT++, RacerPro, Owlgres, Pellet, SHER, snorocket, OWLIM, Jena, Oracle, Prime, QuOnto, Trowl, HermiT, condor, CB, ELK, konclude, RScale
- Precursor of DLs: semantic networks and frame-based systems
  - Semantic networks and frame-based systems are equipped only with intuitive semantics – diverging interpretations
  - DLs provide logic-based formal semantics
- DLs can be seen as decidable fragments of first-order logic, and is closely related to modal logics.
- Focus of DL research: (worst-case) computational complexity of various reasoning tasks.
- Most DLs are of high (computational) complexity, but optimized reasoning algorithms with good average case behaviour exist.

DL building blocks

- (Named) individuals: `adila`, `pascal`, `wsu`, `cs7220`
  - constants in FOL
- Concept names (in OWL: classes): `University`, `Person`, `Course`, `Student`
  - unary predicates in FOL
- (Abstract and concrete) role names (in OWL: object and data properties)
  - binary predicates in FOL

The set of all individuals, concept names, and role names is called the **signature** or **vocabulary**.

Components of DL Knowledge Bases

- **TBox**: contains information about concepts and their taxonomic relationships
- **ABox**: contains information about individuals, and their concept and role memberships
- **RBox**: contains information about roles and their mutual dependencies

A **DL knowledge base** (**KB**) or **ontology** is the union of a TBox, an ABox, and an RBox.

Outline

DaSe Lab

The Description Logic $\mathcal{ALC}$

$\mathcal{ALC}$: Attributive Language with Complement, the simplest DL that is Boolean/propositionally closed.

(Complex) $\mathcal{ALC}$ concepts are defined as follows:

- every concept name is a concept
- $\top$ and $\bot$ are concepts
- for every concept C and D, $\neg C$, $C \sqcap D$, and $C \sqcup D$ are concepts
- for every role R and concept C, $\exists R.C$ and $\forall R.C$ are concepts.

Example: $\texttt{Student} \sqcap \forall \texttt{attends.GraduateCourse}$
Intuitively describes the concept comprising all students who attend only graduate courses.
In OWL, corresponds to the class expression:

```
ObjectIntersectionOf(
  Student
  ObjectAllValuesFrom( attends GraduateCourse ) )
```

- $\top$ corresponds to `owl:Thing`
- $\bot$ corresponds to `owl:Nothing`
- $\sqcap$ corresponds to `ObjectIntersectionOf`
- $\sqcup$ corresponds to `ObjectUnionOf`
- $\neg$ corresponds to `ObjectComplementOf`
- $\forall$ corresponds to `ObjectAllValuesFrom`
- $\exists$ corresponds to `ObjectSomeValuesFrom`

Knowledge bases in $\mathcal{ALC}$ I

TBox is a set of **general concept inclusion** (GCI) axioms.

- Every GCI is an axiom of the form $C \sqsubseteq D$ where C, D are concepts
- An axiom of the form $C \equiv D$ is an abbreviation of two axioms $C \sqsubseteq D$ and $D \sqsubseteq C$
- In OWL:
  - $\sqsubseteq$ corresponds to `SubClassOf`
  - $\equiv$ corresponds to `EquivalentClasses` with two parameters.

Knowledge bases in $\mathcal{ALC}$ II

ABox is a set of **ABox assertions**.

- $\mathcal{ALC}$ ABox assertions can be a concept assertion or a role assertion.
- **Concept assertion** is of the form $C(a)$ where C is a concept and a is an individual
- **Role assertion** is of the form $R(a, b)$ where R is a role and a, b are individuals
- In OWL, concept assertion corresponds to `ClassAssertion` and role assertion corresponds to `ObjectPropertyAssertion`

$\mathcal{ALC}$ does not support RBoxes. But more expressive DLs do support RBoxes.

Semantics of $\mathcal{ALC}$ I

- $\mathcal{ALC}$ is a syntactic variant of the modal logic $\mathbf{K}$ with multiple modalities.
  $\leadsto$ Semantics of $\mathcal{ALC}$ can be stated using the semantics of $\mathbf{K}$.
- $\mathcal{ALC}$ is a fragment/sublanguage of first-order predicate logic.
  $\leadsto$ Semantics of $\mathcal{ALC}$ can be stated using first-order interpretations.
- Here, we define the semantics of $\mathcal{ALC}$ using a definition traditionally presented in the DL literature.

We define a model-theoretic semantics for $\mathcal{ALC}$ via interpretations.
In OWL, this corresponds to the OWL Direct Semantics.

A **DL interpretation** $\mathcal{I}$ consists of a domain/universe $\Delta^\mathcal{I}$ and a function $\cdot^\mathcal{I}$ that maps:

- every individual name a to a universe element $a^\mathcal{I}$ where $a^\mathcal{I} \in \Delta^\mathcal{I}$
- every concept name C to a set of universe elements $C^\mathcal{I}$ where $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$
- every role name R to a set of pairs of universe elements $R^\mathcal{I}$ where $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$

Every complex $\mathcal{ALC}$ concept is interpreted as follows:

$$
\begin{aligned}
\top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &:= \emptyset \\
(\neg C)^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid x \notin C^{\mathcal{I}}\} \\
(C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &:= \{x \in \Delta^{\mathcal{I}} \mid \text{for every } y \in \Delta^{\mathcal{I}} : \langle x, y\rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\} \\
&\leadsto \text{the set of } x \text{ for which there is no } y \text{ with } \langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \notin C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &:= \{x \in \Delta^{\mathcal{I}} \mid \text{there is some } y \in \Delta^{\mathcal{I}} \text{ with } \langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}
\end{aligned}
$$

Semantics of $\mathcal{ALC}$ IV

A DL interpretation $\mathcal{I}$ **satisfies** (**is a model of**) axioms:

- $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ – notation: $\mathcal{I} \models C \sqsubseteq D$
- $C \equiv D$ if $C^\mathcal{I} = D^\mathcal{I}$ – notation: $\mathcal{I} \models C \equiv D$
- $C(a)$ if $a^\mathcal{I} \in C^\mathcal{I}$ – notation: $\mathcal{I} \models C(a)$
- $R(a,b)$ if $\langle a^\mathcal{I}, b^\mathcal{I} \rangle \in R^\mathcal{I}$ – notation: $\mathcal{I} \models R(a,b)$

A DL interpretation $\mathcal{I}$ is:

- a **model of** a TBox $\mathcal{T}$ if $\mathcal{I} \models \alpha$ for every axiom $\alpha \in \mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$
- a **model of** an ABox $\mathcal{A}$ if $\mathcal{I} \models \alpha$ for every assertion $\alpha \in \mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$
- a **model of** an RBox $\mathcal{R}$ if $\mathcal{I} \models \alpha$ for every role axiom in $\mathcal{R}$, written $\mathcal{I} \models \mathcal{R}$; note: for $\mathcal{ALC}$, $\mathcal{R}$ is always empty
- a **model of** a KB $\mathcal{K} = \mathcal{T} \cup \mathcal{A} \cup \mathcal{R}$ if $\mathcal{I} \models \mathcal{T}$, $\mathcal{I} \models \mathcal{A}$, and $\mathcal{I} \models \mathcal{R}$.

An axiom α **(logically) follows** from a KB $\mathcal{K}$, written $\mathcal{K} \models \alpha$, if every model $\mathcal{I}$ of $\mathcal{K}$ is also a model of α.
$\rightsquigarrow$ "logically follows from" = "is entailed by" = "is a logical consequence of"

Basic Inference Problems

- KB consistency/satisfiability: "Does $\mathcal{K}$ have a model?"
- Concept/class inconsistency/unsatisfiability: "$\mathcal{K} \models^? C \sqsubseteq \bot$"
- Concept/class inclusion (subsumption): "$\mathcal{K} \models^? C \sqsubseteq D$"
- Concept/class equivalence: "$\mathcal{K} \models^? C \equiv D$"
- Concept/class disjointness: "$\mathcal{K} \models^? C \sqcap D \sqsubseteq \bot$"
- Instance checking: "$\mathcal{K} \models^? C(a)$"
- Instance retrieval: Find all individual name x with "$\mathcal{K} \models C(x)$".

- A DL is decidable if there is a sound, complete, and terminating algorithm that solves the aforementioned inference problems.
- Otherwise, a DL is called undecidable.
- DL is a fragment of FOL:
  $\leadsto$ FOL inference procedures (resolution, tableaux) can be used
  $\leadsto$ but these algorithms are not guaranteed to terminate
- Main research challenge in DL: find terminating algorithms!
- There is no "naive" solutions for this (unlike in propositional logic where we can use truth tables)

Examples with FOL and OWL Translations I

Rose is a mother.

$$\texttt{Mother}(\texttt{rose}) \qquad \text{(DL/FOL)}$$

$$\texttt{ClassAssertion}(\texttt{Mother rose}) \qquad \text{(OWL)}$$

Rose is a parent of John.

$$\texttt{parentOf}(\texttt{rose}, \texttt{john}) \qquad \text{(DL/FOL)}$$

$$\texttt{ObjectPropertyAssertion}(\texttt{parentOf rose john}) \qquad \text{(OWL)}$$

Every mother is a parent.

$$\texttt{Mother} \sqsubseteq \texttt{Parent} \qquad \text{(DL)}$$
$$\forall x(\texttt{Mother}(x) \to \texttt{Parent}(x)) \qquad \text{(FOL)}$$
$$\texttt{SubClassOf}(\texttt{Mother Parent}) \qquad \text{(OWL)}$$

Parents are exactly either fathers or mothers.

$$\texttt{Parent} \equiv \texttt{Father} \sqcup \texttt{Mother} \qquad \text{(DL)}$$
$$\forall x\big(\texttt{Parent}(x) \leftrightarrow (\texttt{Father}(x) \vee \texttt{Mother}(x))\big) \qquad \text{(FOL)}$$
$$\texttt{EquivalentClasses}(\texttt{Parent ObjectUnionOf}(\texttt{Father Mother})) \qquad \text{(OWL)}$$

Nobody can be both female and male at the same time.

$$\texttt{Male} \sqcap \texttt{Female} \sqsubseteq \bot \qquad \text{(DL)}$$
$$\forall x\big((\texttt{Male}(x) \wedge \texttt{Female}(x)) \to \textbf{false}\big) \qquad \text{(FOL)}$$
$$\texttt{SubClassOf}(\texttt{ObjectIntersectionOf}(\texttt{Male Female})\ \texttt{owl:Nothing}) \qquad \text{(OWL)}$$

Examples with FOL and OWL Translations III

A parent is precisely someone who is a parent of some individual.

$$\texttt{Parent} \equiv \exists \texttt{parentOf}.\top \quad \text{(DL)}$$

$$\forall x \big(\texttt{Parent}(x) \leftrightarrow \exists y (\texttt{parentOf}(x,y))\big) \quad \text{(FOL)}$$

$$\texttt{EquivalentClasses}(\texttt{Parent ObjectSomeValuesFrom}(\texttt{parentOf owl:Thing})) \quad \text{(OWL)}$$

A father without a son is a parent who is not a female and is a parent of only females.

$$\texttt{FatherWithoutSon} \equiv \texttt{Parent} \sqcap \neg \texttt{Female} \sqcap \forall \texttt{parentOf}.\texttt{Female} \quad \text{(DL)}$$

$$\forall x \big(\texttt{FatherWithoutSon}(x) \leftrightarrow \big(\texttt{Parent}(x) \wedge \neg \texttt{Female}(x) \wedge \forall y (\texttt{parentOf}(x,y) \rightarrow \texttt{Female}(y))\big)\big) \quad \text{(FOL)}$$

```
EquivalentClasses(FatherWithoutSon
        ObjectIntersectionOf(
            Parent
            ObjectComplementOf(Female)
            ObjectAllValuesFrom(parentOf Female)))   (OWL)
```

DL Translation into FOL

- Translation of $\mathcal{ALC}$ axioms into FOL only needs at most two variables
  $\rightsquigarrow$ $\mathcal{ALC}$ is a fragment of FOL with two variables (this fragment of FOL is called $\mathcal{L}_2$
  $\rightsquigarrow$ Checking satisfiability of sets of $\mathcal{ALC}$ axioms is decidable.

Outline

- An **(abstract) role** is either a role name or an **invese role**.
- Inverse role is denoted with R^- where R is a role.
- Semantics of inverse roles:

$$(R^-)^\mathcal{I} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R^\mathcal{I}\}$$

- Note that $(R^-)^-$ is equivalent to R.
- Allowing inverse roles in $\mathcal{ALC}$ gives us the DL $\mathcal{ALCI}$
- In OWL, roles are called **object property expressions**, and the notation R^- is written as `ObjectInverseOf`(R)
- **Concrete roles** are used for binary relationships between an element of $\Delta^\mathcal{I}$ and concrete literal values. In OWL, they are called **data properties**

Universal Role and Empty Role

- The **universal role** is denoted by U.
- Semantics: $U^{\mathcal{I}} = \{\langle x, y \rangle \mid x, y \in \Delta^{\mathcal{I}}\}$
- In OWL: denoted by `owl:topObjectProperty`

Universal Role and Empty Role

- The **universal role** is denoted by U.
- Semantics: $U^{\mathcal{I}} = \{\langle x, y \rangle \mid x, y \in \Delta^{\mathcal{I}}\}$
- In OWL: denoted by `owl:topObjectProperty`

- The **empty role** is introduced in OWL as `owl:bottomObjectProperty`, but DL doesn't have a notation for it.
- Semantically, the empty role represents the empty relation.
- Can you express it using concept inclusion?

- A role axiom is either:
  - role inclusion axiom (RIA), which is either
    - role hierarchy, or
    - general role inclusion axiom
  - a role characteristic axiom, which is either
    - role transitivity
    - role functionality
    - role inverse functionality
    - role reflexivity
    - role irreflexivity
    - role symmetry
    - role asymmetry
    - role disjointness
- An **RBox** is a set of role axioms

- Given two roles R, S, a **role hierarchy** is an axiom of the form $R \sqsubseteq S$.
- An interpretation $\mathcal{I}$ is a **model of** a role hierarchy $R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.
- FOL translation: $\forall x \forall y (R(x,y) \rightarrow S(x,y))$
- In OWL: `SubObjectPropertyOf`(R S)
- $R \equiv S$ is a shorthand for $R \sqsubseteq S$ and $S \sqsubseteq R$
  - In OWL: `EquivalentObjectProperties`(R S)
- Extending $\mathcal{ALC}$ with role hierarchy gives us the DL $\mathcal{ALCH}$, and if we also have inverse roles: $\mathcal{ALCHI}$
  $\rightsquigarrow$ $\mathcal{ALCHI}$ allows role hierarchy of the form $R \equiv S^-$, which corresponds to `InverseObjectProperties`(R S) in OWL.

RBox Axioms III

- **Role transitivity axiom** is of the form $\mathsf{Trans}(R)$ where R is a role.
- An interpretation $\mathcal{I}$ is a **model of** $\mathsf{Trans}(R)$ iff $R^{\mathcal{I}}$ is a transitive relation, i.e., $\langle x, y\rangle \in R^{\mathcal{I}}$ and $\langle y, z\rangle \in R^{\mathcal{I}}$ together imply $\langle x, z\rangle \in R^{\mathcal{I}}$
- In FOL: $\forall x \forall y \forall z((R(x, y) \wedge R(y, z)) \rightarrow R(x, z))$
- In OWL: `TransitiveObjectProperty`(R)
- Extending $\mathcal{ALC}$ with role transitivity gives us the DL $\mathcal{S}$ (named after the modal logic S_4 – although it is actually based on K_4)

DaSe Lab

- Role hierarchy and transitivity are special cases of role inclusion axioms (RIAs) of the form:

$$R_1 \circ \cdots \circ R_n \sqsubseteq S$$

where all R_i's and S's are roles and $n \geq 1$.
  - The expression $R_1 \circ \cdots \circ R_n$ is called **role chain** (in OWL, **property chain**)
- Reasoning in $\mathcal{ALC}$ + RIAs is undecidable
$\leadsto$ to regain decidability, the set of RIAs in the RBox needs to be "acyclic" (explained later).
- Interpretation $\mathcal{I}$ is a **model of** $R_1 \circ \cdots \circ R_n \sqsubseteq S$ iff $\langle x, x_1 \rangle \in R_1^{\mathcal{I}}$, $\langle x_1, x_2 \rangle \in R_2^{\mathcal{I}}$, ..., and $\langle x_{n-1}, y \rangle \in R_n^{\mathcal{I}}$ together imply that $\langle x, y \rangle \in S^{\mathcal{I}}$
- In FOL: $\forall x \forall y (\exists x_1 \ldots \exists x_{n-1} (R_1(x, x_1) \wedge \cdots \wedge R_n(x_{n-1}, y)) \rightarrow S(x, y))$
- In OWL: `SubObjectPropertyOf(ObjectPropertyChain(`R_1 R_2 $\ldots R_n$`)` S`)`

To ensure decidability, a set of RIAs need to be **regular**

A set $\mathcal{R}$ be of RIAs is regular if there is a strict linear ordering $\prec$ of all role names in $\mathcal{R}$ such that

- For any two different role names R, S, $R \prec S$ iff $R^- \prec S$
- Every RIA in $\mathcal{R}$ is one of the following forms where $R_i \prec R$ for all $1 \leq i \leq n$:
  - $R \circ R \sqsubseteq R$
  - $R^- \sqsubseteq R$
  - $R_1 \circ \cdots \circ R_n \sqsubseteq R$
  - $R \circ R_1 \circ \cdots \circ R_n \sqsubseteq R$
  - $R_1 \circ \cdots \circ R_n \circ R \sqsubseteq R$

Roughly, a regular set of RIAs does not contain cyclic definitions that involve RIAs with role chains.

- **Role functionality** is of the form Func(R) where R is a role.
  - Here, we say that R is **functional**.
- An interpretation $\mathcal{I}$ is a **model of** Func(R) iff $R^{\mathcal{I}}$ is a binary relation that forms a function, i.e., $\langle x, y_1 \rangle \in R^{\mathcal{I}}$ and $\langle x, y_2 \rangle \in R^{\mathcal{I}}$ together imply $y_1 = y_2$.
- In FOL (with equality): $\forall x \forall y_1 \forall y_2 ((R(x, y_1) \wedge R(x, y_2)) \rightarrow y_1 = y_2)$
- In OWL: `FunctionalObjectProperty`(R)
- $\mathcal{ALC}$ + role functionality = $\mathcal{ALCF}$
- If we also allow inverse roles, we obtain $\mathcal{ALCFI}$ and furthermore, we can also express **role inverse functionality** of the form Func(R^-).
  - Here, we say that R is **inverse functional** (or R^- is functional)
  - In OWL: `InverseFunctionalObjectProperty`(R)
- Interpretation $\mathcal{I}$ is a **model of** Func(R^-) iff $\langle x_1, y \rangle \in R^{\mathcal{I}}$ and $\langle x_2, y \rangle \in R^{\mathcal{I}}$ together imply $x_1 = x_2$

RBox Axioms VII

- **Role disjointness axiom** is of the form Dis(R, S) where R, S must be simple roles (defined later).
- Interpretation $\mathcal{I}$ is a **model of** Dis(R, S) iff $R^{\mathcal{I}}$ and $S^{\mathcal{I}}$ are disjoint, i.e., $R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$, or no x, y that are connected by both R and S.
- In FOL: $\neg\exists x \exists y (R(x, y) \wedge S(x, y))$
- In OWL: `DisjointObjectProperties`(R S)

RBox Axioms VIII

- **Role reflexivity axiom** is of the form Ref(R) where R is a role.
- Interpretation $\mathcal{I}$ is a **model of** Ref(R) iff $R^{\mathcal{I}}$ is reflexive, i.e., $\langle x, x\rangle \in R^{\mathcal{I}}$ for every $x \in \Delta^{\mathcal{I}}$.
- In FOL: $\forall x.R(x, x)$
- In OWL: `ReflexiveObjectProperty`(R)

- **Role irreflexivity axiom** is of the form Irr(R) where R must be a simple role (defined later).
- Interpretation $\mathcal{I}$ is a **model of** Irr(R) iff $R^{\mathcal{I}}$ is irreflexive/anti-reflexive, i.e., $\langle x, x\rangle \notin R^{\mathcal{I}}$ for every $x \in \Delta^{\mathcal{I}}$.
  $\rightsquigarrow$ Being irreflexive is different from being not reflexive!
- In FOL: $\forall x.\neg R(x, x)$
- In OWL: `IrreflexiveObjectProperty`(R)

RBox Axioms IX

- **Role symmetry axiom** is of the form Sym(R) where R is a role.
- Interpretation $\mathcal{I}$ is a **model of** Sym(R) iff $R^{\mathcal{I}}$ is symmetric, i.e., $\langle x, y\rangle \in R^{\mathcal{I}}$ implies $\langle y, x\rangle \in R^{\mathcal{I}}$.
- In FOL: $\forall x(R(x, y) \to R(y, x))$
- In OWL: `SymmetricObjectProperty`(R)
- Can you express it using role hierarchies?

- **Role asymmetry axiom** is of the form Asym(R) where R is a role.
- Interpretation $\mathcal{I}$ is a **model of** Asym(R) iff $R^{\mathcal{I}}$ is asymmetry, i.e., $\langle x, y\rangle \in R^{\mathcal{I}}$ implies $\langle y, x\rangle \notin R^{\mathcal{I}}$.
  $\rightsquigarrow$ Asymmetry is different from non-symmetry and anti-symmetry!
- In FOL: $\forall x(R(x, y) \to \neg R(y, x))$
- In OWL: `AssymmetricObjectProperty`(R)
- Can you express it using role disjointness axiom?

- **(Qualified) number restrictions** are concepts of the form $\leqslant nR.C$, $\geqslant nR.C$, and $=nR.C$ where n is a nonnegative integer, R a role, and C a concept.
  - n is not a variable, i.e., must be given a value
  - To ensure decidability, R must be a simple role (defined later)
  - $=nR.C$ is a shorthand for $\leqslant nR.C$ and $\geqslant nR.C$
  - If C is omitted, then C is actually `owl:Thing` and the number restrictions are called **unqualified number restrictions**
- Semantics:

$$(\leqslant nR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{\langle x, y\rangle \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \leqslant n\}$$
$$(\geqslant nR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{\langle x, y\rangle \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \geqslant n\}$$

- Translation to FOL needs equality or counting quantifiers
- Number restrictions in OWL correspond to `ObjectMinCardinality`, `ObjectMaxCardinality`, `ObjectExactCardinality`
- $\mathcal{ALC}$ + qualified number restrictions = $\mathcal{ALCQ}$
- $\mathcal{ALC}$ + unqualified number restrictions = $\mathcal{ALCN}$

Self Restriction

- **Self restrictions** are concepts of the form $\exists R.\mathsf{Self}$ where R must be a simple role.
- Semantics: $(\exists R.\mathsf{Self})^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \langle x, x \rangle \in R^{\mathcal{I}}\}$
- In OWL: `ObjectHasSelf`(R)

Let $\mathcal{R}$ be a set of RIAs.

- If there is an RIA of the form $R_1 \circ \cdots \circ R_n \sqsubseteq R$ or $R_1 \circ \cdots \circ R_n \sqsubseteq R^-$ where $n > 1$, we say that R is **non-simple**/**composite**.
  - Note: Trans(R) is equivalent to $R \circ R \sqsubseteq R$, i.e., if Trans$(R) \in \mathcal{R}$, then R is non-simple.
- Suppose that R is non-simple. Then if $R \sqsubseteq S \in \mathcal{R}$ or $R \sqsubseteq S^- \in \mathcal{R}$ or $R^- \sqsubseteq S \in \mathcal{R}$ or $R^- \sqsubseteq S^- \in \mathcal{R}$, then S is also non-simple.
- R is **simple** if R is not non-simple.

Roughly, a simple role does not have any direct/indirect subroles/subproperties that are transitive or are defined by property chains.

- Can you get rid of the negation below and still obtain equivalent concepts?
  - $\neg(\leqslant nR.C)$
  - $\neg(\geqslant nR.C)$ (provided that $n \geq 1$)
- Can you express the following concepts using number restrictions?
  - $\bot$
  - $\top$
  - $\exists R.C$
  - $\forall R.\bot$
  - Func(R)
- Can you express the following using concept inclusion and Self-restrictions?
  - Ref(R)
  - Irr(R)

Nominals

- A **nominal** is a concept of the form $\{a\}$ where a is an individual name.
- Semantically, $(\{a\})^{\mathcal{I}} = \{a^{\mathcal{I}}\}$, i.e., it denotes the set with a single element (denoted by a).
- $\{a_1, \ldots, a_n\}$ is a shorthand for $\{a_1\} \sqcup \cdots \sqcup \{a_n\}$
- $\mathcal{ALC}$ + nominals = $\mathcal{ALCO}$
- Corresponds to `ObjectOneOf`
- $\exists R.\{a\}$ corresponds to `ObjectHasValue`(R a) in OWL.

More Kinds of ABox Assertions

Besides

- $C(a)$ (concept assertion)
  - equivalent to $\{a\} \sqsubseteq C$
- $R(a,b)$ (role assertion)
  - equivalent to $\{a\} \sqsubseteq \exists R.\{b\}$

In extensions of $\mathcal{ALC}$, ABox assertions can also have one of the following forms:

- $\neg R(a,b)$ (negative role assertion)
  - equivalent to $\{a\} \sqsubseteq \forall R.(\neg\{b\})$
  - R must be a simple role
- $a \approx b$ (equality assertion)
  - equivalent to $\{a\} \equiv \{b\}$
- $a \not\approx b$ (inequality assertion)
  - equivalent to $\{a\} \sqsubseteq \neg\{b\}$

More Kinds of TBox and RBox axioms

Besides axioms of the form $C \sqsubseteq D$ and $C \equiv D$, in OWL, one can assert the following types of TBox and RBox axioms:

- `DisjointClasses`$(C_1\ C_2\ \ldots\ C_n)$
  - In DL: $C_i \sqcap C_j \sqsubseteq \bot$ for every $1 \leq i < j \leq n$.
- `DisjointUnion`$(C\ C_1\ C_2\ \ldots\ C_n)$
  - In DL: $C \equiv C_1 \sqcup \cdots \sqcup C_n$ and $C_i \sqcap C_j \sqsubseteq \bot$ for every $1 \leq i < j \leq n$
- `ObjectPropertyDomain`$(R\ C)$
  - In DL: dom$(R) \sqsubseteq C$.
  - Equivalent to: $\exists R.\top \sqsubseteq C$
- `ObjectPropertyRange`$(R\ C)$
  - In DL: ran$(R) \sqsubseteq C$
  - Equivalent to: $\top \sqsubseteq \forall R.C$
  - Equivalent to: $\exists R^{-}.\top \sqsubseteq C$

$\mathcal{ALC}$	Attributive Language with Complement
$\mathcal{S}$	$\mathcal{ALC}$ + role transitivity
$\mathcal{H}$	role hierarchy
$\mathcal{R}$	general role inclusion axioms
$\mathcal{I}$	inverse roles
$\mathcal{O}$	nominals
$\mathcal{N}$	unqualified number restrictions
$\mathcal{Q}$	qualified number restrictions
$\mathcal{F}$	functional roles
(D)	datatypes – in DL literature, called concrete domains

Notable DLs (some do not follow the previous nomenclature):

- $\mathcal{EL} \rightsquigarrow$ allows only $\top$, $\sqcap$, $\exists$
- $\mathcal{SHOIN}(\mathrm{D})$ – OWL 1 DL
- $\mathcal{SHIF}(\mathrm{D})$ – OWL 1 Lite
- $\mathcal{SROIQ}(\mathrm{D})$ – OWL 2 DL
- $\mathcal{SROEL}(\mathrm{D})$ – OWL 2 EL (a profile of OWL 2 DL)
  - A rather inconsistent naming as $\mathcal{SROEL}$ disallows negation, union and universal restriction ☺
  - Originally, was called $\mathcal{EL}^{++}$
- DL-Lite – OWL 2 QL (a profile of OWL 2 DL)
- DLP – OWL 2 RL (a profile of OWL 2 DL)

OWA Open World Assumption
- the existence of further individuals is possible, unless they are explicitly excluded
- DL/OWL uses the OWA

CWA Closed World Assumption
- the knowledge base is assumed to contain all individuals and facts

- $\{\text{hasChild}(\text{bill}, \text{bob}), \text{Man}(\text{bob})\} \models^{?} (\forall \text{hasChild}.\text{Man})(\text{bill})$
  - DL answer:

Open World vs. Closed World Assumption

OWA Open World Assumption
- the existence of further individuals is possible, unless they are explicitly excluded
- DL/OWL uses the OWA

CWA Closed World Assumption
- the knowledge base is assumed to contain all individuals and facts

- $\{\text{hasChild}(\text{bill}, \text{bob}), \text{Man}(\text{bob})\} \models^{?} (\forall \text{hasChild}.\text{Man})(\text{bill})$
  - DL answer: "Don't know"
  - CWA-based answer:

OWA Open World Assumption

- the existence of further individuals is possible, unless they are explicitly excluded
- DL/OWL uses the OWA

CWA Closed World Assumption

- the knowledge base is assumed to contain all individuals and facts

- $\{\mathsf{hasChild}(\mathsf{bill},\mathsf{bob}),\mathsf{Man}(\mathsf{bob})\} \models^? (\forall\mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer: "Don't know"
  - CWA-based answer: "Yes"
- $\{(\leqslant 1\mathsf{hasChild})(\mathsf{bill})\} \models^? (\forall\mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer:

Open World vs. Closed World Assumption

OWA Open World Assumption

- the existence of further individuals is possible, unless they are explicitly excluded
- DL/OWL uses the OWA

CWA Closed World Assumption

- the knowledge base is assumed to contain all individuals and facts

- $\{\mathsf{hasChild}(\mathsf{bill}, \mathsf{bob}), \mathsf{Man}(\mathsf{bob})\} \models^{?} (\forall \mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer: "Don't know"
  - CWA-based answer: "Yes"
- $\{(\leqslant 1 \mathsf{hasChild})(\mathsf{bill})\} \models^{?} (\forall \mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer: "Yes"
  - CWA-based answer:

OWA Open World Assumption

- the existence of further individuals is possible, unless they are explicitly excluded
- DL/OWL uses the OWA

CWA Closed World Assumption

- the knowledge base is assumed to contain all individuals and facts

- $\{\mathsf{hasChild}(\mathsf{bill},\mathsf{bob}),\mathsf{Man}(\mathsf{bob})\} \models^{?} (\forall\mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer: "Don't know"
  - CWA-based answer: "Yes"
- $\{(\leqslant 1\mathsf{hasChild})(\mathsf{bill})\} \models^{?} (\forall\mathsf{hasChild}.\mathsf{Man})(\mathsf{bill})$
  - DL answer: "Yes"
  - CWA-based answer: "Yes"

Outline

Concept expressions

- concept name $A, B, \ldots$
- conjunction $C \sqcap D$
- disjunction $C \sqcup D$
- negation $\neg C$
- existential role restriction $\exists R.C$
- universal role restriction $\forall R.C$
- Self restriction $\exists R.\mathsf{Self}$
- at-least restriction $\geqslant n R.C$
- at-most restriction $\leqslant n R.C$
- nominals $\{a\}, \{b\}, \ldots$

TBox axioms

- inclusion $C \sqsubseteq D$
- equivalence $C \equiv D$

ABox restrictions

- concept membership $C(a)$
- role membership $R(a, b)$
- negated role membership $\neg R(a, b)$
- equality $a \approx b$
- inequality $a \not\approx b$

Role expressions

- Role name $R, S, \ldots$
- Inverse roles R^-
- Universal role U

RBox axioms

- inclusion $R_1 \sqsubseteq R$
- complex inclusion $R_1 \circ \cdots \circ R_n \sqsubseteq R$
- transitivity $\mathsf{Trans}(R)$
- symmetry $\mathsf{Sym}(R)$
- asymmetry $\mathsf{Asym}(R)$
- reflexivity $\mathsf{Ref}(R)$
- irreflexivity $\mathsf{Irr}(R)$
- role disjointness $\mathsf{Dis}(R, S)$
- domain restriction $\mathsf{dom}(R) \sqsubseteq C$
- range restriction $\mathsf{ran}(R) \sqsubseteq C$
- role functionality $\mathsf{Func}(R)$
- role inverse functionality $\mathsf{Func}(R^-)$

- **OWL 2 Profile**: a fragment/sublanguage/trimmed down version of OWL 2 DL that trades some expressive power (i.e., by disallowing some OWL 2 constructors) for the efficiency of reasoning.
- Each profile achieves efficiency in a different way and is useful in different application scenarios.
- Spec at `https://www.w3.org/TR/owl2-profiles/` defines three profiles:
  - OWL 2 EL
  - OWL 2 QL
  - OWL 2 RL

  Obtained by putting restrictions on the syntax of OWL 2 DL.
- There are many other profiles of OWL 2 DL (e.g., a whole family that extends OWL 2 QL), but not standardized.
- Since OWL 1 Lite is a fragment of OWL 1 DL and OWL 1 DL is a fragment of OWL 2 DL, both OWL 1 Lite and OWL 1 DL can also be viewed as profiles of OWL 2 DL.

Underlying DL: $\mathcal{SROEL}(\mathrm{D})$

- KB consistency/satisfiability, concept subsumption, instance checking are P-complete.
- Full concept hierarchy (i.e., all subsumption relationships between atomic concepts) can be computed in one pass in polynomial time.
- Captures expressive power of many biomedical ontologies, e.g., SNOMED CT, Gene Ontology.

From $\mathcal{SROIQ}$, $\mathcal{SROEL}$ is obtained by

- allowing:
  - GCIs with concept expressions: class names, $\top$, $\bot$, $C \sqcap D$, $\exists R.C$, $\{a\}$, $\exists R$.Self
  - Complex role inclusions, range restrictions (under certain conditions)
- not allowing:
  - negation, disjunction, universal restrictions, number restrictions
  - inverse roles, role disjointness, irreflexivity, functionality and inverse functionality, symmetry and asymmetry.

OWL 2 QL

Underlying DL: DL-Lite

- **Data complexity** of conjunctive query answering is in AC^0
    - Data complexity: complexity with respect to only the size of data (ABox assertions).
    - In AC^0: conjunctive queries can be rewritten into SQL queries.
- Allows efficient implementation for accessing data on RDBMS via ontology.

From $\mathcal{SROIQ}$, DL-Lite is obtained by:

- allowing concept inclusion $C \sqsubseteq D$ where
    - C must be of the form: A, $\top$, $\bot$, $\exists R.\top$, $\exists R^{-}.\top$
    - D must be of the form: A, $\top$, $\bot$, $D_1 \sqcap D_2$, $\neg C$, $\exists R.A$, $\exists R^{-}.A$

  where C is any concept allowed on the LHS of $C \sqsubseteq D$, D is any concept allowed on the RHS of $C \sqsubseteq D$, A is concept name, D_1, D_2 are any concept allowed for D.
- allowing role hierarchy $R \sqsubseteq S$, $R^{-} \sqsubseteq S$, $R \sqsubseteq S^{-}$
- not allowing other constructs that cannot be written using any of the allowed constructs above.

Inspired by the DLs: DLP, pD* (OWL-Horst)

- Scalable rule-based reasoning suitable for OWL 2 applications (sacrificing expressive power) and RDFS applications (with additional expressive power).
- Can operate directly on RDF triples to enrich instance data.

From $\mathcal{SROIQ}$, OWL 2 RL is obtained by:

- allowing concept inclusion $C \sqsubseteq D$ where
  - C is of the form: A, $\bot$, $\{a\}$, $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\exists R.C$, $\exists R^-.C$, $\exists R.\top$
  - D is of the form: A, $\bot$, $D_1 \sqcap D_2$, $\neg C$, $\forall R.D$, $\forall R^-.D$, $\leqslant 1 R.D$, $\leqslant 1 R^-.D$, $\leqslant 1 R.\top$, $\leqslant 1 R^-.\top$, $\leqslant 0 R.D$, $\leqslant 0 R^-.D$, $\leqslant 0 R.\top$, $\leqslant 0 R^-.\top$
- allowing complex role inclusion
- disallowing union on the RHS of concept inclusion, and role reflexivity.