1. Description Logics: Introduction

2. The Description Logic $ALC$

3. Extensions of $ALC$

4. The DL $SROIQ$
Materials in this presentation are adapted from:

Outline

1. Description Logics: Introduction
2. The Description Logic ALC
3. Extensions of ALC
4. The DL SROIQ
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.
(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

1. Description Logics: Introduction
2. The Description Logic $ALC$
3. Extensions of $ALC$
4. The DL $SROIQ$
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: $\text{Student} \sqcap \forall \text{attends.GraduateCourse}$
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 \texttt{owl:Thing} \\
\( \bot \) corresponds to \texttt{owl:Nothing} \\
\( \sqcap \) corresponds to \texttt{ObjectIntersectionOf} \\
\( \sqcup \) corresponds to \texttt{ObjectUnionOf} \\
\( \neg \) corresponds to \texttt{ObjectComplementOf} \\
\( \forall \) corresponds to \texttt{ObjectAllValuesFrom} \\
\( \exists \) corresponds to \texttt{ObjectSomeValuesFrom}
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.
ABox is a set of **ABox assertions**.

- **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.

**ALC** does not support RBoxes. But more expressive DLs do support RBoxes.
• \( \mathcal{ALC} \) is a syntactic variant of the modal logic \( \mathcal{K} \) with multiple modalities.  
  ~\( \Rightarrow \) Semantics of \( \mathcal{ALC} \) can be stated using the semantics of \( \mathcal{K} \).

• \( \mathcal{ALC} \) is a fragment/sublanguage of first-order predicate logic.  
  ~\( \Rightarrow \) 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:

$$
\top^\mathcal{I} := \Delta^\mathcal{I}
$$

$$
\bot^\mathcal{I} := \emptyset
$$

$$(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I} = \{ x \in \Delta^\mathcal{I} \mid x \notin C^\mathcal{I} \}$$

$$(C \cap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$$

$$(C \cup D)^\mathcal{I} := C^\mathcal{I} \cup D^\mathcal{I}$$

$$(\forall R.C)^\mathcal{I} := \{ x \in \Delta^\mathcal{I} \mid \text{for every } y \in \Delta^\mathcal{I} : \langle x, y \rangle \in R^\mathcal{I} \text{ implies } y \in C^\mathcal{I} \}$$

$$
\leadsto \text{the set of } x \text{ for which there is no } y \text{ with } \langle x, y \rangle \in R^\mathcal{I} \text{ and } y \notin C^\mathcal{I}
$$

$$(\exists R.C)^\mathcal{I} := \{ x \in \Delta^\mathcal{I} \mid \text{there is some } y \in \Delta^\mathcal{I} \text{ with } \langle x, y \rangle \in R^\mathcal{I} \text{ and } y \in C^\mathcal{I} \}$$
A DL interpretation $\mathcal{I}$ satisfies (is a model of) axioms:

- $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ – notation: $\mathcal{I} \models C \subseteq D$
- $C \equiv D$ 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 $\alpha$ (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 $\alpha$.

$\models$ "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: “$\mathcal{K} \models ? C \subseteq \bot$”
- Concept/class inclusion (subsumption): “$\mathcal{K} \models ? C \subseteq D$”
- Concept/class equivalence: “$\mathcal{K} \models ? C \equiv D$”
- Concept/class disjointness: “$\mathcal{K} \models ? C \cap D \subseteq \bot$”
- 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:

- FOL inference procedures (resolution, tableaux) can be used
- 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)
Rose is a mother.

\[
\text{Mother}(\text{rose}) \quad \text{(DL/FOL)}
\]

\[
\text{ClassAssertion} (\text{Mother} \text{ rose}) \quad \text{(OWL)}
\]

Rose is a parent of John.

\[
\text{parentOf}(\text{rose}, \text{john}) \quad \text{(DL/FOL)}
\]

\[
\text{ObjectPropertyAssertion} (\text{parentOf} \text{ rose john}) \quad \text{(OWL)}
\]
Every mother is a parent.

\[ \text{Mother} \sqsubseteq \text{Parent} \]  
\[ \forall x (\text{Mother}(x) \rightarrow \text{Parent}(x)) \]  
\[ \text{SubClassOf(Mother Parent)} \] (FOL)

Parents are exactly either fathers or mothers.

\[ \text{Parent} \equiv \text{Father} \sqcup \text{Mother} \]  
\[ \forall x (\text{Parent}(x) \leftrightarrow (\text{Father}(x) \lor \text{Mother}(x))) \]  
\[ \text{EquivalentClasses(Parent ObjectUnionOf(Father Mother))} \] (FOL)

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

\[ \text{Male} \sqcap \text{Female} \sqsubseteq \bot \]  
\[ \forall x ((\text{Male}(x) \land \text{Female}(x)) \rightarrow \text{false}) \]  
\[ \text{SubClassOf(ObjectIntersectionOf(Male Female) owl:Nothing)} \] (FOL)
A parent is precisely someone who is a parent of some individual.

\[ \text{Parent} \equiv \exists \text{parentOf}. \top \]  
\[ \forall x (\text{Parent}(x) \leftrightarrow \exists y (\text{parentOf}(x, y))) \]  
EquivalentClasses(\text{Parent ObjectSomeValuesFrom} (\text{parentOf owl:Thing}))  

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

\[ \text{FatherWithoutSon} \equiv \text{Parent} \sqcap \neg \text{Female} \sqcap \forall \text{parentOf}. \text{Female} \]  
\[ \forall x (\text{FatherWithoutSon}(x) \leftrightarrow (\text{Parent}(x) \land \neg \text{Female}(x) \land \forall y (\text{parentOf}(x, y) \rightarrow \text{Female}(y)))) \]  
EquivalentClasses(\text{FatherWithoutSon ObjectIntersectionOf} (\text{Parent ObjectComplementOf} (\text{Female ObjectAllValuesFrom} (\text{parentOf Female}))))
Translation of $\mathcal{ALC}$ axioms into FOL only needs at most two variables

$\mapsto \mathcal{ALC}$ is a fragment of FOL with two variables (this fragment of FOL is called $\mathcal{L}_2$

$\mapsto$ Checking satisfiability of sets of $\mathcal{ALC}$ axioms is decidable.
Outline

1. Description Logics: Introduction
2. The Description Logic $ALC$
3. Extensions of $ALC$
4. The DL $SROIQ$
An **abstract** role is either a role name or an **inverse role**.

Inverse role is denoted with $R^-$ where $R$ is a role.

Semantics of inverse roles:

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

Note that $(R^-)^- = R$.

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 $\text{ObjectInverseOf}(R)$.

**Concrete roles** are used for binary relationships between an element of $\Delta^I$ and concrete literal values. In OWL, they are called **data properties**.
- 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 **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 \texttt{owl:topObjectProperty}

The **empty role** is introduced in OWL as \texttt{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: $\mathcal{ALCHI}$

$\leftrightarrow \mathcal{ALCHI}$ allows role hierarchy of the form $R \equiv S^-$, which corresponds to InverseObjectProperties(R S) in OWL.
Role transitivity axiom is of the form $\text{Trans}(R)$ where $R$ is a role.

An interpretation $\mathcal{I}$ is a model of $\text{Trans}(R)$ 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: $\forall x \forall y \forall z ((R(x, y) \land R(y, z)) \rightarrow R(x, z))$

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 $K_4$)
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

\[ \Rightarrow \] 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 \( \langle x, x_1 \rangle \in R_1^\mathcal{I}, \langle x_1, x_2 \rangle \in R_2^\mathcal{I}, \ldots, \) 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: \( \forall x \forall y (\exists x_1 \ldots \exists x_{n-1} (R_1(x, x_1) \land \cdots \land R_n(x_{n-1}, y)) \rightarrow S(x, y)) \)

- In OWL: \( \text{SubObjectPropertyOf}(\text{ObjectPropertyChain}(R_1 \ R_2 \ldots \ R_n) \ S) \)
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 $\text{Func}(R)$ where $R$ is a role.
  
  · Here, we say that $R$ is **functional**.

· An interpretation $\mathcal{I}$ is a **model of** $\text{Func}(R)$ 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): $\forall x \forall y_1 \forall y_2 ((R(x, y_1) \land R(x, y_2)) \rightarrow y_1 = y_2)$

· In OWL: FunctionalObjectProperty($R$)

· $\mathcal{ALC} + \text{role functionality} = \mathcal{ALCF}$

· If we also allow inverse roles, we obtain $\mathcal{ALCFI}$ and furthermore, we can also express **role inverse functionality** of the form $\text{Func}(R^-)$.

  · Here, we say that $R$ is **inverse functional** (or $R^-$ is functional)

  · In OWL: InverseFunctionalObjectProperty($R$)

· Interpretation $\mathcal{I}$ is a **model of** $\text{Func}(R^-)$ iff $\langle x_1, y \rangle \in R^\mathcal{I}$ and $\langle x_2, y \rangle \in R^\mathcal{I}$ together imply $x_1 = x_2$
- **Role disjointness axiom** is of the form $\text{Dis}(R, S)$ where $R, S$ must be simple roles (defined later).

- Interpretation $\mathcal{I}$ is a model of $\text{Dis}(R, S)$ 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: $\neg \exists x \exists y (R(x, y) \land S(x, y))$

- In OWL: $\text{DisjointObjectProperties}(R \ S)$
Role reflexivity axiom is of the form \( \text{Ref}(R) \) where \( R \) is a role.

Interpretation \( \mathcal{I} \) is a model of \( \text{Ref}(R) \) 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 \( \text{Irr}(R) \) where \( R \) must be a simple role (defined later).

Interpretation \( \mathcal{I} \) is a model of \( \text{Irr}(R) \) iff \( R^\mathcal{I} \) is irreflexive/anti-reflexive, i.e., \( \langle x, x \rangle \notin R^\mathcal{I} \) for every \( x \in \Delta^\mathcal{I} \).

\( \leadsto \) Being irreflexive is different from being not reflexive!

In FOL: \( \forall x. \neg R(x, x) \)

In OWL: IrreflexiveObjectProperty\( (R) \)
Role symmetry axiom is of the form $\text{Sym}(R)$ where $R$ is a role.

Interpretation $\mathcal{I}$ is a model of $\text{Sym}(R)$ 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: $\forall x (R(x, y) \rightarrow R(y, x))$

In OWL: SymmetricObjectProperty(R)

Can you express it using role hierarchies?

Role asymmetry axiom is of the form $\text{Asym}(R)$ where $R$ is a role.

Interpretation $\mathcal{I}$ is a model of $\text{Asym}(R)$ iff $R^{\mathcal{I}}$ is asymmetry, i.e., $\langle x, y \rangle \in R^{\mathcal{I}}$ implies $\langle y, x \rangle \notin R^{\mathcal{I}}$.

$\iff$ Asymmetry is different from non-symmetry and anti-symmetry!

In FOL: $\forall x (R(x, y) \rightarrow \neg R(y, x))$

In OWL: AsymmetricObjectProperty(R)

Can you express it using role disjointness axiom?
• **(Qualified) number restrictions** are concepts of the form $\leq nR.C$, $\geq nR.C$, 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 $\leq nR.C$ and $\geq nR.C$
  - If $C$ is omitted, then $C$ is actually `owl:Thing` and the number restrictions are called **unqualified number restrictions**

• Semantics:

  $$\begin{align*}
  (\leq nR.C)^I &= \{ x \in \Delta^I \mid \# \{ \langle x, y \rangle \in R^I \mid y \in C^I \} \leq n \} \\
  (\geq nR.C)^I &= \{ x \in \Delta^I \mid \# \{ \langle x, y \rangle \in R^I \mid y \in C^I \} \geq n \}
  \end{align*}$$

• Translation to FOL needs equality or counting quantifiers

• Number restrictions in OWL correspond to `ObjectMinCardinality`, `ObjectMaxCardinality`, `ObjectExactCardinality`

• $\mathcal{ALC} + \text{qualified number restrictions} = \mathcal{ALCQ}$

• $\mathcal{ALC} + \text{unqualified number restrictions} = \mathcal{ALCN}$
**Self restrictions** are concepts of the form $\exists R.\text{Self}$ where $R$ must be a simple role.

Semantics: $(\exists R.\text{Self})^I = \{ x \in \Delta^I \mid \langle x, x \rangle \in R^I \}$

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?
- \( \neg (\leq nR.C) \)
- \( \neg (\geq nR.C) \) (provided that \( n \geq 1 \))

Can you express the following concepts using number restrictions?
- \( \bot \)
- \( \top \)
- \( \exists R.C \)
- \( \forall R. \bot \)
- \( \text{Func}(R) \)

Can you express the following using concept inclusion and Self-restrictions?
- \( \text{Ref}(R) \)
- \( \text{Irr}(R) \)
A **nominal** is a concept of the form \( \{a\} \) where \( a \) is an individual name.

Semantically, \( (\{a\})^I = \{a^I\} \), i.e., it denotes the set with a single element (denoted by \( a \)).

\( \{a_1, \ldots, a_n\} \) is a shorthand for \( \{a_1\} \sqcup \cdots \sqcup \{a_n\} \).

\( \mathcal{ALC} + \text{nominals} = \mathcal{ALCO} \)

Corresponds to \texttt{ObjectOneOf}

\( \exists R.\{a\} \) corresponds to \texttt{ObjectHasValue(R a)} in OWL.
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 \( \{a\} \sqsubseteq \neg \{b\} \)
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: $C_i \cap C_j \sqsubseteq \bot$ 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 $C_i \cap C_j \sqsubseteq \bot$ for every $1 \leq i < j \leq n$.

- **ObjectPropertyDomain**($R C$)
  - In DL: $\text{dom}(R) \sqsubseteq C$.
  - Equivalent to: $\exists R. \top \sqsubseteq C$

- **ObjectPropertyRange**($R C$)
  - In DL: $\text{ran}(R) \sqsubseteq C$.
  - Equivalent to: $\top \sqsubseteq \forall R.C$
  - Equivalent to: $\exists R^- \top \sqsubseteq C$
**$\mathcal{ALC}$** Attributive Language with Complement

- **$S \ \mathcal{ALC}$** + role transitivity
- **$\mathcal{H}$** role hierarchy
- **$R$** general role inclusion axioms
- **$I$** inverse roles
- **$O$** nominals
- **$N$** unqualified number restrictions
- **$Q$** qualified number restrictions
- **$F$** functional roles
- **(D)** datatypes – in DL literature, called concrete domains
Notable DLs (some do not follow the previous nomenclature):

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

- \{\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: “Yes”

- \{(\leq 1\text{hasChild})(\text{bill})\} \models? (\forall \text{hasChild}. \text{Man})(\text{bill})
  - DL 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

{\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: “Yes”

{\text{(\leq 1 \text{hasChild})(\text{bill})}} \models (?) (\forall \text{hasChild}. \text{Man})(\text{bill})

- DL answer: “Yes”
- CWA-based 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: “Yes”

\[
\{\leq 1 \text{hasChild}(\text{bill})\} \models \forall \text{hasChild}. \text{Man}(\text{bill})
\]
- DL answer: “Yes”
- CWA-based answer: “Yes”
Outline

1. Description Logics: Introduction
2. The Description Logic $ALC$
3. Extensions of $ALC$
4. The DL $SROIQ$
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 $\exists R.\text{Self}$
- at-least restriction $\geq n R.C$
- at-most restriction $\leq n R.C$
- nominals $\{a\}, \{b\}, \ldots$

TBox axioms
- inclusion $C \subseteq D$
- 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 $R_1 \subseteq R$
- Complex inclusion $R_1 \circ \cdots \circ R_n \subseteq R$
- Transitivity $\text{Trans}(R)$
- Symmetry $\text{Sym}(R)$
- Asymmetry $\text{Asym}(R)$
- Reflexivity $\text{Ref}(R)$
- Irreflexivity $\text{Irr}(R)$
- Role disjointness $\text{Dis}(R, S)$
- Domain restriction $\text{dom}(R) \subseteq C$
- Range restriction $\text{ran}(R) \subseteq C$
- Role functionality $\text{Func}(R)$
- Role inverse functionality $\text{Func}(R^-)$
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: $SROEL(D)$

- 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 $SROIQ$, $SROEL$ is obtained by

- allowing:
  - GCIs with concept expressions: class names, $\top$, $\bot$, $C \sqcap D$, $\exists R.C$, $\{a\}$, $\exists R.\text{Self}$
  - 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.
Underlying DL: DL-Lite

- **Data complexity** of conjunctive query answering is in $\mathbf{AC}^0$
  - Data complexity: complexity with respect to only the size of data (ABox assertions).
  - In $\mathbf{AC}^0$: 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 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: $A, \bot, D_1 \sqcap D_2, \neg C, \forall R.D, \forall R^-D, \leq 1 R.D, \leq 1 R^-D, \leq 1 R.\top, \leq 1 R^-\top, \leq 0 R.D, \leq 0 R^-D, \leq 0 R.\top, \leq 0 R^-\top$
- allowing complex role inclusion
- disallowing union on the RHS of concept inclusion, and role reflexivity.