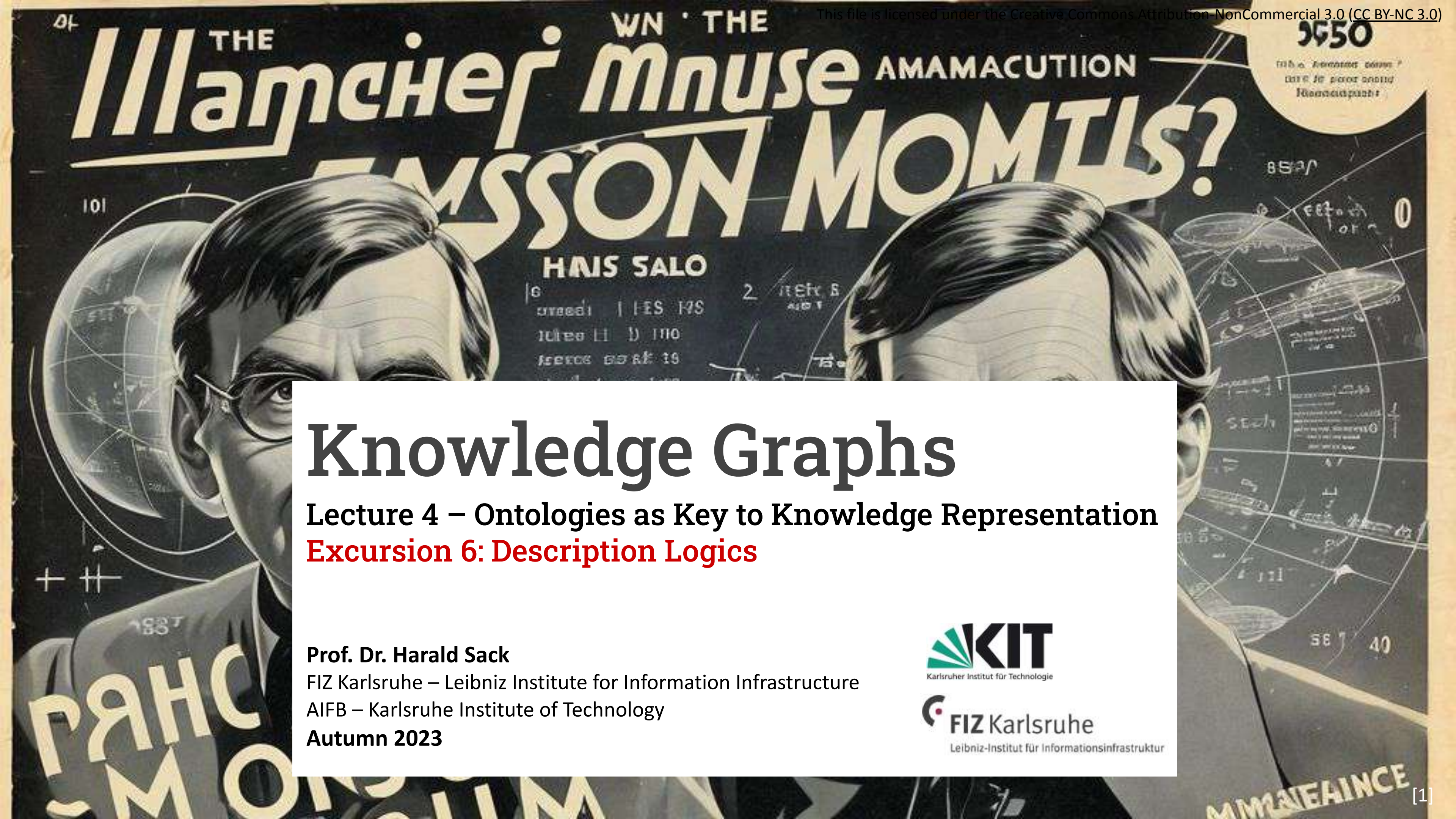


[1]

Knowledge Graphs

Lecture 4 – Ontologies as Key to Knowledge Representation

Excursion 6: Description Logics

Prof. Dr. Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure
AIFB – Karlsruhe Institute of Technology
Autumn 2023

Knowledge Graphs

Lecture 4: Ontologies as Key to Knowledge Representation

[1]

Knowledge Representation with FOL

- Why not simply take FOL for Ontologies?
- With FOL you can do everything, but...
  you could also program everything with assembler instead of higher programming languages

- FOL has high **expressivity**
- FOL is too **bulky** for modelling
- FOL is not appropriate to find consensus in modelling
- FOL proof theoretically is very **complex** (semi-decidable)
- *FOL is of course not a Markup Language for the Web*

- **Basic idea:**
  - Look for an **appropriate fragment of FOL**....
  - *...and then make it a vocabulary for RDF(S)*

Description Logics – DL

- DLs are fragments of FOL (*compromise of expressivity and scalability*)
- A DL models **concepts**, **roles** and **individuals**, as well as their relationships
- In DL from *simple descriptions* more *complex descriptions* are created with the help of ***constructors***

- DLs differ in the applied constructors (Expressivity)
- DLs have been developed from “semantic networks”
- DLs are **decidable** (most times)
- DLs possess **sufficient expressivity** (most times)
- DLs are related to modal logics
- Example for a DL:
  W3C Standard OWL 2 DL is based on description logics $\mathcal{SROIQ}(\mathcal{D})$

General DL Architecture

DL Knowledge Base

TBox $\mathcal{T}$ — **Terminological Knowledge**

Knowledge about concepts of a domain

$\texttt{Writer} \equiv \texttt{Person} \sqcap \exists \texttt{author.Book}$

ABox $\mathcal{A}$ — **Assertional Knowledge**

Knowledge about Individuals / Entities

```
Writer(GeorgeOrwell)
author(AnimalFarm, GeorgeOrwell)
```

RBox $\mathcal{R}$ — **Role-centric Knowledge**

Knowledge about roles interdependencies

$\texttt{coAuthor} \sqsubseteq \texttt{author}$

→ **Inference Engine** → **Interface**

Description Logics – DL

- DLs are a **family** of logic-based formalisms applied for knowledge representation
- $\mathcal{ALC}$ (***A**ttribute **L**anguage with **C**omplement*)
  is the smallest deductively complete DL
  - **Conjunction**, **Disjunction**, **Negation** are class constructors, denoted as $\sqcap$, $\sqcup$, $\neg$
  - Quantifiers restrict domain and range of roles

Attributive Language with Complements – $\mathcal{ALC}$

Basic Building Blocks:

- Classes
- Roles/Properties
- Individuals

- `Person(IsaacAsimov)`
  Individual IsaacAsimov is of class Person

- `Book(Foundation)`
  Individual Foundation is of class Book

- `author(Foundation, IsaacAsimov)`
  The book Foundation has the author IsaacAsimov

$\mathcal{ALC}$ – Building Blocks

- $\mathcal{ALC}$ **Atomic Types**
  - Concept names A, B, ...
  - Special concepts
    - ⊤ - Top (universal concept)
    - ⊥ - Bottom concept
  - Role names R, S, ...

- $\mathcal{ALC}$ **Constructors**
  - Negation: $\neg C$
  - Conjunction: $C \sqcap D$
  - Disjunction: $C \sqcup D$
  - Existential quantifier: $\exists R.C$
  - Universal quantifier: $\forall R.C$

$\mathcal{ALC}$ – Building Blocks

- **Class Inclusion**

  Novel ⊑ Book

  - *every novel is also a book*
  - equals FOL $(\forall x)\ (Novel(x) \rightarrow Book(x))$

- **Class Equivalence**

  Novel ≡ Prose

  - *all Prose are exactly Novels*
  - equals FOL $(\forall x)\ (Novel(x) \leftrightarrow Prose(x))$

$\mathcal{ALC}$ – Complex Class Relations

- **Conjunction** ⊓
- **Disjunction** ⊔
- **Negation** ¬

$$\text{Novel} \sqsubseteq (\text{Book} \sqcap \text{Fiction}) \sqcup (\text{Paperback} \sqcap \neg\text{Poetry})$$

$\mathcal{ALC}$

$$(\forall x)(\text{Novel}(x) \rightarrow ((\text{Book}(x) \land \text{Fiction}(x)) \lor (\text{Paperback}(x) \land \neg\text{Poetry}(x)))$$

FOL

$\mathcal{ALC}$ – Quantifiers on Roles

- **Strict Binding** of the Range of a Role to a Class
  - $\text{Book} \sqsubseteq \forall \text{author.Writer}$
  - *A Book must be authored by a Writer*
  - $(\forall x)(\text{Book}(x) \rightarrow (\forall y)(\text{author}(x,y) \rightarrow \text{Writer}(y)))$
- **Open Binding** of the Range of a Role to a Class
  - $\text{Book} \sqsubseteq \exists \text{author.Person}$
  - *Every Book has at least one author (who is a person)*
  - $(\forall x)(\text{Book}(x) \rightarrow (\exists y)(\text{author}(x,y) \land \text{Person}(y)))$

$\mathcal{ALC}$ – Formal Syntax

- Production rules for creating classes in $\mathcal{ALC}$:
  (A is an atomic class, C and D are complex Classes and R a Role)
  - $C,D ::= A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$
- An $\mathcal{ALC}$ **TBox** contains assertions of the form $C \sqsubseteq D$ and $C \equiv D$, where C,D are complex classes.
- An $\mathcal{ALC}$ **ABox** contains assertions of the form $C(a)$ and $R(a,b)$, where C is a complex Class, R a Role and a,b Individuals.
- An $\mathcal{ALC}$-**Knowledge Base** contains an ABox and a TBox.

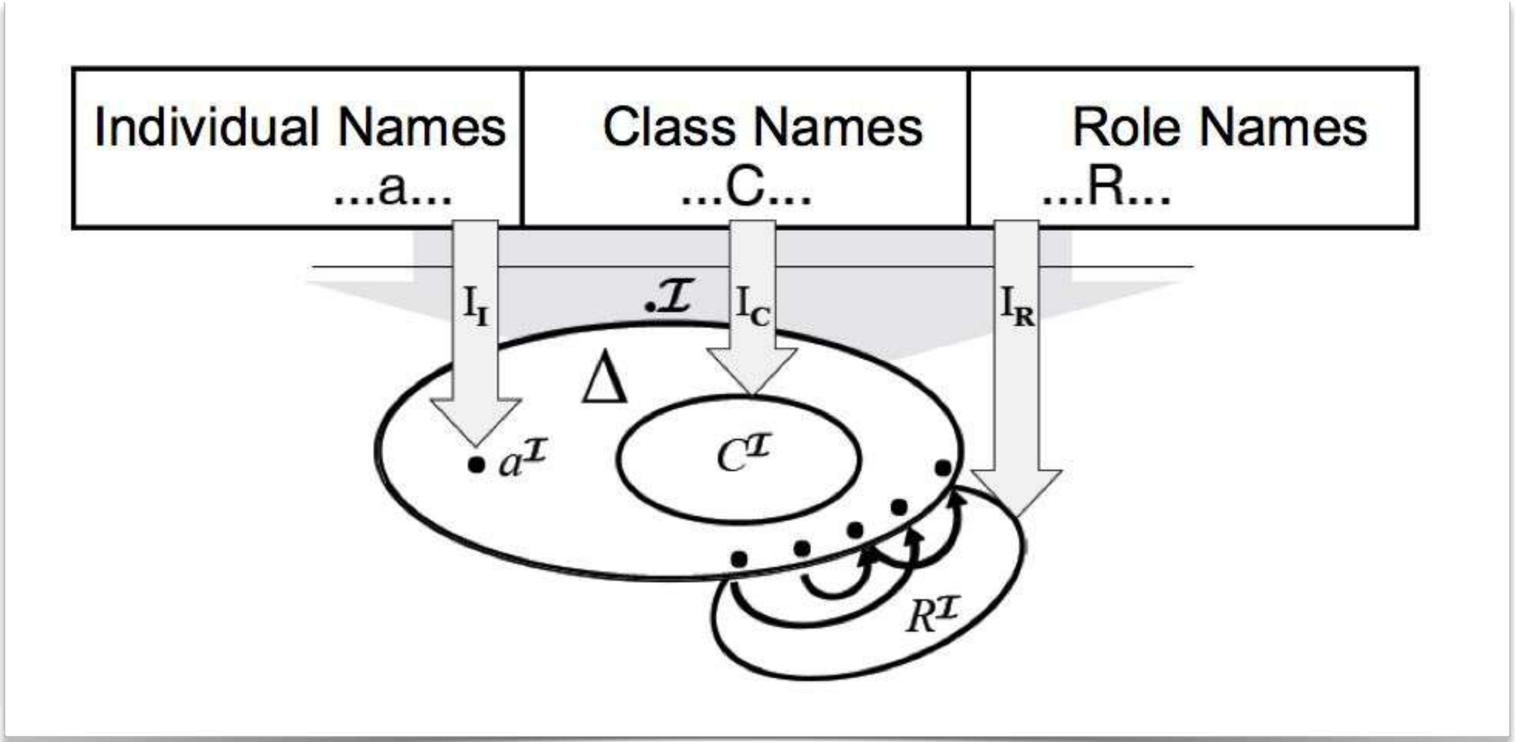
Syntax = symbols without meaning, defines rules on how to construct well-formed and valid sequences of symbols (strings)

$\mathcal{ALC}$ – Model-theoretic Semantics

- The model-theoretic semantics for $\mathcal{ALC}$ is defined via interpretations
- An **Interpretation** $I=(\Delta^I, .^I)$ contains
  - a set Δ^I (**Domain**) of individuals and
  - an **interpretation function** $.^I$ that maps
    - Individual names a to domain elements $a^I \in \Delta^I$
    - Class names C to a set of domain elements $C^I \subseteq \Delta^I$
    - Role names R to a set of pairs of domain elements $R^I \subseteq \Delta^I \times \Delta^I$

Model-theoretic Semantics performs the semantic interpretation of artificial and natural languages by "*identifying meaning with an exact and formally defined interpretation with a model*"

$\mathcal{ALC}$ – Model-theoretic Semantics



from Hitzler, Krötzsch, Rudolph: Foundations of Semantic Web Technologies, CRC Press, 2010

$\mathcal{ALC}$ - Model-theoretic Semantics

Extension (of Interpretation I) for complex classes:

- $\top^I = \Delta^I$ and $\bot^I = \emptyset$
- $(C \sqcup D)^I = C^I \cup D^I$ and $(C \sqcap D)^I = C^I \cap D^I$
- $(\neg C)^I = \Delta^I \setminus C^I$
- $\forall R.C = \{a \in \Delta^I \mid (\forall b \in \Delta^I)((a,b) \in R^I \rightarrow b \in C^I)\}$
- $\exists R.C = \{a \in \Delta^I \mid (\exists b \in \Delta^I)((a,b) \in R^I \land b \in C^I)\}$

$\mathcal{ALC}$ – Model-theoretic Semantics

Extension (of Interpretation I) for axioms:

- $C(a)$ holds, iff $a^I \in C^I$
- $R(a,b)$ holds, iff $(a^I, b^I) \in R^I$
- $C \sqsubseteq D$ holds, iff $C^I \subseteq D^I$
- $C \equiv D$ holds, iff $C^I = D^I$

Beyond $\mathcal{ALC}$ – More DL Constructors

- **Number restrictions** for roles: `≥1 children` , `≤1 mother`
- **Qualified number restrictions** for roles:
  `≥2 children.Female, ≤1 parent.Male`
- **Nominals** (definition by extension):
  `{Foundation, FoundationAndEmpire, SecondFoundation}`
- **Concrete domains** (data types): `hasAge.(≥21)`
- **Inverse roles**: $\texttt{children}^{-} \equiv \texttt{parent}$
- **Transitive roles**: $\texttt{ancestor} \sqsubseteq^{+} \texttt{ancestor}$
- **Role composition**: `parent.brother(uncle)`

Operator/Constructor	Syntax	Language	
Conjunction	$A \sqcap B$	$\mathcal{FL}$	$\mathcal{S}^{\star}$
Value Restriction	$\forall R.C$		
Existential Quantifier	$\exists R$		
Top	$\top$	$\mathcal{AL}^{\star}$	
Bottom	$\bot$		
Negation	$\neg A$		
Disjunction	$A \sqcup B$		
Existential Restriction	$\exists R.C$		
Number Restriction	$(\leq nR)$ $(\geq nR)$		
Set of Individuals	$\{a_1, ..., a_2\}$		
Role Hierarchy	$R \sqsubseteq S$	$\mathcal{H}$	
Inverse Role	R^{-1}	I	
Qualified Number Restriction	$(\leq nR.C)$ $(\geq nR.C)$	$\mathcal{Q}$	

Description Logics Family

- $\mathcal{ALC}$: Attribute Language with Complement
- $\mathcal{S}$: $\mathcal{ALC}$ + Transitivity of Roles
- $\mathcal{H}$: Role Hierarchies
- $\mathcal{O}$: Nominals
- $\mathcal{I}$: Inverse Roles
- $\mathcal{N}$: Number restrictions ≤nR etc.
- $\mathcal{Q}$: Qualified number restrictions ≤nR.C etc.
- $(\mathcal{D})$: Datatypes
- $\mathcal{F}$: Functional Roles
- $\mathcal{R}$: Role Constructors

- **OWL 2** DL is $\mathcal{SROIQ}(\mathcal{D})$

The Web Ontology Language Owl

Next Lecture...

[2]

Bibliographic References:

- Sebastian Rudolph (2011), *Foundations of Description Logics*, Karlsruhe Institute of Technology.
- Franz Baader, Ian Horrocks, and Ulrike Sattler (2008). *Description Logics*. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, Handbook of Knowledge Representation, chapter 3, pp. 135–180. Elsevier.

Picture References:

[1] “In this 1960s pulp cover picture, in the waning days of a future Galactic Empire, the mathematician Hari Seldon spends his life developing a theory of psychohistory, a new and effective mathematics of sociology. Using statistical laws of mass action, it can predict the future of large populations.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[2] “A Scifi movie poster "The Owls of Mars) depicting a huge owl sitting in the lonely red prairies of Mars in a retro futuristic rural environment of planet Mars. A rocket ship is starting in the background far away leaving contrails behind.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[2]