Symbolic Artificial Intelligence (COMP3008)
Lecture 9: Description Logic (Onthology)
24 November 2025

Daniel Karapetyan
daniel.karapetyan@nottingham.ac.uk

Updates

Project interviews and support

Project interviews going well

If you haven't booked a slot for your interview, please do this urgently (by Wednesday 26th November)

If you have an extension, you can still get support but we'll be busy during the computing classes; please arrange a meeting with me

Description Logics

Expressivity vs Performance

Logic	Expressivity	Reasoning	Decidable
FOL	Highest	Slow	No
Horn clauses (FOL case)	Medium	Medium	No
Propositional logic	Medium	Medium	Yes
CSP (finite)	Medium	Medium	Yes
Horn clauses (prop. case)	Lowest	Fast	Yes

Recap

- We have seen that FOL subsumes many other languages
- To make reasoning manageable, we can restrict it in some way
- Automated reasoning systems such as Z3, Prolog and CP-SAT (OR-Tools) usually work well enough
- Sometimes encoding knowledge was awkward

FOL for real-world knowledge bases (recap)

Encoding knowledge bases in FOL:

- We used predicates of arity 1 to identify classes of entities, e.g. $Female(x)$
- To specify relations between objects, we would usually use predicates of arity two: $Loves(x, y)$
- To define a subclass, we would use the universal quantifier:

$$\forall x.\Big(WomenInLoveWithPeter(x) \leftrightarrow Female(x) \wedge Loves(x, peter)\Big)$$

- Claiming quantities can be done using nested quantifiers:

$$\forall x.\Big(HasTwoChildren(x) \leftrightarrow \exists y \exists z.y \neq z \wedge Child(x, y) \wedge Child(x, z) \wedge \neg\exists s.s \neq y \wedge s \neq z \wedge Child(x, s)\Big)$$

Real-world oriented language

FOL issues:

- Encoding is sometimes awkward
- Overly flexible; does not enforce a solid structure of the knowledge base
- So powerful that it is untractable (undecidable)

Aim: design a less powerful language more suitable for real-world knowledge bases and with more efficient reasoning

Description Logics

Description logics (DLs) are built around classes of objects, e.g. people, animals, universities, ...

DLs have a set-like flavour:

- $\sqcap$ for intersection of classes
- $\sqcup$ for union of classes
- $\sqsubseteq$ for a subclass
- ...

Advantages of DLs:

- Intuitive
- Variable-free
- Sufficiently restricted to enable efficient reasoning

ALC

ALC

ALC stands for Attributive Concept Language with Complements

It is a basis for many descriptive logics including OWL (will discuss later)

Most of the other descriptive logics add more features to ALC

- An example of such a feature is an ‘inverse’ role, e.g. *parent* is the inverse of *child*
- Additional features generally make reasoning harder

Syntax of ALC

Logical symbols:

- Concept-forming operators $\forall$, $\exists$, $\sqcup$, $\sqcap$, $\neg$
- Connectives $\sqsubseteq$, $\equiv$
- Assertion operator ':'
- Brackets

Non-logical symbols:

- Concepts
- Roles
- Individuals

ALC primitives

Concepts also called classes, e.g. Person, Dog, Employee, ...

Equivalent to 1-arity predicates in FOL

Begin with upper-case letter

Roles defining connections between concepts, e.g. child, madeOf, ofAge, ...

Equivalent to 2-arity predicates in FOL

Begin with lower-case letter

Individuals i.e. specific instances of concepts, e.g. peter, bottle, ...

Equivalent to 0-arity functions in FOL (constants)

Begin with lower-case letter

Concepts

A concept can be defined in the following ways:

- Atomic concept: a named concept, e.g., *Person*
- $\forall r.C$, e.g. $\forall.child.Girl$ means a concept of an individual with all the children being girls
- $\exists r.C$, e.g. $\exists.child.Girl$ means a concept of an individual with at least one daughter
- $\neg C$, e.g. $\neg Female$ is a concept for not-female (note that if individuals include, e.g., computer science modules, then COMP3008 could be in $\neg Female$)
- $C \sqcap C'$, e.g. $Female \sqcap Young$ is a concept representing a girl
- $C \sqcup C'$, e.g. $Boy \sqcup Man$ is a concept representing a male

Here C and C' are concepts and r is a role

Formal interpretation of $\forall$

Let C_1 and C_2 be concepts and r be a role

The meaning of $C_2 \equiv \forall r.C_1$ literally is "C_2 is a concept that includes exactly those individuals that are r-related **only** to the individuals in the concept C_1"

Let $P_1(x)$ and $P_2(x)$ be FOL predicates equivalent to the concepts C_1 and C_2, respectively, and let $R(x, y)$ be an FOL predicate equivalent to the role r

An FOL sentence equivalent to $C_2 \equiv \forall r.C_1$ is as follows:

$$\forall x.P_2(x) \leftrightarrow \forall y.R(x, y) \rightarrow P_1(y)$$

Formal interpretation of $\exists$

The meaning of $C_2 \equiv \exists r.C_1$ literally is "C_2 is a concept that includes exactly those individuals that are r-related **to at least one** individual in the concept C_1"

An equivalent FOL sentence is as follows:

$$\forall x.P_2(x) \leftrightarrow \exists y.R(x,y) \wedge P_1(y)$$

Sentences

Sentences define relations between concepts and/or roles

- Definitions of new concepts:

$$\mathit{HasDaughter} \equiv \exists \mathit{child}.\mathit{Female}$$

- Axioms – statements of what has to hold:

$$\mathit{Mother} \sqsubseteq \mathit{Female}$$

meaning that every individual included in *Mother* is also included in *Female*

Assertions

To specify knowledge about individuals, one uses *assertions*:

- Concept assertions:

  $$\mathit{jane} : \mathit{Female}$$

  means that individual ‘jane’ is in the class ‘Female’

- Role assertion:

  $$(\mathit{jane}, \mathit{peter}) : \mathit{child}$$

  means that individuals Jane and Peter are related with the role ‘child’

TBox and ABox

ALC clearly separates data and conceptual knowledge

TBox (terminological statements):

- Describes facts that apply to the entire knowledge base – not concerned with specific individuals
- Implemented using sentences
- Can be compared to schema in a database

ABox (assertion statements):

- Knowledge about specific individuals – not concerned with general knowledge
- Implemented using assertions
- Can be compared to the content of tables in a database

Reasoning in ALC

Definitions of interpretation, entailment and satisfiability is ALC are similar to those in FOL

Complexity of ALC:

- ALC is designed in such a way that it is decidable
- Testing satisfiability in ALC is NP-complete
- Some variations have better performance guarantees
- Some extensions are undecidable in general

Semantic Web

Semantic Web

- Currently, World Wide Web contains huge amounts of high-quality data
- This data is usually not semantically organised
  - There is no straightforward approach to query this data
  - Instead, we use textual search to find appropriate sources
- The concept of *Semantic Web*
  - Organise all the information in a unified way
  - Enable a reasoning mechanism to query this data flexibly

OWL

- Developed by the World Wide Web Consortium (W3C)
- Web Ontology Language (OWL)
- Current version: OWL 2
- Part of the Semantic Web effort
- Supports several formats, most notably
  - XML-based format
  - JSON-DL (JSON Linked Data), widely used in web pages
  - Turtle format, often used by reasoners

OWL Species

OWL has three ‘species’ (levels):

Species	Expressivity	Aim
OWL Lite	Hierarchy of concepts, roles	Efficient and simple tools
OWL DL	Same as OWL Full but with some restrictions	Expressive and decidable
OWL Full	All the features	Maximum expressivity, undecidabile

Different vocabulary

Historically, the terminology of OWL is different to that of DLs:

DL	OWL
Concept	Class
Role	Property
Individual	Individual

Some features of OWL

- Distributed: an ontology can import another ontology; ontologies are inherently extendable
  - References based on URIs
  - Compatibility of independent ontologies is achieved via translations
  - Open world assumption: lack of proof is not equivalent to disproving
- Properties can relate individuals to individuals or individuals to data values (numbers, strings, dates, etc.)
  - e.g. a property can relate a human to their age
- Metaclasses (only in OWL Full) – classes of classes

Summary

- Semantic Web is an effort to formalise data on the web
- Description logics provide an appropriate mechanism for that
- OWL is developed as a practical solution to the diversity of data sources
- The technologies are still in development