Description Logic $\mathcal{ALC}$

Petr Křemen

November 20, 2025

1 Understanding $\mathcal{ALC}$

Consider the following $\mathcal{ALC}$ theory $\mathcal{K} = (\mathcal{T}, \{\})$, where $\mathcal{T}$ contains the following axioms:

$$\begin{aligned}
Man &\sqsubseteq Person \\
Woman &\sqsubseteq Person \sqcap \neg Man \\
Father &\equiv Man \sqcap \exists hasChild \cdot Person \\
GrandFather &\equiv \exists hasChild \cdot \exists hasChild \cdot \top \\
Sister &\equiv Person \sqcap \neg Man \sqcap \exists hasSibling \cdot Person
\end{aligned}$$

Ex. 1 — What is the meaning of these axioms ? Do they reflect your understanding of reality ?

Ex. 2 — Consider the following interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}})$:

$$\begin{aligned}
\Delta^{\mathcal{I}} &= Person^{\mathcal{I}} = \{B, A\} \\
Man^{\mathcal{I}} &= \{B\} \\
Woman^{\mathcal{I}} &= \{A\} \\
Father^{\mathcal{I}} &= GrandFather^{\mathcal{I}} = \{B\} \\
hasChild^{\mathcal{I}} &= \{(B, B)\} \\
hasSibling^{\mathcal{I}} &= \{\} \\
Sister^{\mathcal{I}} &= \{B\}
\end{aligned}$$

1. Is $\mathcal{I}$ a model $\mathcal{K}$? If yes, decide, whether $\mathcal{I}$ reflects reality.
2. We know that $\mathcal{ALC}$ has the *tree model property* and *finite model property*. In case $\mathcal{I}$ is a model, is $\mathcal{I}$ tree-shaped? If not, find a model that is tree-shaped.

Ex. 3 — How does the situation change when we consider $\mathcal{I}_1$ which coincides with $\mathcal{I}$, except that $Sister^{\mathcal{I}}_1 = \{\}$?

Ex. 4 — Using the vocabulary from $\mathcal{K}$, define the concept “A father having just sons.”

Ex. 5 — Using the vocabulary from $\mathcal{K}$, define the concept "A man who has no brother, but at least one sister with at least one child."

Ex. 6 — During knowledge modeling, it is often necessary to specify:

global domain and range of given role, e.g. "By $hasChild$ (role) we always connect a $Person$ (domain) with another $Person$ (range)".

local range of given role, e.g. "Every father having only sons (domain) can be connected by $hasChild$ (role) just with a Man (range)".

Show, in which way it is possible to model global domain and range of these roles in $\mathcal{ALC}$.

2 Inference Procedures

Ex. 7 — Why inconsistency of an OWL-DL ontology is a problem? What is its consequence?

Ex. 8 — Show that disjointness of two concepts can be reduced to unsatisfiability of a single concept.

Ex. 9 — A concept C is satisfiable w.r.t. $\mathcal{K}$ iff it is interpreted as a non-empty set in at least one model of $\mathcal{K}$. Is it possible to find out that C is interpreted as a non-empty set in all models of $\mathcal{K}$?

3 Tableaux Algorithm for $\mathcal{ALC}$

Ex. 10 — Decide, whether the $\mathcal{ALC}$ concept $\exists hasChild \cdot (Student \sqcap Employee) \sqcap \neg(\exists hasChild \cdot Student \sqcap \exists hasChild \cdot Employee)$ is satisfiable (w.r.t. an empty TBox). Show the run of the tableau algorithm in detail.

Ex. 11 — Decide, whether the theory/ontology $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent. Show the run of the tableau algorithm in detail.

- $\mathcal{T} = \{\exists hasChild \cdot \top \equiv Parent\}$
- $\mathcal{A} = \{hasChild(JOHN, MARY), Woman(MARY)\}$

Ex. 12 — Decide and show, whether the ontology

$$\mathcal{K}_1 = (\mathcal{T} \cup \{Parent \sqsubseteq \forall hasChild \cdot \neg Woman\}, \mathcal{A})$$

is consistent.

Ex. 13 — Decide and show, whether the ontology

$$\mathcal{K}_2 = (\mathcal{T} \cup \{Parent \sqsubseteq \exists hasChild \cdot Parent\}, \mathcal{A})$$

is consistent.

4 Practically

Ex. 14 — Go through the Protégé Crash Course on the tutorial web pages.

Ex. 15 — Model the ontology in Section 1 in Protégé and check (using the Pellet/HermiT reasoner) whether your solutions in the previous tasks were correct.

Ex. 16 — Adjust the Pizza ontology (`https://github.com/owlcs/pizza-ontology`), so that the class *IceCream* and *CheeseyVegetableTopping* become satisfiable. Explain, why the Pizza ontology is consistent, although it contains unsatisfiable classes.

Ex. 17 — Upload the original pizza ontology into GraphDB - try different repository types (OWL-Max, OWL-Horst) and see how the inferences differ (e.g. Find all kinds of food, find all kinds of CheeseyPizza). Notice the weak OWL reasoning capabilities in GraphDB – to use more complicated OWL reasoning you might export inferences using "Export inferred axioms as ontology" and import into GraphDB.