Knowledge Representation & Processing

Yizheng Zhao[1,2]

1. National Key Laboratory for Novel Software Technology
2. School of Artificial Intelligence, Nanjing University

Introduction to Description Logics & the Description Logic $\mathcal{EL}$

What are Description Logics?

There is no precise definition of what a description logic is. They form a **huge family** of logic-based **knowledge representation formalisms** with a number of common properties:

- They are descendants of **semantic networks** and **KL-ONE** from the 1960-70s.
- They describe a domain of interest in terms of
  - **concepts** (also called classes),
  - **roles** (also called relations or properties),
  - **individuals**
- Modulo a simple translation, they are subsets of predicate logic.
- Distinction between terminology and data (see next slide).

DL architecture

Knowledge Base (KB)

TBox (terminological box, schema)

$$\mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male}$$
$$\mathsf{Father} \equiv \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\top$$
...

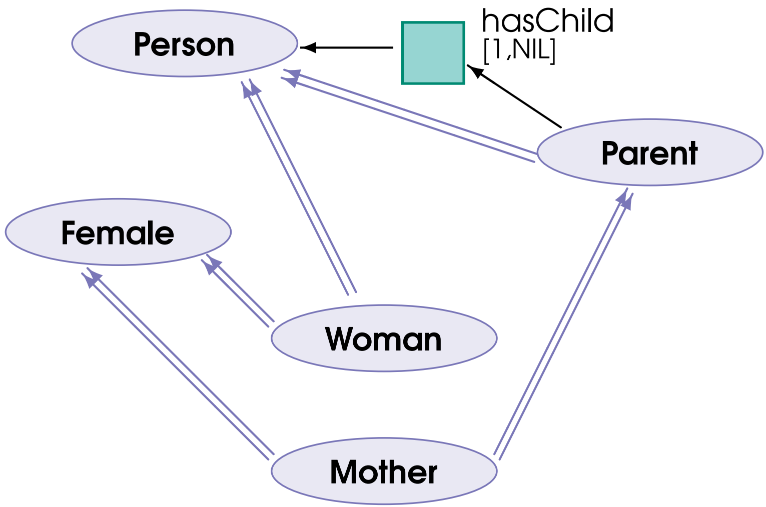
ABox (assertion box, data)

$$\mathsf{john} : \mathsf{Man}$$
$$(\mathsf{john}, \mathsf{mary}) : \mathsf{hasChild}$$
...

Reasoning System

A Semantic Network

Example: knowledge concerning persons, parents, etc.
described as a **semantic network**:



Semantic networks without a semantics!

Description Logics to be discussed

We first discuss the **terminological part** of the description logics

- $\mathcal{EL}$ (the DL underpinning OWL2 EL);
- DL-Lite (the DL underpinning OWL2 QL);
- The DL underpinning Schema.org;
- $\mathcal{ALC}$ and some extensions (the DL underpinning OWL2).

We will later discuss how description logics are used to access **instance data**.

The description logic $\mathcal{EL}$: the terminological part

Language for $\mathcal{EL}$ concepts

The language for $\mathcal{EL}$ concepts consists of:

- concept names $A_0, A_1, \ldots$

  A concept name denotes a set of objects. Typical examples are 'Person' and 'Female'. We also use A, B, B_0, B_1 ... etc as concept names.

  Concept names are also called class names.

- role names $r_0, r_1, \ldots$

  A role name denotes a set of pairs of objects. Typical examples are 'hasChild' and 'loves'. We also use r, s, s_0, s_1 ... etc as role names.

  Role names are also called property names.

- the concept $\top$ (often called "thing")

  $\top$ denotes the set of all objects in the domain.

- the concept constructor $\sqcap$. It is often called intersection, conjunction, or simply "and".

- the concept constructor $\exists$. It is often called existential restriction.

Definition of $\mathcal{EL}$ concepts

$\mathcal{EL}$ concepts are defined inductively as follows:

- all concept names are $\mathcal{EL}$ concepts
- $\top$ is a $\mathcal{EL}$ concept
- if C and D are $\mathcal{EL}$ concepts and r is a role name, then

  $$C \sqcap D, \quad \exists r.C$$

  are $\mathcal{EL}$ concepts.
- nothing else is a $\mathcal{EL}$ concept.

Examples

Assume that **Human** and **Female** are concept names and that **hasChild**, **gender**, and **hasParent** are role names. Then we obtain the following $\mathcal{EL}$ concepts:

- $\exists \mathbf{hasChild}.\top$ (somebody who has a child),
- $\mathbf{Human} \sqcap \exists \mathbf{hasChild}.\top$ (a human who has a child),
- $\mathbf{Human} \sqcap \exists \mathbf{hasChild}.\mathbf{Human}$ (a human who has a child that is human),
- $\mathbf{Human} \sqcap \exists \mathbf{gender}.\mathbf{Female}$ (a woman),
- $\mathbf{Human} \sqcap \exists \mathbf{hasChild}.\top \sqcap \exists \mathbf{hasParent}.\top$ (a human who has a child and has a parent),
- $\mathbf{Human} \sqcap \exists \mathbf{hasChild}.\exists \mathbf{gender}.\mathbf{Female}$ (a human who has a daughter),
- $\mathbf{Human} \sqcap \exists \mathbf{hasChild}.\exists \mathbf{hasChild}.\top$ (a human who has a grandchild).

Concept definitions in $\mathcal{EL}$

Let A be a concept name and C a $\mathcal{EL}$ concept. Then

- $A \equiv C$ is called a **concept definition**. C describes necessary and sufficient conditions for being an A. We sometimes read this as "A is equivalent to C".
- $A \sqsubseteq C$ is a **primitive concept definition**. C describes necessary conditions for being an A. We sometimes read this as "A is subsumed by C".

Examples:

- $\textbf{Father} \equiv \textbf{Person} \sqcap \exists \textbf{gender}.\textbf{Male} \sqcap \exists \textbf{hasChild}.\top$.
- $\textbf{Student} \equiv \textbf{Person} \sqcap \exists \textbf{is\_registered\_at}.\textbf{University}$.
- $\textbf{Father} \sqsubseteq \textbf{Person}$.
- $\textbf{Father} \sqsubseteq \exists \textbf{hasChild}.\top$.

$\mathcal{EL}$ terminology

A **$\mathcal{EL}$ terminology $\mathcal{T}$** is a finite set of definitions of the form

$$A \equiv C, \quad A \sqsubseteq C$$

such that no concept name occurs more than once on the left hand side of a definition.

So, in a terminology it is **impossible** to have two distinct definitions:

- $\textbf{University} \equiv \textbf{Institution} \sqcap \exists \textbf{grants.academicdegree}$
- $\textbf{University} \equiv \textbf{Institution} \sqcap \exists \textbf{supplies.higher\_education}$

However, we can have cyclic definitions such as

$$\textbf{Human\_being} \equiv \exists \textbf{has\_parent.Human\_being}$$

A **acyclic $\mathcal{EL}$ terminology $\mathcal{T}$** is a $\mathcal{EL}$ terminology that does not contain (even indirect) cyclic definitions.

Example: SNOMED CT (see http://www.ihtsdo.org/)

- Comprehensive healthcare terminology with approximately 400 000 definitions (400 000 concept names and 60 role names)
- Almost (except inclusions between role names) an acyclic $\mathcal{EL}$ terminology
- Property rights owned by not-for-profit organisation IHSTDO (International Health terminology Standards Development Organisation).
- IHSTDO founded in 2007. Currently owned and governed by 27 nations.
- Aim: enabling clinicians, researchers and patients to share and exchange healthcare and clinical knowledge worldwide.
- In the NHS, SNOMED CT is specified as the single terminology to be used across the health system by 2020.

SNOMED CT Snippet

$$
\begin{aligned}
\mathsf{EntireFemur} &\sqsubseteq \mathsf{StructureOfFemur} \\
\mathsf{FemurPart} &\sqsubseteq \mathsf{StructureOfFemur} \sqcap \\
&\quad \exists \mathsf{part\_of}.\mathsf{EntireFemur} \\
\mathsf{BoneStructureOfDistalFemur} &\sqsubseteq \mathsf{FemurPart} \\
\mathsf{EntireDistalFemur} &\sqsubseteq \mathsf{BoneStructureOfDistalFemur} \\
\mathsf{DistalFemurPart} &\sqsubseteq \mathsf{BoneStructureOfDistalFemur} \sqcap \\
&\quad \exists \mathsf{part\_of}.\mathsf{EntireDistalFemur} \\
\mathsf{StructureofDistalEpiphysisOfFemur} &\sqsubseteq \mathsf{DistalFemurPart} \\
\mathsf{EntireDistalEpiphysisOfFemur} &\sqsubseteq \mathsf{StructureOfDistalEpiphysisOfFemur}
\end{aligned}
$$

SNOMED CT most general concept names

- Clinical finding
- Procedure
- Observable Entity
- Body structure
- Organism
- Substance
- Biological product
- Specimen
- Physical object

Typical roles in SNOMED CT

- Finding Site. Example

$$\textbf{Kidney\_disease} \equiv \textbf{Disorder} \sqcap \exists \textbf{Finding\_Site.Kidney\_Structure}$$

- Associated Morphology. Example

$$\textbf{Bone\_marrow\_hyperplasia} \sqsubseteq \exists \textbf{Associated\_Morphology.Hyperplasia}$$

- Due to. Example

$$\textbf{Acute\_pancreatitis\_due\_to\_infection} \sqsubseteq \textbf{Acute\_pancreatitis} \sqcap \exists \textbf{Due\_to.Infection}$$

$\mathcal{EL}$ concept inclusion (CI)

We generalise $\mathcal{EL}$ concept definitions and primitive $\mathcal{EL}$ concept definitions. Let C and D be $\mathcal{EL}$ concepts. Then

- $C \sqsubseteq D$ is called a **$\mathcal{EL}$ concept inclusion**. It states that every C **is-a** D. We also say that C is subsumed by D or that D subsumes C. Sometimes we also say that C is included in D.
- $C \equiv D$ is is called a **$\mathcal{EL}$ concept equation**. We regard this as an abbreviation for the two concept inclusions $C \sqsubseteq D$ and $D \sqsubseteq C$. We sometimes read this as "C and D are equivalent".

Examples:

- $\mathbf{Disease} \sqcap \exists \mathbf{has\_location.Heart} \sqsubseteq \mathbf{NeedsTreatment}$
- $\exists \mathbf{student\_of.ComputerScience} \sqsubseteq \mathbf{Human\_being} \sqcap \exists \mathbf{knows.Programming\_Language}$

Observations

- Every $\mathcal{EL}$ concept definition is a $\mathcal{EL}$ concept equation, but not every $\mathcal{EL}$ concept equation is a $\mathcal{EL}$ concept definition.
- Every primitive $\mathcal{EL}$ concept definition is a $\mathcal{EL}$ concept inclusion, but not every $\mathcal{EL}$ concept inclusion is a primitive $\mathcal{EL}$ concept definition.

$\mathcal{EL}$ TBox

A $\mathcal{EL}$ **TBox** is a finite set $\mathcal{T}$ of $\mathcal{EL}$ concept inclusions and $\mathcal{EL}$ concept equations. Observe:

- Every acyclic $\mathcal{EL}$ terminology is a $\mathcal{EL}$ terminology;
- every $\mathcal{EL}$ terminology is a $\mathcal{EL}$ TBox.

Example:

$$
\begin{array}{rcl}
\mathsf{Pericardium} & \sqsubseteq & \mathsf{Tissue} \sqcap \exists \mathsf{cont\_in}.\mathsf{Heart} \\
\mathsf{Pericarditis} & \sqsubseteq & \mathsf{Inflammation} \sqcap \exists \mathsf{has\_loc}.\mathsf{Pericardium} \\
\mathsf{Inflammation} & \sqsubseteq & \mathsf{Disease} \sqcap \exists \mathsf{acts\_on}.\mathsf{Tissue} \\
\mathsf{Disease} \sqcap \exists \mathsf{has\_loc}.\exists \mathsf{cont\_in}.\mathsf{Heart} & \sqsubseteq & \mathsf{Heartdisease} \sqcap \mathsf{NeedsTreatment}
\end{array}
$$

How are TBoxes (eg, SNOMED CT) used?

The **concept hierarchy** induced by a TBox $\mathcal{T}$ is defined as

$$\{A \sqsubseteq B \mid A, B \text{ concept names in } \mathcal{T} \text{ and } \mathcal{T} \text{ implies } A \sqsubseteq B\}$$

Eg, the concept hierarchy induced by the SNOMED CT snippet above is

$$\begin{gathered}\mathsf{EntireDistalEpiphysisOfFemur}\\ \sqsubseteq\\ \mathsf{StructureOfDistalEpiphysisOfFemur}\\ \sqsubseteq\\ \mathsf{DistalFemurPart}\\ \sqsubseteq\\ \mathsf{BoneStructureOfDistalFemur}\\ \sqsubseteq\\ \mathsf{FemurPart}\end{gathered}$$

Standard application of SNOMED CT based on concept hierarchy

- SNOMED CT is used to produce a hierarchy of medical terms (concept names). Each term is annotated with a numerical code and an axiom defining its meaning.
- This hierachy is used by physicians to
  - generate,
  - process
  - and store

  electronic medical records (EMRs) containing diagnoses, treatments, medication, lab records, etc.

Problem: we do not yet have a precise definition of what it means that $A \sqsubseteq B$ follows from $\mathcal{T}$ (or is implied by $\mathcal{T}$). So: we do not have a precise definition of the concept hierarchy induced by a TBox.

$\mathcal{EL}$ (semantics)

- An **interpretation** is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ in which
  - $\Delta^{\mathcal{I}}$ is the **domain** (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an **interpretation function** that maps:
    * every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ $(A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}})$
    * every role name r to a binary relation $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$ $(r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$
- The interpretation $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of an arbitrary $\mathcal{EL}$ concept C in $\mathcal{I}$ is defined inductively:
  - $(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}$
  - $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
  - $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid$ exists $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in r^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$

Example

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where

- $\Delta^{\mathcal{I}} = \{a, b, c, d, A, B\}$;
- $\mathsf{Person}^{\mathcal{I}} = \{a, b, c, d\}$, $\mathsf{Female}^{\mathcal{I}} = \{A\}$;
- $\mathsf{hasChild}^{\mathcal{I}} = \{(a, b), (b, c)\}$, $\mathsf{gender}^{\mathcal{I}} = \{(a, A), (b, B), (c, A)\}$.

Compute:

- $(\mathsf{Person} \sqcap \exists \mathsf{gender}.\top)^{\mathcal{I}}$,
- $(\mathsf{Person} \sqcap \exists \mathsf{gender}.\mathsf{Female})^{\mathcal{I}}$,
- $(\mathsf{Person} \sqcap \exists \mathsf{hasChild}.\mathsf{Person})^{\mathcal{I}}$,
- $(\mathsf{Person} \sqcap \exists \mathsf{hasChild}.\exists \mathsf{gender}.\mathsf{Female})^{\mathcal{I}}$,
- $(\mathsf{Person} \sqcap \exists \mathsf{hasChild}.\exists \mathsf{hasChild}.\top)^{\mathcal{I}}$.

Semantics: when is a concept inclusion true in an interpretation?

Let $\mathcal{I}$ be an interpretation, $C \sqsubseteq D$ a concept inclusion, and $\mathcal{T}$ a TBox.

- We write $\mathcal{I} \models C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$. If this is the case, then we say that
  - $\mathcal{I}$ satisfies $C \sqsubseteq D$ or, equivalently,
  - $C \sqsubseteq D$ is true in $\mathcal{I}$ or, equivalently,
  - $\mathcal{I}$ is a model of $C \sqsubseteq D$.
- We write $\mathcal{I} \models C \equiv D$ if $C^\mathcal{I} = D^\mathcal{I}$
- We write $\mathcal{I} \models \mathcal{T}$ if $\mathcal{I} \models E \sqsubseteq F$ for all $E \sqsubseteq F$ in $\mathcal{T}$. If this is the case, then we say that
  - $\mathcal{I}$ satisfies $\mathcal{T}$ or, equivalently,
  - $\mathcal{I}$ is a model of $\mathcal{T}$.

Semantics: when does a concept inclusion follow from a TBox?

Let $\mathcal{T}$ be a TBox and $C \sqsubseteq D$ a concept inclusion. We say that **$C \sqsubseteq D$ follows from $\mathcal{T}$** if, and only if, every model of $\mathcal{T}$ is a model of $C \sqsubseteq D$.

Instead of saying that $C \sqsubseteq D$ follows from $\mathcal{T}$ we often write

- $\mathcal{T} \models C \sqsubseteq D$ or
- $C \sqsubseteq_{\mathcal{T}} D$.

Example: let MED be the $\mathcal{EL}$ TBox

$$
\begin{array}{rcl}
\textbf{Pericardium} & \sqsubseteq & \textbf{Tissue} \sqcap \exists \textbf{cont\_in}.\textbf{Heart} \\
\textbf{Pericarditis} & \sqsubseteq & \textbf{Inflammation} \sqcap \exists \textbf{has\_loc}.\textbf{Pericardium} \\
\textbf{Inflammation} & \sqsubseteq & \textbf{Disease} \sqcap \exists \textbf{acts\_on}.\textbf{Tissue} \\
\textbf{Disease} \sqcap \exists \textbf{has\_loc}.\exists \textbf{cont\_in}.\textbf{Heart} & \sqsubseteq & \textbf{Heartdisease} \sqcap \textbf{NeedsTreatment}
\end{array}
$$

Pericarditis needs treatment if, and only if, $\textbf{Percarditis} \sqsubseteq_{\mathsf{MED}} \textbf{NeedsTreatment}$.

Examples

Let $\mathcal{T} = \{A \sqsubseteq \exists r.B\}$. Then

$$\mathcal{T} \not\models A \sqsubseteq B.$$

To see this, construct an interpretation $\mathcal{I}$ such that

- $\mathcal{I} \models \mathcal{T}$;
- $\mathcal{I} \not\models A \sqsubseteq B$.

Namely, let $\mathcal{I}$ be defined by

- $\Delta^{\mathcal{I}} = \{a, b\}$;
- $A^{\mathcal{I}} = \{a\}$;
- $r^{\mathcal{I}} = \{(a, b)\}$;
- $B^{\mathcal{I}} = \{b\}$.

Then $A^{\mathcal{I}} = \{a\} \subseteq \{a\} = (\exists r.B)^{\mathcal{I}}$ and so $\mathcal{I} \models \mathcal{T}$. But $A^{\mathcal{I}} \not\subseteq B^{\mathcal{I}}$ and so $\mathcal{I} \not\models A \sqsubseteq B$.

Examples

Let again $\mathcal{T} = \{A \sqsubseteq \exists r.B\}$. Then

$$\mathcal{T} \not\models \exists r.B \sqsubseteq A.$$

To see this, construct an interpretation $\mathcal{I}$ such that

- $\mathcal{I} \models \mathcal{T}$;
- $\mathcal{I} \not\models \exists r.B \sqsubseteq A$.

Let $\mathcal{I}$ be defined by

- $\Delta^{\mathcal{I}} = \{a\}$;
- $A^{\mathcal{I}} = \emptyset$;
- $r^{\mathcal{I}} = \{(a, a)\}$;
- $B^{\mathcal{I}} = \{a\}$.

Then $A^{\mathcal{I}} = \emptyset \subseteq \{a\} = (\exists r.B)^{\mathcal{I}}$ and so $\mathcal{I} \models \mathcal{T}$. But $(\exists r.B)^{\mathcal{I}} = \{a\} \not\subseteq \emptyset = A^{\mathcal{I}}$ and so $\mathcal{I} \not\models \exists r.B \sqsubseteq A$.

Deciding whether $C \sqsubseteq_{\mathcal{T}} D$ for $\mathcal{EL}$ TBoxes $\mathcal{T}$

We give a polynomial time (tractable) algorithm deciding whether $C \sqsubseteq_{\mathcal{T}} D$

The algorithm actually decides whether $A \sqsubseteq_{\mathcal{T}} B$ only for concept names A and B in $\mathcal{T}$.

This is sufficient because the following two conditions are equivalent:

- $C \sqsubseteq_{\mathcal{T}} D$
- $A \sqsubseteq_{\mathcal{T}'} B$, where A and B are concept names that do not occur in $\mathcal{T}$ and the TBox $\mathcal{T}'$ is defined by

$$\mathcal{T}' = \mathcal{T} \cup \{A \equiv C, B \equiv D\}$$

Thus, if we want to know whether $C \sqsubseteq_{\mathcal{T}} D$, we first construct $\mathcal{T}'$ and then apply the algorithm to $\mathcal{T}'$, A, and B.

Pre-processing

A $\mathcal{EL}$ TBox is in *normal form* if it consists of inclusions of the form

(sform) $A \sqsubseteq B$, where A and B are concept names;

(cform) $A_1 \sqcap A_2 \sqsubseteq B$, where A_1, A_2, B are concept names;

(rform) $A \sqsubseteq \exists r.B$, where A, B are concept names;

(lform) $\exists r.A \sqsubseteq B$, where A, B are concept names.

Given a $\mathcal{EL}$ Box $\mathcal{T}$, one can compute in polynomial time a TBox $\mathcal{T}'$ in normal form such that for all concept names A, B in $\mathcal{T}$:

$$A \sqsubseteq_{\mathcal{T}} B \quad \Leftrightarrow \quad A \sqsubseteq_{\mathcal{T}'} B.$$

Algorithm for Pre-processing

Given a TBox $\mathcal{T}$, apply the following rules exhaustively:

- Replace each $C_1 \equiv C_2$ by $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$;
- Replace each $C \sqsubseteq C_1 \sqcap C_2$ by $C \sqsubseteq C_1$ and $C \sqsubseteq C_2$;
- If $\exists r.C$ occurs in $\mathcal{T}$ and C is complex, replace C in $\mathcal{T}$ by a fresh concept name X and add $X \sqsubseteq C$ and $C \sqsubseteq X$ to $\mathcal{T}$;
- If $C \sqsubseteq D$ in $\mathcal{T}$ and $\exists r.B$ occurs in C (but $C \neq \exists r.B$), then remove $C \sqsubseteq D$, take a fresh concept name X, and add

  $$X \sqsubseteq \exists r.B, \quad \exists r.B \sqsubseteq X, \quad C' \sqsubseteq D$$

  to $\mathcal{T}$, where C' is the concept obtained from C by replacing $\exists r.B$ by X.

Algorithm for Pre-processing

- If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq D$ in $\mathcal{T}$ and $n > 2$, then remove it, take a fresh concept name X, and add

$$A_2 \sqcap \cdots \sqcap A_n \sqsubseteq X, \quad X \sqsubseteq A_2 \sqcap \cdots \sqcap A_n, \quad A_1 \sqcap X \sqsubseteq D$$

  to $\mathcal{T}$.

- If $\exists r.B \sqsubseteq \exists s.E$ in $\mathcal{T}$, then remove it, take a fresh concept name X, and add

$$\exists r.B \sqsubseteq X, \quad X \sqsubseteq \exists s.E$$

  to $\mathcal{T}$.

Pre-Processing: Example

Consider $\mathcal{T}$:

$$A_0 \sqsubseteq B \sqcap \exists r.B', \quad A_1 \sqcap \exists r.B \sqsubseteq A_2$$

Step 1 gives:

$$A_0 \sqsubseteq B, \quad A_0 \sqsubseteq \exists r.B', \quad A_1 \sqcap \exists r.B \sqsubseteq A_2$$

Step 4 gives:

$$\begin{aligned}
A_0 &\sqsubseteq B \\
A_0 &\sqsubseteq \exists r.B' \\
A_1 \sqcap X &\sqsubseteq A_2 \\
\exists r.B &\sqsubseteq X \\
X &\sqsubseteq \exists r.B
\end{aligned}$$

Pre-Processing applied to Example MED

$$
\begin{aligned}
\text{Pericardium} &\sqsubseteq \text{Tissue} \\
\text{Pericardium} &\sqsubseteq Y \\
\text{Pericarditis} &\sqsubseteq \text{Inflammation} \\
\text{Pericarditis} &\sqsubseteq \exists \text{has\_loc}.\text{Pericardium} \\
\text{Inflammation} &\sqsubseteq \text{Disease} \\
\text{Inflammation} &\sqsubseteq \exists \text{acts\_on}.\text{Tissue} \\
\text{Disease} \sqcap X &\sqsubseteq \text{Heartdisease} \\
\text{Disease} \sqcap X &\sqsubseteq \text{NeedsTreatment}
\end{aligned}
$$

$$
\exists \text{has\_loc}.Y \sqsubseteq X, X \sqsubseteq \exists \text{has\_loc}.Y, \quad \exists \text{cont\_in}.\text{Heart} \sqsubseteq Y, Y \sqsubseteq \exists \text{cont\_in}.\text{Heart}
$$

Algorithm deciding $A \sqsubseteq_{\mathcal{T}} B$: Intuition

Given $\mathcal{T}$ in normal form, we compute functions S and R:

- S maps every concept name A from $\mathcal{T}$ to a set of concept names B;
- R maps every role name r from $\mathcal{T}$ to a set of pairs (B_1, B_2) of concept names.

We will have $A \sqsubseteq_{\mathcal{T}} B$ if, and only if, $B \in S(A)$.

Intuitively, we construct an interpretation $\mathcal{I}$ with

- $\Delta^{\mathcal{I}}$ is the set of concept names in $\mathcal{T}$.
- $A^{\mathcal{I}}$ is the set of all B such that $A \in S(B)$;
- $r^{\mathcal{I}}$ is the set of all $(A, B) \in R(r)$.

This will be a model of $\mathcal{T}$ and $A \sqsubseteq_{\mathcal{T}} B$ if, and only if, $A \in B^{\mathcal{I}}$.

Algorithm

Input: $\mathcal{T}$ in normal form. Initialise: $S(A) = \{A\}$ and $R(r) = \emptyset$ for A and r in $\mathcal{T}$.

Apply the following four rules to S and R exhaustively:

(simpleR) If $A' \in S(A)$ and $A' \sqsubseteq B \in \mathcal{T}$ and $B \notin S(A)$, then

$$S(A) := S(A) \cup \{B\}.$$

(conjR) If $A_1, A_2 \in S(A)$ and $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$ and $B \notin S(A)$, then

$$S(A) := S(A) \cup \{B\}.$$

(rightR) If $A' \in S(A)$ and $A' \sqsubseteq \exists r.B \in \mathcal{T}$ and $(A, B) \notin R(r)$, then

$$R(r) := R(r) \cup \{(A, B)\}.$$

(leftR) If $(A, B) \in R(r)$ and $B' \in S(B)$ and $\exists r.B' \sqsubseteq A' \in \mathcal{T}$ and $A' \notin S(A)$, then

$$S(A) := S(A) \cup \{A'\}.$$

Example

$$\begin{aligned} A_0 &\sqsubseteq \exists r.B \\ B &\sqsubseteq E \\ \exists r.E &\sqsubseteq A_1 \end{aligned}$$

Initialise: $S(A_0) = \{A_0\}$, $S(A_1) = \{A_1\}$, $S(B) = \{B\}$, $S(E) = \{E\}$, $R(r) = \emptyset$.

- Application of (rightR) and axiom 1 gives: $R(r) = \{(A_0, B)\}$;
- Application of (simpleR) and axiom 2 gives: $S(B) = \{B, E\}$;
- Application of (leftR) and axiom 3 gives: $S(A_0) = \{A_0, A_1\}$;
- No more rules are applicable.

Thus, $R(r) = \{(A_0, B)\}$, $S(B) = \{B, E\}$, $S(A_0) = \{A_0, A_1\}$ and no changes for the remaining values. We obtain $A_0 \sqsubseteq_{\mathcal{T}} A_1$.

Fragment of MED

$$
\begin{array}{rcl}
\textbf{Pericardium (Pm)} & \sqsubseteq & Y \\
\textbf{Pericarditis (Ps)} & \sqsubseteq & \textbf{Inflammation (Inf)} \\
\textbf{Ps} & \sqsubseteq & \exists \textbf{has\_loc.Pm} \\
\textbf{Inf} & \sqsubseteq & \textbf{Disease (Dis)} \\
\textbf{Disease} \sqcap X & \sqsubseteq & \textbf{NeedsTreatment} \\
\exists \textbf{has\_loc.}Y & \sqsubseteq & X
\end{array}
$$

Partial run of the algorithm (showing that $\textbf{Ps} \sqsubseteq_{\mathsf{MED}} \textbf{NeedsTreatment}$):

- Applications of (simpleR) give: $S(\textbf{Pm}) = \{Y, \textbf{Pm}\}$, $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}\}$;
- Application of (rightR) give: $R(\textbf{has\_loc}) = \{(\textbf{Ps}, \textbf{Pm})\}$,
- Application of (leftR) gives: $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}, X\}$
- Application of (conjR) gives: $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}, X, \textbf{NeedsTreatment}\}$

Analysing the output of the algorithm

Let $\mathcal{T}$ be in normal form and S, R the output of the algorithm.

Theorem. For all concept names A, B in $\mathcal{T}$: $A \sqsubseteq_{\mathcal{T}} B$ if, and only if, $B \in S(A)$.

In fact, the following holds: Define an interpretation $\mathcal{I}$ by

- $\Delta^{\mathcal{I}}$ is the set of concept names in $\mathcal{T}$.
- $A^{\mathcal{I}}$ is the set of all B such that $A \in S(B)$;
- $r^{\mathcal{I}}$ is the set of all $(A, B) \in R(r)$.

Then

- $\mathcal{I}$ satisfies $\mathcal{T}$ and
- for all concept names A from $\mathcal{T}$ and $\mathcal{EL}$-concepts C:

$$A \sqsubseteq_{\mathcal{T}} C \quad \Leftrightarrow \quad A \in C^{\mathcal{I}}.$$