Description Logics: background
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), and individuals.
- Modulo a simple translation, they are subsets of predicate logic.
- Distinction between terminology and data (see next slide).
Knowledge Base (KB)

**TBox** (terminological box, schema)

\[
\text{Man} \equiv \text{Human} \cap \text{Male} \\
\text{Father} \equiv \text{Man} \cap \exists \text{hasChild. } T \\
... 
\]

**ABox** (assertion box, data)

\[
\text{john} : \text{Man} \\
(\text{john, mary}) : \text{hasChild} \\
... 
\]
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

- **EL** (the DL underpinning OWL2 EL);
- DL-Lite (the DL underpinning OWL2 QL);
- The DL underpinning Schema.org;
- **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 $A, B, B_0, B_1 \ldots$ 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 $r, s, s_0, s_1 \ldots$ 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
  \[ C \sqcap D, \exists r.C \]
  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:

- $\exists\text{hasChild}.\top$ (somebody who has a child),

- $\text{Human} \sqcap \exists\text{hasChild}.\top$ (a human who has a child),

- $\text{Human} \sqcap \exists\text{hasChild.}\text{Human}$ (a human who has a child that is human),

- $\text{Human} \sqcap \exists\text{gender.}\text{Female}$ (a woman),

- $\text{Human} \sqcap \exists\text{hasChild.}\top \sqcap \exists\text{hasParent.}\top$ (a human who has a child and has a parent),

- $\text{Human} \sqcap \exists\text{hasChild.}\exists\text{gender.}\text{Female}$ (a human who has a daughter),

- $\text{Human} \sqcap \exists\text{hasChild.}\exists\text{hasChild.}\top$ (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:

- $\text{Father} \equiv \text{Person} \sqcap \exists \text{gender} . \text{Male} \sqcap \exists \text{hasChild} . \top$.

- $\text{Student} \equiv \text{Person} \sqcap \exists \text{is_registered_at} . \text{University}$.

- $\text{Father} \sqsubseteq \text{Person}$.

- $\text{Father} \sqsubseteq \exists \text{hasChild} . \top$. 

Ontology Languages
EL terminology

A **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:

- University $\equiv$ Institution $\sqcap$ $\exists$grants.academicdegree
- University $\equiv$ Institution $\sqcap$ $\exists$supplies.higher_education

However, we can have cyclic definitions such as

$$\text{Human\_being} \equiv \exists\text{has\_parent}\cdot\text{Human\_being}$$

A **acyclic EL terminology** $\mathcal{T}$ is a 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 E L terminology
- Property rights owned by not-for-profit organisation IHSTDO (International Health terminology Standards Development Organisation).
- 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

EntireFemur ⊆ StructureOfFemur

FemurPart ⊆ StructureOfFemur

∃part_of.EntireFemur

BoneStructureOfDistalFemur ⊆ FemurPart

EntireDistalFemur ⊆ BoneStructureOfDistalFemur

DistalFemurPart ⊆ BoneStructureOfDistalFemur

∃part_of.EntireDistalFemur

StructureofDistalEpiphysisOfFemur ⊆ DistalFemurPart

EntireDistalEpiphysisOfFemur ⊆ StructureOfDistalEpiphysisOfFemur
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

  \[ \text{Kidney \_disease} \equiv \text{Disorder} \sqcap \exists \text{Finding \_Site. Kidney \_Structure} \]

- Associated Morphology. Example

  \[ \text{Bone \_marrow \_hyperplasia} \sqsubseteq \exists \text{Associated \_Morphology. Hyperplasia} \]

- Due to. Example

  \[ \text{Acute \_pancreatitis \_due \_to \_infection} \sqsubseteq \text{Acute \_pancreatitis} \sqcap \exists \text{Due \_to. Infection} \]
EL concept inclusion (CI)

We generalise EL concept definitions and primitive EL concept definitions. Let $C$ and $D$ be EL concepts. Then

- $C \sqsubseteq D$ is called a **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 **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:

- Disease $\sqcap \exists$has_location.Heart $\sqsubseteq$ NeedsTreatment

- $\exists$student_of.ComputerScience $\sqsubseteq$ Human_being$\sqcap \exists$knows.Programming_Language
Observations

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

- Every primitive $\text{EL}$ concept definition is a $\text{EL}$ concept inclusion, but not every $\text{EL}$ concept inclusion is a primitive $\text{EL}$ concept definition.
\textbf{\(EL\) TBox}

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

- Every acyclic \(EL\) terminology is a \(EL\) terminology;
- every \(EL\) terminology is a \(EL\) TBox.

Example:

\begin{align*}
\text{Pericardium} & \sqsubseteq \text{Tissue} \sqcap \exists \text{cont_in.Heart} \\
\text{Pericarditis} & \sqsubseteq \text{Inflammation} \sqcap \exists \text{has_loc.Pericardium} \\
\text{Inflammation} & \sqsubseteq \text{Disease} \sqcap \exists \text{acts_on.Tissue} \\
\text{Disease} \sqcap \exists \text{has_loc.}\exists \text{cont_in.Heart} & \sqsubseteq \text{Heartdisease} \sqcap \text{NeedsTreatment}
\end{align*}
How are TBoxes (e.g., SNOMED CT) used?

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

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

E.g., the concept hierarchy induced by the SNOMED CT snippet above is

$\text{EntireDistalEpiphysisOfFemur} \sqsubseteq \text{StructureOfDistalEpiphysisOfFemur} \sqsubseteq \text{DistalFemurPart} \sqsubseteq \text{BoneStructureOfDistalFemur} \sqsubseteq \text{FemurPart}$
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 hierarchy 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 $A \subseteq B$ follows from $T$ (or is implied by $T$). So: we do not have a precise definition of the concept hierarchy induced by a TBox.
**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 **EL** concept $C$ in $\mathcal{I}$ is defined inductively:
  - $(\top)^\mathcal{I} = \Delta^\mathcal{I}$
  - $(C \cap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
  - $(\exists r.C)^\mathcal{I} = \{ x \in \Delta^\mathcal{I} | \text{exists } y \in \Delta^\mathcal{I} \text{ such that } (x, y) \in r^\mathcal{I} \text{ and } y \in C^\mathcal{I} \}$
Example

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

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

Compute:

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

- $\text{Pericardium} \sqsubseteq \text{Tissue} \sqcap \exists \text{cont_in.Heart}$
- $\text{Pericarditis} \sqsubseteq \text{Inflammation} \sqcap \exists \text{has_loc.Pericardium}$
- $\text{Inflammation} \sqsubseteq \text{Disease} \sqcap \exists \text{acts_on.Tissue}$
- $\text{Disease} \sqcap \exists \text{has_loc.\exists cont_in.Heart} \sqsubseteq \text{Heartdisease} \sqcap \text{NeedsTreatment}$

Pericarditis needs treatment if, and only if, $\text{Pericarditis} \sqsubseteq_{\text{MED}} \text{NeedsTreatment}$. 
Examples

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

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

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

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

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 $\mathcal{I} \not\models A \subseteq B$. 

Ontology Languages
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 $C \sqsubseteq_T D$ for EL TBoxes $\mathcal{T}$

We give a polynomial time (tractable) algorithm deciding whether $C \sqsubseteq_T D$.

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

This is sufficient because the following two conditions are equivalent:

- $C \sqsubseteq_T D$
- $A \sqsubseteq_{\mathcal{T}'} B$, 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 $C \sqsubseteq_T D$, 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) $A_1 \cap A_2 \sqsubseteq B$, 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 $\mathcal{T}$:

$$A \sqsubseteq_{\mathcal{T}} B \iff A \sqsubseteq_{\mathcal{T}'} B.$$
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 $C \sqsubseteq C_1 \cap C_2$ 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 $X$. 

Ontology Languages
Algorithm for Pre-processing

- If $A_1 \cap \cdots \cap A_n \subseteq D$ in $\mathcal{T}$ and $n > 2$, then remove it, take a fresh concept name $X$, and add
  \[ A_2 \cap \cdots \cap A_n \subseteq X, \quad X \subseteq A_2 \cap \cdots \cap A_n, \quad A_1 \cap X \subseteq D \]
  to $\mathcal{T}$.

- If $\exists r.B \subseteq \exists s.E$ in $\mathcal{T}$, then remove it, take a fresh concept name $X$, and add
  \[ \exists r.B \subseteq X, \quad X \subseteq \exists s.E \]
  to $\mathcal{T}$.
Pre-Processing: Example

Consider $\mathcal{T}$:

$$A_0 \subseteq B \cap \exists r.B', \quad A_1 \cap \exists r.B \subseteq A_2$$

Step 1 gives:

$$A_0 \subseteq B, \quad A_0 \subseteq \exists r.B', \quad A_1 \cap \exists r.B \subseteq A_2$$

Step 4 gives:

$$A_0 \subseteq B$$

$$A_0 \subseteq \exists r.B'$$

$$A_1 \cap X \subseteq A_2$$

$$\exists r.B \subseteq X$$

$$X \subseteq \exists r.B$$
Pre-Processing applied to Example MED

Pericardium ⊑ Tissue
Pericardium ⊑ Y
Pericarditis ⊑ Inflammation
Pericarditis ⊑ ∃has_loc.Pericardium
Inflammation ⊑ Disease
Inflammation ⊑ ∃acts_on.Tissue
Disease ⊓ X ⊑ Heartdisease
Disease ⊓ X ⊑ NeedsTreatment
∃has_loc.Y ⊑ X, X ⊑ ∃has_loc.Y, ∃cont_in.Heart ⊑ Y, Y ⊑ ∃cont_in.Heart
Algorithm deciding $A \sqsubseteq_T B$: 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 $A \sqsubseteq_T B$ 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 $A \sqsubseteq_T B$ 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 $B \not\in S(A)$, then

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

(conjR) If $A_1, A_2 \in S(A)$ and $A_1 \cap A_2 \sqsubseteq B \in \mathcal{T}$ and $B \not\in S(A)$, then

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

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

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

\[ A_0 \sqsubseteq \exists r.B \]
\[ B \sqsubseteq E \]
\[ \exists r.E \sqsubseteq A_1 \]

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 \( A_0 \sqsubseteq_T A_1 \).
Partial run of the algorithm (showing that $\text{Ps} \sqsubseteq_{\text{MED}} \text{NeedsTreatment}$):

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

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