

# ONTOLOGIES & DESCRIPTION LOGICS

Part of Logic and Languages Module

---

**Meghyn Bienvenu** (LaBRI - CNRS & *Université de Bordeaux*)

## REASONING WITH LIGHTWEIGHT DLS

---

Some applications require **very large ontologies and/or data**

Scalability concerns led to proposal of **DLs with lower complexity**

**$\mathcal{EL}$  family of DLs** (basis for **OWL 2 EL**)

- designed to allow **efficient reasoning with large ontologies**
- key technique: **saturation** ( $\sim$  forward chaining)

**DL-Lite family of DLs** (basis for **OWL 2 QL**)

- designed for ontology-mediated query answering
- key technique: **query rewriting** ( $\sim$  backward chaining)

## REASONING IN EL

---

The **logic  $\mathcal{EL}$**  and its extensions are designed for applications requiring **very large ontologies**.

This family of DLs is **well suited for biomedical applications**.

Examples of large biomedical ontologies:

- **GO (Gene Ontology)**, around 20,000 concepts
- **NCI (cancer ontology)**, around 30,000 concepts
- **SNOMED (medical ontology)**, over 350,000 concepts (!)

$\text{Pericarditis} \sqsubseteq \text{Inflammation} \sqcap \exists \text{loc. Pericardium}$   
 $\text{Pericardium} \sqsubseteq \text{Tissue} \sqcap \exists \text{partOf. Heart} \quad \text{Inflammation} \sqsubseteq \text{Disease}$   
 $\text{Disease} \sqcap \exists \text{loc.} \exists \text{partOf. Heart} \sqsubseteq \text{HeartDisease}$

In  $\mathcal{EL}$ , complex concepts are built as follows:

$$C := \top \mid A \mid C_1 \sqcap C_2 \mid \exists r.C$$

Only concept inclusions  $C_1 \sqsubseteq C_2$  in the TBox

In  $\mathcal{EL}$ , complex concepts are built as follows:

$$C := \top \mid A \mid C_1 \sqcap C_2 \mid \exists r.C$$

Only concept inclusions  $C_1 \sqsubseteq C_2$  in the TBox

Some possible extensions:

- $\perp$  (to express **disjoint classes**)
- **domain restrictions**  $\text{dom}(r) \sqsubseteq C$
- **range restrictions**  $\text{range}(r) \sqsubseteq C$
- **complex role inclusions**  $r_1 \circ \dots \circ r_n \sqsubseteq r_{n+1}$  (**transitivity**:  $r \circ r \sqsubseteq r$ )

**OWL 2 EL** profile includes all these extensions

In  $\mathcal{EL}$ , complex concepts are built as follows:

$$C := \top \mid A \mid C_1 \sqcap C_2 \mid \exists r.C$$

Only concept inclusions  $C_1 \sqsubseteq C_2$  in the TBox

Some possible extensions:

- $\perp$  (to express **disjoint classes**)
- **domain restrictions**  $\text{dom}(r) \sqsubseteq C$
- **range restrictions**  $\text{range}(r) \sqsubseteq C$
- **complex role inclusions**  $r_1 \circ \dots \circ r_n \sqsubseteq r_{n+1}$  (**transitivity**:  $r \circ r \sqsubseteq r$ )

**OWL 2 EL** profile includes all these extensions

We will **focus on plain  $\mathcal{EL}$**  (without these extensions)



$\mathcal{T}$  is in **normal form** if it contains only inclusions of the forms:

$$A \sqsubseteq B \quad A_1 \sqcap A_2 \sqsubseteq B \quad A \sqsubseteq \exists r.B \quad \exists r.A \sqsubseteq B$$

where  $A, A_1, A_2, B$  are concept names (or  $\top$ ).

Use term **basic axioms** for such inclusions, and **basic assertions** to refer to non-complex assertions

$\mathcal{T}$  is in **normal form** if it contains only inclusions of the forms:

$$A \sqsubseteq B \quad A_1 \sqcap A_2 \sqsubseteq B \quad A \sqsubseteq \exists r.B \quad \exists r.A \sqsubseteq B$$

where  $A, A_1, A_2, B$  are concept names (or  $\top$ ).

Use term **basic axioms** for such inclusions, and **basic assertions** to refer to non-complex assertions

**Theorem:** for every  $\mathcal{EL}$  TBox  $\mathcal{T}$ , we can **construct in PTIME** a TBox  $\mathcal{T}'$  **in normal form (possibly using new concept names)** such that:

- for every inclusion  $C \sqsubseteq D$  which uses only concept names from  $\mathcal{T}$ , we have  $\mathcal{T} \models C \sqsubseteq D$  **iff**  $\mathcal{T}' \models C \sqsubseteq D$
- for every ABox  $\mathcal{A}$  and assertion  $\alpha$  that only uses concept names from  $(\mathcal{T}, \mathcal{A})$ , we have  $\mathcal{T}, \mathcal{A} \models \alpha$  **iff**  $\mathcal{T}', \mathcal{A} \models \alpha$

$\mathcal{T}$  is in **normal form** if it contains only inclusions of the forms:

$$A \sqsubseteq B \quad A_1 \sqcap A_2 \sqsubseteq B \quad A \sqsubseteq \exists r.B \quad \exists r.A \sqsubseteq B$$

where  $A, A_1, A_2, B$  are concept names (or  $\top$ ).

Use term **basic axioms** for such inclusions, and **basic assertions** to refer to non-complex assertions

**Theorem:** for every  $\mathcal{EL}$  TBox  $\mathcal{T}$ , we can **construct in PTIME** a TBox  $\mathcal{T}'$  **in normal form (possibly using new concept names)** such that:

- for every inclusion  $C \sqsubseteq D$  which uses only concept names from  $\mathcal{T}$ , we have  $\mathcal{T} \models C \sqsubseteq D$  **iff**  $\mathcal{T}' \models C \sqsubseteq D$
- for every ABox  $\mathcal{A}$  and assertion  $\alpha$  that only uses concept names from  $(\mathcal{T}, \mathcal{A})$ , we have  $\mathcal{T}, \mathcal{A} \models \alpha$  **iff**  $\mathcal{T}', \mathcal{A} \models \alpha$

More specifically:  $\mathcal{T}'$  is a **conservative extension** of  $\mathcal{T}$  (see homework)

Exhaustively apply the following **normalization rules** to  $\mathcal{T}$ : (any order)

$$\begin{array}{lll}
 \hat{D} \sqsubseteq \hat{E} & \rightsquigarrow & \hat{D} \sqsubseteq A_{\text{new}} \quad A_{\text{new}} \sqsubseteq \hat{E} \\
 C \sqcap \hat{D} \sqsubseteq B & \rightsquigarrow & \hat{D} \sqsubseteq A_{\text{new}} \quad C \sqcap A_{\text{new}} \sqsubseteq B \\
 \hat{D} \sqcap C \sqsubseteq B & \rightsquigarrow & \hat{D} \sqsubseteq A_{\text{new}} \quad A_{\text{new}} \sqcap C \sqsubseteq B \\
 \exists r. \hat{D} \sqsubseteq B & \rightsquigarrow & \hat{D} \sqsubseteq A_{\text{new}} \quad \exists r. A_{\text{new}} \sqsubseteq B \\
 B \sqsubseteq \exists r. \hat{D} & \rightsquigarrow & B \sqsubseteq \exists r. A_{\text{new}} \quad A_{\text{new}} \sqsubseteq \hat{D} \\
 B \sqsubseteq D \sqcap E & \rightsquigarrow & B \sqsubseteq D \quad B \sqsubseteq E
 \end{array}$$

where:

- $C, D, E$  are arbitrary  $\mathcal{EL}$  concepts,
- $\hat{D}, \hat{E}$  are neither concept names nor  $\top$ ,
- $B$  is a concept name,
- $A_{\text{new}}$  is a **fresh (new) concept name**

## EXAMPLE: NORMALIZATION

Applying the fourth rule to  $\exists r.(\exists s.A \sqcap H) \sqsubseteq B \sqcap D$

$$\exists s.A \sqcap H \sqsubseteq E \quad \exists r.E \sqsubseteq B \sqcap D$$

Use third rule to transform  $\exists s.A \sqcap H \sqsubseteq E$  into

$$\exists s.A \sqsubseteq F \quad F \sqcap H \sqsubseteq E$$

Last rule used to replace  $\exists r.E \sqsubseteq B \sqcap D$  by

$$\exists r.E \sqsubseteq B \quad \exists r.E \sqsubseteq D$$

End result:

$$\exists s.A \sqsubseteq F \quad F \sqcap H \sqsubseteq E \quad \exists r.E \sqsubseteq B \quad \exists r.E \sqsubseteq D$$

## Rules for deriving ontology axioms

$$\frac{}{A \sqsubseteq A} \text{ T1}$$

$$\frac{}{A \sqsubseteq \top} \text{ T2}$$

$$\frac{A \sqsubseteq B \quad B \sqsubseteq D}{A \sqsubseteq D} \text{ T3}$$

$$\frac{A \sqsubseteq B_1 \quad A \sqsubseteq B_2 \quad B_1 \sqcap B_2 \sqsubseteq D}{A \sqsubseteq D} \text{ T4}$$

$$\frac{A \sqsubseteq \exists r.B_1 \quad B_1 \sqsubseteq B_2 \quad \exists r.B_2 \sqsubseteq D}{A \sqsubseteq D} \text{ T5}$$

## Rules for deriving assertions

$$\frac{A \sqsubseteq B \quad A(c)}{B(c)} \text{ A1}$$

$$\frac{A_1 \sqcap A_2 \sqsubseteq B \quad A_1(c) \quad A_2(c)}{B(c)} \text{ A2}$$

$$\frac{\exists r.A \sqsubseteq B \quad r(c, d) \quad A(d)}{B(c)} \text{ A3}$$

**Premises** = axioms / assertions above the line

**Conclusion** = axiom / assertion below the line

Assume w.l.o.g. that start from KB whose **TBox is in normal form** & whose **ABox contains  $\top(a)$  for each of its individuals  $a$**

Assume w.l.o.g. that start from KB whose **TBox is in normal form** & whose **ABox contains  $\top(a)$  for each of its individuals  $a$**

### Instantiated rule:

- obtained from one of the 'abstract' saturation rules by replacing  $A, B, D$  by  $\mathcal{EL}$ -concepts and  $r$  by some role name
- must **only contain basic axioms & assertions** (important!)



Assume w.l.o.g. that start from KB whose **TBox is in normal form** & whose **ABox contains  $\top(a)$  for each of its individuals  $a$**

**Instantiated rule:**

- obtained from one of the ‘abstract’ saturation rules by replacing  $A, B, D$  by  $\mathcal{EL}$ -concepts and  $r$  by some role name
- must **only contain basic axioms & assertions** (important!)

**Instantiated rule** with premises  $\alpha_1, \dots, \alpha_n$  and conclusion  $\beta$  is **applicable in  $\mathcal{K}$**  if  $\{\alpha_1, \dots, \alpha_n\} \subseteq \mathcal{K}$  and  $\beta \notin \mathcal{K}$

- in this case, can **apply the rule by adding  $\beta$  to  $\mathcal{K}$**

Saturation procedure: **exhaustively apply instantiated rules** until no rule is applicable

## EXAMPLE: SATURATION RULES

TBox  $\mathcal{T}$  contains axioms:

- |   |  |
|---|--|
| (1) $\exists \text{hasIngred}.\text{Spicy} \sqsubseteq \text{Spicy}$      | (2) $\text{Spicy} \sqcap \text{Dish} \sqsubseteq \text{SpicyDish}$ |
| (3) $\text{ArrabSauce} \sqsubseteq \exists \text{hasIngred}.\text{Chili}$ | (4) $\text{Chili} \sqsubseteq \text{Spicy}$                        |

ABox  $\mathcal{A}$  contains:

- |                      |                              |                            |
|----------------------|------------------------------|----------------------------|
| (5) $\text{Dish}(p)$ | (6) $\text{hasIngred}(p, s)$ | (7) $\text{ArrabSauce}(s)$ |
|----------------------|------------------------------|----------------------------|

## EXAMPLE: SATURATION RULES

TBox  $\mathcal{T}$  contains axioms:

- |   |  |
|---|--|
| (1) $\exists \text{hasIngred}.\text{Spicy} \sqsubseteq \text{Spicy}$      | (2) $\text{Spicy} \sqcap \text{Dish} \sqsubseteq \text{SpicyDish}$ |
| (3) $\text{ArrabSauce} \sqsubseteq \exists \text{hasIngred}.\text{Chili}$ | (4) $\text{Chili} \sqsubseteq \text{Spicy}$                        |

ABox  $\mathcal{A}$  contains:

- (5)  $\text{Dish}(p)$       (6)  $\text{hasIngred}(p, s)$       (7)  $\text{ArrabSauce}(s)$

Saturation procedure adds the following axioms and assertions:

- |  |                                   |
|--|-----------------------------------|
| (8) $\text{ArrabSauce} \sqsubseteq \text{Spicy}$ | using (1), (3), (4) and rule T5   |
| (9) $\text{Spicy}(s)$                            | using (7), (8), and rule A1       |
| (10) $\text{Spicy}(p)$                           | using (1), (6), (9), and rule A3  |
| (11) $\text{SpicyDish}(p)$                       | using (2), (5), (10), and rule A2 |

## EXAMPLE: SATURATION RULES

TBox  $\mathcal{T}$  contains axioms:

- |   |  |
|---|--|
| (1) $\exists \text{hasIngred}.\text{Spicy} \sqsubseteq \text{Spicy}$      | (2) $\text{Spicy} \sqcap \text{Dish} \sqsubseteq \text{SpicyDish}$ |
| (3) $\text{ArrabSauce} \sqsubseteq \exists \text{hasIngred}.\text{Chili}$ | (4) $\text{Chili} \sqsubseteq \text{Spicy}$                        |

ABox  $\mathcal{A}$  contains:

- (5)  $\text{Dish}(p)$       (6)  $\text{hasIngred}(p, s)$       (7)  $\text{ArrabSauce}(s)$

Saturation procedure adds the following axioms and assertions:

- |  |                                   |
|--|-----------------------------------|
| (8) $\text{ArrabSauce} \sqsubseteq \text{Spicy}$ | using (1), (3), (4) and rule T5   |
| (9) $\text{Spicy}(s)$                            | using (7), (8), and rule A1       |
| (10) $\text{Spicy}(p)$                           | using (1), (6), (9), and rule A3  |
| (11) $\text{SpicyDish}(p)$                       | using (2), (5), (10), and rule A2 |

Examining the result, **return  $p$  as answer to instance query**  
 **$q(x) = \text{SpicyDish}(x)$**

Denote by  $\text{sat}(\mathcal{K})$  or  $\text{sat}(\mathcal{T}, \mathcal{A})$  (resp.  $\text{sat}(\mathcal{T})$ ) result of exhaustively applying saturation rules to KB  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  (resp. TBox  $\mathcal{T}$ )

Denote by  $\text{sat}(\mathcal{K})$  or  $\text{sat}(\mathcal{T}, \mathcal{A})$  (resp.  $\text{sat}(\mathcal{T})$ ) result of exhaustively applying saturation rules to KB  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  (resp. TBox  $\mathcal{T}$ )

To find all **instances of concept name  $A$**  w.r.t.  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ :

1. Normalize  $\mathcal{T}$ , yielding  $\mathcal{T}'$ , then construct  $\text{sat}(\mathcal{T}', \mathcal{A})$
2. Return **all individuals  $c$  such that  $A(c) \in \text{sat}(\mathcal{T}', \mathcal{A})$ .**

Denote by  $\text{sat}(\mathcal{K})$  or  $\text{sat}(\mathcal{T}, \mathcal{A})$  (resp.  $\text{sat}(\mathcal{T})$ ) result of exhaustively applying saturation rules to KB  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  (resp. TBox  $\mathcal{T}$ )

To find all **instances of concept name  $A$**  w.r.t.  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ :

1. Normalize  $\mathcal{T}$ , yielding  $\mathcal{T}'$ , then construct  $\text{sat}(\mathcal{T}', \mathcal{A})$
2. Return **all individuals  $c$  such that  $A(c) \in \text{sat}(\mathcal{T}', \mathcal{A})$** .

To **test whether  $\mathcal{T} \models A \sqsubseteq B$**  ( $A, B$  concept names):

1. Normalize  $\mathcal{T}$ , yielding  $\mathcal{T}'$ , then construct  $\text{sat}(\mathcal{T}')$   
(can alternatively construct  $\text{sat}(\mathcal{T}', \mathcal{A})$  if have an ABox  $\mathcal{A}$ )
2. Check if  **$\text{sat}(\mathcal{T}')$  contains  $A \sqsubseteq B$** , return yes if so, else no.

Denote by  $\text{sat}(\mathcal{K})$  or  $\text{sat}(\mathcal{T}, \mathcal{A})$  (resp.  $\text{sat}(\mathcal{T})$ ) result of exhaustively applying saturation rules to KB  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  (resp. TBox  $\mathcal{T}$ )

To find all **instances of concept name  $A$**  w.r.t.  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ :

1. Normalize  $\mathcal{T}$ , yielding  $\mathcal{T}'$ , then construct  $\text{sat}(\mathcal{T}', \mathcal{A})$
2. Return **all individuals  $c$  such that  $A(c) \in \text{sat}(\mathcal{T}', \mathcal{A})$** .

To **test whether  $\mathcal{T} \models A \sqsubseteq B$**  ( $A, B$  concept names):

1. Normalize  $\mathcal{T}$ , yielding  $\mathcal{T}'$ , then construct  $\text{sat}(\mathcal{T}')$   
(can alternatively construct  $\text{sat}(\mathcal{T}', \mathcal{A})$  if have an ABox  $\mathcal{A}$ )
2. Check if  **$\text{sat}(\mathcal{T}')$  contains  $A \sqsubseteq B$** , return yes if so, else no.

What about **assertions / inclusions involving complex concepts**?

- **Use new concept names to represent complex concepts**, e.g. if  $C$  is a complex concept, add  $X_C \sqsubseteq C$  and  $C \sqsubseteq X_C$  to  $\mathcal{T}$  (with  $X_C$  fresh).
- Proceed as above but **use  $X_C$  in place of  $C$** .



**Theorem.** All exhaustive sequences of rule applications lead to a **unique saturated KB**.

**Theorem.** The saturated KB  $\text{sat}(\mathcal{K})$  can be **constructed in polynomial time** in  $|\mathcal{K}|$

**Theorem.** All exhaustive sequences of rule applications lead to a **unique saturated KB**.

**Theorem.** The saturated KB  $\text{sat}(\mathcal{K})$  can be **constructed in polynomial time** in  $|\mathcal{K}|$

**Theorem.** The saturation procedure is **correct and complete for axiom entailment and instance checking** involving concept names. Specifically:

- for every concept inclusion  $A \sqsubseteq B$  (with  $A, B$  concept names):  
 $\mathcal{T} \models A \sqsubseteq B$  iff  $A \sqsubseteq B \in \text{sat}(\mathcal{K})$  iff  $A \sqsubseteq B \in \text{sat}(\mathcal{T})$

**Theorem.** All exhaustive sequences of rule applications lead to a **unique saturated KB**.

**Theorem.** The saturated KB  $\text{sat}(\mathcal{K})$  can be **constructed in polynomial time** in  $|\mathcal{K}|$

**Theorem.** The saturation procedure is **correct and complete for axiom entailment and instance checking** involving concept names. Specifically:

- for every concept inclusion  $A \sqsubseteq B$  (with  $A, B$  concept names):  
 $\mathcal{T} \models A \sqsubseteq B$  iff  $A \sqsubseteq B \in \text{sat}(\mathcal{K})$  iff  $A \sqsubseteq B \in \text{sat}(\mathcal{T})$
- for every ABox assertion  $\alpha = A(b)$  with  $A$  a concept name:  
 $\mathcal{K} \models A(b)$  iff  $A(b) \in \text{sat}(\mathcal{K})$

Next slides: sketch proofs for correctness and completeness

Aim to show that:

- if  $A \sqsubseteq B \in \text{sat}(\mathcal{K})$ , then  $\mathcal{T} \models A \sqsubseteq B$
- if  $A(b) \in \text{sat}(\mathcal{K})$ , then  $\mathcal{K} \models A(b)$

Aim to show that:

- if  $A \sqsubseteq B \in \text{sat}(\mathcal{K})$ , then  $\mathcal{T} \models A \sqsubseteq B$
- if  $A(b) \in \text{sat}(\mathcal{K})$ , then  $\mathcal{K} \models A(b)$

As  $\text{sat}(\mathcal{K})$  is the result of a sequence of rule applications, it suffices to show the following lemma:

**Lemma.** If a saturation rule application produces  $\beta$  from the premises  $\alpha_1, \dots, \alpha_n$ , and  $\mathcal{K} \models \alpha_i$  ( $1 \leq i \leq n$ ), then  $\mathcal{K} \models \beta$ .

Aim to show that:

- if  $A \sqsubseteq B \in \text{sat}(\mathcal{K})$ , then  $\mathcal{T} \models A \sqsubseteq B$
- if  $A(b) \in \text{sat}(\mathcal{K})$ , then  $\mathcal{K} \models A(b)$

As  $\text{sat}(\mathcal{K})$  is the result of a sequence of rule applications, it suffices to show the following lemma:

**Lemma.** If a saturation rule application produces  $\beta$  from the premises  $\alpha_1, \dots, \alpha_n$ , and  $\mathcal{K} \models \alpha_i$  ( $1 \leq i \leq n$ ), then  $\mathcal{K} \models \beta$ .

Proof sketch:

- trivial for rules T1 and T2 (produced axioms hold in any model)
- easy arguments for other rules, e.g. for T3:
  - suppose  $\mathcal{K} \models A \sqsubseteq B$  and  $\mathcal{K} \models B \sqsubseteq D$ , take any model  $\mathcal{I}$  of  $\mathcal{K}$  and  $e \in A^{\mathcal{I}}$ , must have  $e \in B^{\mathcal{I}}$  due to  $A \sqsubseteq B$ , hence  $e \in D^{\mathcal{I}}$  due to  $B \sqsubseteq D$ , yielding  $\mathcal{I} \models A \sqsubseteq D$  as required

We prove the contrapositive, namely:

- if  $A \sqsubseteq B \notin \text{sat}(\mathcal{K})$ , then  $\mathcal{T} \not\models A \sqsubseteq B$
- if  $A(b) \notin \text{sat}(\mathcal{K})$ , then  $\mathcal{K} \not\models A(b)$

We prove the contrapositive, namely:

- if  $A \sqsubseteq B \notin \text{sat}(\mathcal{K})$ , then  $\mathcal{T} \not\models A \sqsubseteq B$
- if  $A(b) \notin \text{sat}(\mathcal{K})$ , then  $\mathcal{K} \not\models A(b)$

Proof strategy:

- build an interpretation  $\mathcal{C}_{\mathcal{K}}$  from  $\text{sat}(\mathcal{K})$
- show that  $\mathcal{C}_{\mathcal{K}}$  is a model of  $\mathcal{K}$
- show that  $\mathcal{C}_{\mathcal{K}} \not\models A \sqsubseteq B$  when  $A \sqsubseteq B \notin \text{sat}(\mathcal{K})$
- show that  $\mathcal{C}_{\mathcal{K}} \not\models A(b)$  when  $A(b) \notin \text{sat}(\mathcal{K})$



Define  $\mathcal{C}_{\mathcal{K}}$ , as follows:

- $\Delta^{\mathcal{C}_{\mathcal{K}}} = \text{Ind}(\mathcal{A}) \cup \{w_A \mid A \text{ concept name appearing in } \mathcal{K}\} \cup \{w_{\top}\}$
- $A^{\mathcal{C}_{\mathcal{K}}} = \{b \mid A(b) \in \text{sat}(\mathcal{K})\} \cup \{w_B \mid B \sqsubseteq A \in \text{sat}(\mathcal{K})\}$
- $r^{\mathcal{C}_{\mathcal{K}}} = \{(a, b) \mid r(a, b) \in \mathcal{K}\} \cup \{(w_A, w_B) \mid A \sqsubseteq \exists r.B \in \text{sat}(\mathcal{K})\} \\ \cup \{(a, w_B) \mid A \sqsubseteq \exists r.B \in \text{sat}(\mathcal{K}), A(a) \in \text{sat}(\mathcal{K}) \text{ for some } A\}$
- $a^{\mathcal{C}_{\mathcal{K}}} = a$  for all  $a \in \text{Ind}(\mathcal{A})$

where  $\text{Ind}(\mathcal{A})$  is set of individual names in  $\mathcal{A}$

Define  $\mathcal{C}_{\mathcal{K}}$ , as follows:

- $\Delta^{\mathcal{C}_{\mathcal{K}}} = \text{Ind}(\mathcal{A}) \cup \{w_A \mid A \text{ concept name appearing in } \mathcal{K}\} \cup \{w_{\top}\}$
- $A^{\mathcal{C}_{\mathcal{K}}} = \{b \mid A(b) \in \text{sat}(\mathcal{K})\} \cup \{w_B \mid B \sqsubseteq A \in \text{sat}(\mathcal{K})\}$
- $r^{\mathcal{C}_{\mathcal{K}}} = \{(a, b) \mid r(a, b) \in \mathcal{K}\} \cup \{(w_A, w_B) \mid A \sqsubseteq \exists r.B \in \text{sat}(\mathcal{K})\} \\ \cup \{(a, w_B) \mid A \sqsubseteq \exists r.B \in \text{sat}(\mathcal{K}), A(a) \in \text{sat}(\mathcal{K}) \text{ for some } A\}$
- $a^{\mathcal{C}_{\mathcal{K}}} = a$  for all  $a \in \text{Ind}(\mathcal{A})$

where  $\text{Ind}(\mathcal{A})$  is set of individual names in  $\mathcal{A}$

Observe that by construction, we have:

- $\mathcal{C}_{\mathcal{K}} \not\models A \sqsubseteq B$  when  $A \sqsubseteq B \notin \text{sat}(\mathcal{K})$   $w_A \in A^{\mathcal{C}_{\mathcal{K}}}$  but  $w_A \notin B^{\mathcal{C}_{\mathcal{K}}}$
- $\mathcal{C}_{\mathcal{K}} \not\models A(b)$  when  $A(b) \notin \text{sat}(\mathcal{K})$

for all concept names  $A, B$  and individuals  $b$  occurring in  $\mathcal{K}$

By definition,  $\mathcal{C}_K$  is a model of  $\mathcal{A}$

By definition,  $\mathcal{C}_{\mathcal{K}}$  is a model of  $\mathcal{A}$

To show it is a model of  $\mathcal{T}$ , consider different kinds of axioms:

- **Case 1:**  $A \sqsubseteq B \in \mathcal{T}$  and  $e \in A^{\mathcal{C}_{\mathcal{K}}}$ 
  - If  $e \in \text{Ind}(\mathcal{A})$ , then  $A(e) \in \text{sat}(\mathcal{K})$ . Due to **A1**,  $B(e) \in \text{sat}(\mathcal{K})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .
  - If  $e = w_D$ , then  $D \sqsubseteq A \in \text{sat}(\mathcal{T})$ . Due to **T3**,  $D \sqsubseteq B \in \text{sat}(\mathcal{T})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .

By definition,  $\mathcal{C}_{\mathcal{K}}$  is a model of  $\mathcal{A}$

To show it is a model of  $\mathcal{T}$ , consider different kinds of axioms:

- **Case 1:**  $A \sqsubseteq B \in \mathcal{T}$  and  $e \in A^{\mathcal{C}_{\mathcal{K}}}$ 
  - If  $e \in \text{Ind}(\mathcal{A})$ , then  $A(e) \in \text{sat}(\mathcal{K})$ . Due to **A1**,  $B(e) \in \text{sat}(\mathcal{K})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .
  - If  $e = w_D$ , then  $D \sqsubseteq A \in \text{sat}(\mathcal{T})$ . Due to **T3**,  $D \sqsubseteq B \in \text{sat}(\mathcal{T})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .
- **Case 2:**  $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$  and  $e \in (A_1 \sqcap A_2)^{\mathcal{C}_{\mathcal{K}}}$ 
  - similar argument using **A2** and **T4**
- **Case 3:**  $A \sqsubseteq \exists r.B \in \mathcal{T}$  and  $e \in A^{\mathcal{C}_{\mathcal{K}}}$ 
  - argument uses **T3**
- **Case 4:**  $\exists r.A \sqsubseteq B$ ,  $e' \in A^{\mathcal{C}_{\mathcal{K}}}$ , and  $(e, e') \in r^{\mathcal{C}_{\mathcal{K}}}$ 
  - argument uses **A3** and **T5**

By definition,  $\mathcal{C}_{\mathcal{K}}$  is a model of  $\mathcal{A}$

To show it is a model of  $\mathcal{T}$ , consider different kinds of axioms:

- **Case 1:**  $A \sqsubseteq B \in \mathcal{T}$  and  $e \in A^{\mathcal{C}_{\mathcal{K}}}$ 
  - If  $e \in \text{Ind}(\mathcal{A})$ , then  $A(e) \in \text{sat}(\mathcal{K})$ . Due to **A1**,  $B(e) \in \text{sat}(\mathcal{K})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .
  - If  $e = w_D$ , then  $D \sqsubseteq A \in \text{sat}(\mathcal{T})$ . Due to **T3**,  $D \sqsubseteq B \in \text{sat}(\mathcal{T})$ , so  $e \in B^{\mathcal{C}_{\mathcal{K}}}$ .
- **Case 2:**  $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$  and  $e \in (A_1 \sqcap A_2)^{\mathcal{C}_{\mathcal{K}}}$ 
  - similar argument using **A2** and **T4**
- **Case 3:**  $A \sqsubseteq \exists r.B \in \mathcal{T}$  and  $e \in A^{\mathcal{C}_{\mathcal{K}}}$ 
  - argument uses **T3**
- **Case 4:**  $\exists r.A \sqsubseteq B$ ,  $e' \in A^{\mathcal{C}_{\mathcal{K}}}$ , and  $(e, e') \in r^{\mathcal{C}_{\mathcal{K}}}$ 
  - argument uses **A3** and **T5**

Call  $\mathcal{C}_{\mathcal{K}}$  the (compact) canonical model for  $\mathcal{K}$

**Theorem.** Axiom entailment and instance checking over  $\mathcal{EL}$  KBs are PTIME-complete

- upper bound: saturation procedure from previous slides
- lower bound: entailment from propositional Horn theories

**Theorem.** Axiom entailment and instance checking over  $\mathcal{EL}$  KBs are PTIME-complete

- upper bound: saturation procedure from previous slides
- lower bound: entailment from propositional Horn theories

**Note:** with only  $\sqcap$  and  $\forall r.C$ , same problems are EXPTIME-complete!



**Theorem.** Axiom entailment and instance checking over  $\mathcal{EL}$  KBs are **PTIME-complete**

- upper bound: saturation procedure from previous slides
- lower bound: entailment from propositional Horn theories

**Note:** with **only  $\sqcap$  and  $\forall r.C$** , same problems are **EXPTIME-complete**!

Further advantage of saturation approach: **'single-pass' reasoning**

- **compute saturation once**, then read off all entailed assertions and inclusions involving concept names

**Theorem.** Axiom entailment and instance checking over  $\mathcal{EL}$  KBs are **PTIME-complete**

- upper bound: saturation procedure from previous slides
- lower bound: entailment from propositional Horn theories

**Note:** with **only  $\sqcap$  and  $\forall r.C$** , same problems are **EXPTIME-complete**!

Further advantage of saturation approach: **'single-pass' reasoning**

- **compute saturation once**, then read off all entailed assertions and inclusions involving concept names

In practice:

- **huge ontologies like SNOMED can be classified in a few seconds**

We can add all of the following without losing tractability:

- $\perp$
- $\text{dom}(r) \sqsubseteq C, \text{range}(r) \sqsubseteq C$
- $r_1 \circ \dots \circ r_n \sqsubseteq r_{n+1}$  (complex role inclusions)

We can add all of the following without losing tractability:

- $\perp$
- $\text{dom}(r) \sqsubseteq C, \text{range}(r) \sqsubseteq C$
- $r_1 \circ \dots \circ r_n \sqsubseteq r_{n+1}$  (complex role inclusions)

But adding any of the following makes reasoning EXPTIME-hard:

- negation  $\neg$
- disjunction  $\sqcup$
- at-least or at-most restrictions:  $\geq 2r, \leq 1r$
- functional roles ( $\text{funct } r$ )
- inverse roles  $r^-$

The DL  $\mathcal{ELI}$  is obtained by adding inverse roles to  $\mathcal{EL}$

Reasoning in  $\mathcal{ELI}$  is much more difficult (EXPTIME-complete)

However,  $\mathcal{ELI}$  retains some nice properties:

- admits a canonical model, hence no ‘case-based’ reasoning

Can extend saturation procedure to  $\mathcal{ELI}$

- still deterministic
- may be exponential since need to consider sets of concept names
  - deduce  $A \sqcap D \sqsubseteq \exists r.(B \sqcap D)$  from  $A \sqsubseteq \exists r.B$  and  $\exists r^-.D \sqsubseteq E$

In practice:  $\mathcal{ELI}$  and other ‘Horn DLs’ easier to handle than  $\mathcal{ALC}$