

# **Knowledge Representation & Reasoning**

---

Shyamanta M Hazarika  
CSE, SoE  
Tezpur University

# Formal semantics

---

Interpretation  $\mathcal{I} = \langle D, I \rangle$  as in FOL, where

- for every constant  $c$ ,  $I[c] \in D$
- for every atomic concept  $a$ ,  $I[a] \subseteq D$
- for every role  $r$ ,  $I[r] \subseteq D \times D$

We then extend the interpretation to all concepts as subsets of the domain as follows:

- $I[\text{Thing}] = D$
- $I[[\text{ALL } r \ d]] = \{x \in D \mid \text{for any } y, \text{ if } \langle x, y \rangle \in I[r] \text{ then } y \in I[d]\}$
- $I[[\text{EXISTS } n \ r]] = \{x \in D \mid \text{there are at least } n \ y \text{ such that } \langle x, y \rangle \in I[r]\}$
- $I[[\text{FILLS } r \ c]] = \{x \in D \mid \langle x, I[c] \rangle \in I[r]\}$
- $I[[\text{AND } d_1 \ \dots \ d_k]] = I[d_1] \cap \dots \cap I[d_k]$

A sentence of DL will then be true or false as follows:

- $\mathcal{I} \models (d \sqsubseteq e)$  iff  $I[d] \subseteq I[e]$
- $\mathcal{I} \models (d \doteq e)$  iff  $I[d] = I[e]$
- $\mathcal{I} \models (c \rightarrow e)$  iff  $I[c] \in I[e]$

# Entailment and reasoning

---

Entailment in DL is defined as in FOL:

A set of DL sentences  $S$  entails a sentence  $\alpha$  (which we write  $S \models \alpha$ ) iff  
for every  $\mathcal{I}$ , if  $\mathcal{I} \models S$  then  $\mathcal{I} \models \alpha$

A sentence is valid iff it is entailed by the empty set.

Given a KB consisting of DL sentences, there are two basic sorts of reasoning we consider:

1. determining if  $KB \models (c \rightarrow e)$

whether a named individual satisfies a certain description

2. determining if  $KB \models (d \sqsubseteq e)$

whether one description is subsumed by another

the other case,  $KB \models (d \dot{\sqsubseteq} e)$  reduces to

$KB \models (d \sqsubseteq e)$  and  $KB \models (e \sqsubseteq d)$

# Entailment vs. validity

---

In some cases, an entailment will hold because the sentence in question is valid.

- ([AND Doctor Female]  $\models$  Doctor)
- ([FILLS :Child sue]  $\models$  [EXISTS 1 :Child])
- (john  $\rightarrow$  [ALL :Hobby Thing])

But in most other cases, the entailment depends on the sentences in the KB.

For example,

([AND Surgeon Female]  $\models$  Doctor)

is not valid.

But it is entailed by a KB that contains

(Surgeon  $\stackrel{\bullet}{\models}$  [AND Specialist [FILLS :Specialty surgery]])

(Specialist  $\models$  Doctor)

# Computing subsumption

---

We begin with computing subsumption, that is, determining whether or not  $\text{KB} \models (d \sqsubseteq e)$ .

and therefore  
whether  $d \dot{\sqsubseteq} e$

Some simplifications to the KB:

- we can remove the  $(c \rightarrow d)$  assertions from the KB
- we can replace  $(d \sqsubseteq e)$  in KB by  $(d \dot{\sqsubseteq} [\text{AND } e a])$ , where  $a$  is a new atomic concept
- we assume that in the KB for each  $(d \dot{\sqsubseteq} e)$ , the  $d$  is atomic and appears only once on the LHS
- we assume that the definitions in the KB are acyclic  
vs. cyclic  $(d \dot{\sqsubseteq} [\text{AND } e f]), (e \dot{\sqsubseteq} [\text{AND } d g])$

Under these assumptions, it is sufficient to do the following:

- normalization: using the definitions in the KB, put  $d$  and  $e$  into a special normal form,  $d'$  and  $e'$
- structure matching: determine if each part of  $e'$  is matched by a part of  $d'$ .

# Normalization

Repeatedly apply the following operations to the two concepts:

- expand a definition: replace an atomic concept by its KB definition
- flatten an AND concept:

$$[\text{AND } \dots [\text{AND } d e f] \dots] \Rightarrow [\text{AND } \dots d e f \dots]$$

- combine the ALL operations with the same role:

$$[\text{AND } \dots [\text{ALL } r d] \dots [\text{ALL } r e] \dots] \Rightarrow [\text{AND } \dots [\text{ALL } r [\text{AND } d e]] \dots]$$

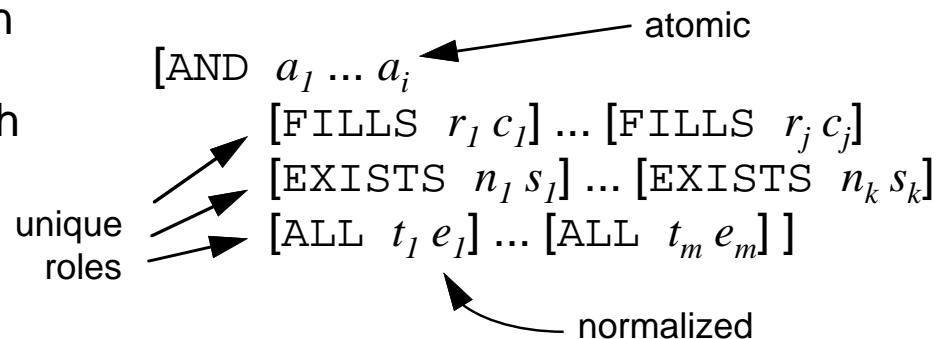
- combine the EXISTS operations with the same role:

$$[\text{AND } \dots [\text{EXISTS } n_1 r] \dots [\text{EXISTS } n_2 r] \dots] \Rightarrow$$

$$[\text{AND } \dots [\text{EXISTS } n r] \dots] \quad (\text{where } n = \text{Max}(n_1, n_2))$$

- remove a vacuous concept: Thing, [ALL r Thing], [AND]
- remove a duplicate expression

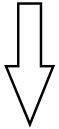
In the end, we end up with a normalized concept of the following form



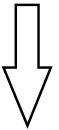
# Normalization example

---

```
[AND Person
  [ALL :Friend Doctor]
  [EXISTS 1 :Accountant]
  [ALL :Accountant [EXISTS 1 :Degree]]
  [ALL :Friend Rich]
  [ALL :Accountant [AND Lawyer [EXISTS 2 :Degree]]]]
```



```
[AND Person
  [EXISTS 1 :Accountant]
  [ALL :Friend [AND Rich Doctor]]
  [ALL :Accountant [AND Lawyer [EXISTS 1 :Degree] [EXISTS 2 :Degree]]]]
```



```
[AND Person
  [EXISTS 1 :Accountant]
  [ALL :Friend [AND Rich Doctor]]
  [ALL :Accountant [AND Lawyer [EXISTS 2 :Degree]]]]
```

# Structure matching

---

Once we have replaced atomic concepts by their definitions, we no longer need to use the KB.

To see if a normalized concept  $[\text{AND } e_1 \dots e_m]$  subsumes a normalized concept  $[\text{AND } d_1 \dots d_n]$ , we do the following:

For each component  $e_j$ , check that there is a matching component  $d_i$ , where

- if  $e_j$  is atomic or  $[\text{FILLS } r \ c]$ , then  $d_i$  must be identical to it;
- if  $e_j = [\text{EXISTS } 1 \ r]$ , then  $d_i$  must be  $[\text{EXISTS } n \ r]$  or  $[\text{FILLS } r \ c]$ ;
- if  $e_j = [\text{EXISTS } n \ r]$  where  $n > 1$ , then  $d_i$  must be of the form  $[\text{EXISTS } m \ r]$  where  $m \geq n$ ;
- if  $e_j = [\text{ALL } r \ e']$ , then  $d_i$  must be  $[\text{ALL } r \ d']$ , where recursively  $e'$  subsumes  $d'$ .

In other words, for every part of the more general concept, there must be a corresponding part in the more specific one.

It can be shown that this procedure is sound and complete:  
it returns YES iff  $\text{KB} \models (d \sqsubseteq e)$ .

# Structure matching example

---

