Knowledge Representation & Reasoning

Shyamanta M Hazarika
CSE, SoE
Tezpur University

Formal semantics

Interpretation $\mathfrak{I} = \langle D, I \rangle$ 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:

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

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

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

Entailment and reasoning

Entailment in DL is defined as in FOL:

A set of DL sentences S <u>entails</u> a sentence α (which we write $S \models \alpha$) iff
for every $\mathfrak{I}$, if $\mathfrak{I} \models S$ then $\mathfrak{I} \models \alpha$

A sentence is <u>valid</u> 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 <u>satisfies</u> a certain description

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

   whether one description is <u>subsumed</u> by another

the other case, KB $\models (d \doteq e)$ 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] ⊑ Doctor)
- ([FILLS :Child sue] ⊑ [EXISTS 1 :Child])
- (john → [ALL :Hobby Thing])

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

For example,

([AND Surgeon Female] ⊑ Doctor)

is not valid.

But it is entailed by a KB that contains

(Surgeon ≐ [AND Specialist [FILLS :Specialty surgery]])

(Specialist ⊑ Doctor)

Computing subsumption

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

and therefore whether $d \doteq e$

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 $(d \doteq$ [AND e a]), where a is a new atomic concept
- we assume that in the KB for each $(d \doteq e)$, the d is atomic and appears only once on the LHS
- we assume that the definitions in the KB are acyclic

  vs. cyclic ($d \doteq$ [AND e f]), ($e \doteq$ [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:

  [AND ... [AND d e f] ...] $\Rightarrow$ [AND ... d e f ...]

- combine the ALL operations with the same role:

  [AND ... [ALL r d] ... [ALL r e] ...] $\Rightarrow$ [AND ... [ALL r [AND d e]] ...]

- combine the EXISTS operations with the same role:

  [AND ... [EXISTS n_1 r] ... [EXISTS n_2 r] ...] $\Rightarrow$
  [AND ... [EXISTS n r] ...] (where n =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 <u>normalized</u> concept of the following form

[AND a_1 ... a_i
[FILLS r_1 c_1] ... [FILLS r_j c_j]
[EXISTS n_1 s_1] ... [EXISTS n_k s_k]
[ALL t_1 e_1] ... [ALL t_m e_m]]

atomic

unique roles

normalized

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 [AND e_1 .... e_m] subsumes a normalized concept [AND d_1 ... 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 [FILLS r c], then d_i must be identical to it;
- if e_j = [EXISTS 1 r], then d_i must be [EXISTS n r] or [FILLS r c];
- if e_j = [EXISTS n r] where n >1, then d_i must be of the form [EXISTS m r] where $m \geq n$;
- if e_j = [ALL r e'], then d_i must be [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 KB $\models (d \sqsubseteq e)$.

Structure matching example

[AND Person
[FILLS :Age 27]
[EXISTS 2 :Child]
[EXISTS 1 :Spouse]
[ALL :Friend [AND Doctor Rich]]]

[AND Person
Female
[FILLS :Age 27]
[EXISTS 3 :Child]
[FILLS :Spouse jack]
[FILLS :Mother sue]
[ALL :Friend [AND Doctor
[FILLS :Specialty surgery]
Rich]]]

not needed