Motivation: Reasoning services

What do we want to know?

- We want to check whether the knowledge base is reasonable:
  - Is each defined concept in a TBox satisfiable?
  - Is a given TBox satisfiable?
  - Is a given ABox satisfiable?
- What can we conclude from the represented knowledge?
  - Is concept $X$ subsumed by concept $Y$?
  - Is an object an instance of a concept $X$?
- These problems can be reduced to logical satisfiability or implication – using the logical semantics.
- However, we take a different route: we will try to simplify these problems and then we specify direct inference methods.
2 Basic Reasoning Services

- Satifiability without a TBox
- Satifiability in TBox

Satisfiability of concept descriptions

Satisfiability of concept descriptions
Given a concept description $C$ in “isolation”, i.e., in an empty TBox, is $C$ satisfiable?

Test:
- Does there exist an interpretation $I$ such that $C^I \neq \emptyset$?
- Translated into FOL: Is the formula $\exists x \ C(x)$ satisfiable?

Example
$\text{Woman} \cap (\leq 0 \ \text{has-child}) \cap (\geq 1 \ \text{has-child})$ is unsatisfiable.

3 Eliminating the TBox

- Normalization
- Unfolding

Example
$\text{Mother-without-daughter} \cap \forall \text{has-child}. \text{Female}$ is unsatisfiable, given our previously specified family TBox.
Reduction: Getting rid of the TBox

We can reduce satisfiability problem of concept descriptions in a TBox to the satisfiability problem of concept descriptions in the empty TBox.

Idea:
- Since TBoxes are cycle-free, one can understand a concept definition as a kind of “macro”.
- For a given TBox $T$ and a given concept description $C$, all defined concept symbols appearing in $C$ can be expanded until $C$ contains only undefined concept symbols.
- An expanded concept description is then satisfiable if and only if $C$ is satisfiable in $T$.
- Problem: What do we do with partial definitions (using $\sqsubseteq$)?

Normalized terminologies

- A terminology is called normalized when it does not contain definitions of the form $A \sqsubseteq C$.
- In order to normalize a terminology, replace $A \sqsubseteq C$ by $A \equiv A^* \sqcap C$, where $A^*$ is a fresh concept symbol (not appearing elsewhere in $T$).
- If $T$ is a terminology, the normalized terminology is denoted by $\tilde{T}$.

Normalizing is reasonable

Theorem (Normalization invariance)

If $I$ is a model of the terminology $T$, then there exists a model $I'$ of $\tilde{T}$ such that for all concept symbols $A$ occurring in $T$, it holds $A^I = A^{'I}$, and vice versa.

Proof.
"$\Rightarrow$": Let $I$ be a model of $T$. This model should be extended to $I'$ so that the freshly introduced concept symbols also get interpretations. Assume $(A \sqsubseteq C) \in T$, i.e., we have $(A \equiv A^* \sqcap C) \in \tilde{T}$. Then set $A^{'I} := A^I$. $I'$ obviously satisfies $\tilde{T}$ and has the same interpretation for all symbols in $\tilde{T}$.

"$\Leftarrow$": Given a model $I'$ of $\tilde{T}$, its restriction to symbols of $T$ is the interpretation we look for.

TBox unfolding

- We say that a normalized TBox is unfolded by one step when all defined concept symbols on the right sides are replaced by their defining terms.
- Example: Mother $\equiv$ Woman $\sqcap \ldots$ is unfolded to Mother $\equiv$ (Human $\sqcap$ Female) $\sqcap \ldots$.
- We write $U(T)$ to denote a one-step unfolding and $U^n(T)$ to denote an $n$-step unfolding.
- We say that $T$ is unfolded if $U(T) = T$.
- $U^n(T)$ is called the unfolding of $T$ if $U^n(T) = U^{n+1}(T)$.
- If such an unfolding exists, it is denoted by $\hat{T}$. 
Properties of unfoldings (1): Existence

Theorem (Existence of unfolded terminology)
Each normalized terminology \( T \) can be unfolded, i.e., its unfolding \( \hat{T} \) exists.

Proof idea.
The main reason is that terminologies have to be cycle-free. The proof can be done by induction of the definition depth of concepts.

Properties of unfoldings (2): Equivalence

Theorem (Model equivalence for unfolded terminologies)
\( I \) is a model of a normalized terminology \( T \) if and only if it is a model of \( \hat{T} \).

Proof sketch.
\( \Rightarrow \): Let \( I \) be a model of \( T \). Then it is also a model of \( U(T) \), since on the right side of the definitions only terms with identical interpretations are substituted. However, then it must also be a model of \( \hat{T} \).
\( \Leftarrow \): Let \( I \) be a model for \( U(T) \). Clearly, this is also a model of \( T \) (with the same argument as above). This means that any model \( \hat{T} \) is also a model of \( T \).

Generating models

- All concept and role names not occurring on the left hand side of definitions in a terminology \( T \) are called primitive components.
- Interpretations restricted to primitive components are called initial interpretations.

Theorem (Model extension)
For each initial interpretation \( J \) of a normalized TBox, there exists a unique interpretation \( I \) extending \( J \) and satisfying \( T \).

Proof idea.
Use \( \hat{T} \) and compute an interpretation for all defined symbols.

Corollary (Model existence for TBoxes)
Each TBox has at least one model.

Unfolding of concept descriptions

- Similar to the unfolding of TBoxes, we can define the unfolding of a concept description.
- We write \( \hat{C} \) for the unfolded version of \( C \).

Theorem (Satisfiability of unfolded concepts)
An concept description \( C \) is satisfiable in a terminology \( T \) if and only if \( \hat{C} \) is satisfiable in an empty terminology.

Proof:
\( \Rightarrow \): trivial.
\( \Leftarrow \): Use the interpretation for all the symbols in \( \hat{C} \) to generate an initial interpretation of \( T \). Then extend it to a full model \( I \) of \( T \). This satisfies \( T \) as well as \( \hat{C} \). Since \( \hat{C} = C \), it satisfies also \( C \).
4 General TBox Reasoning Services

- Subsumption
- Subsumption vs. Satisfiability
- Classification

February 3, 2014 Nebel, Wölfl, Hué – KRR 21 / 37

Subsumption in a TBox

Given a terminology \( T \) and two concept descriptions \( C \) and \( D \), is \( C \) subsumed by (or a sub-concept of) \( D \) in \( T \) (symb. \( C \sqsubseteq_T D \))?

Test:
- Is \( C \) interpreted as a subset of \( D \) in each model \( I \) of \( T \), i.e.
  \[ C^I \subseteq D^I \]?
- Is the formula \( \forall x (C(x) \rightarrow D(x)) \) a logical consequence of the translation of \( T \) into FOL?

Example
Given our family TBox, it holds \( \text{Grandmother} \sqsubseteq_T \text{Mother} \).

February 3, 2014 Nebel, Wölfl, Hué – KRR 22 / 37

Subsumption (without a TBox)

Given two concept descriptions \( C \) and \( D \), is \( C \) subsumed by \( D \) regardless of a TBox (or in an empty TBox) (symb. \( C \sqsubseteq D \))?

Test:
- Is \( C \) interpreted as a subset of \( D \) for all interpretations \( I \)
  \( C^I \subseteq D^I \)?
- Is the formula \( \forall x (C(x) \rightarrow D(x)) \) logically valid?

Example
Clearly, \( \text{Human} \sqcap \text{Female} \sqsubseteq \text{Human} \).

February 3, 2014 Nebel, Wölfl, Hué – KRR 23 / 37

Reductions

Subsumption in a TBox can be reduced to subsumption in the empty TBox:
- ... normalize and unfold TBox and concept descriptions.

Subsumption in the empty TBox can be reduced to unsatisfiability:
- ... \( C \sqsubseteq D \) iff \( C \sqcap \neg D \) is unsatisfiable.

Unsatisfiability can be reduced to subsumption:
- ... \( C \) is unsatisfiable iff \( C \sqsubseteq (C \sqcap \neg C) \).

February 3, 2014 Nebel, Wölfl, Hué – KRR 24 / 37
Classification

Compute all subsumption relationships (and represent them using only a minimal number of relationships)!

Useful in order to:
- check the modeling
- use the precomputed relations later when subsumption queries have to be answered

Problem can be reduced to subsumption checking: then it is a generalized sorting problem!

Example

<table>
<thead>
<tr>
<th>Female</th>
<th>Human</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woman</td>
<td>Man</td>
<td></td>
</tr>
<tr>
<td>Parent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mother-w-o-d</td>
<td>Grandmother</td>
<td></td>
</tr>
</tbody>
</table>

Living_Entity

ABox satisfiability

Satisfiability of an ABox
Given an ABox $\mathcal{A}$, does this set of assertions have a model?

- **Notice**: ABoxes representing the real world, should always have a model.

Example

The ABox

$$X : (\forall r . \neg C), \hspace{1cm} Y : C, \hspace{1cm} (X, Y) : r$$

is not satisfiable.

ABox satisfiability in a TBox

ABox satisfiability in a TBox
Given an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, is $\mathcal{A}$ consistent with the terminology introduced in $\mathcal{T}$, i.e., is $\mathcal{T} \cup \mathcal{A}$ satisfiable?

Example

If we extend our example with

$$\text{MARGRET}: \text{Woman}$$

$$(\text{DIANA}, \text{MARGRET}): \text{has-child},$$

then the ABox becomes unsatisfiable in the given TBox.

- Problem is reducible to satisfiability of an ABox:
  - ... normalize terminology, then unfold all concept and role descriptions in the ABox
Instance relations

Which additional ABox formulae of the form $a : C$ follow logically from a given ABox and TBox?

- Is $a^T \in C^T$ true in all models $I$ of $T \cup A$?
- Does the formula $C(a)$ logically follow from the translation of $A$ and $T$ to predicate logic?

Reductions:
- Instance relations wrt. an ABox and a TBox can be reduced to instance relations wrt. ABox: use normalization and unfolding.
- Instance relations in an ABox can be reduced to ABox unsatisfiability:

  $a : C$ holds in $A \iff A \cup \{a : \neg C\}$ is unsatisfiable

Realization

For a given object $a$, determine the most specialized concept symbols such that $a$ is an instance of these concepts.

**Motivation:**
- Similar to classification.
- Is the minimal representation of the instance relations (in the set of concept symbols).
- Will give us faster answers for instance queries!

**Reduction:** Can be reduced to (a sequence of) instance relation tests.

Examples

**Example**

- **ELIZABETH:** Mother-with-many-children?
  - yes
- **WILLIAM:** Female?
  - yes
- **ELIZABETH:** Mother-without-daughter?
  - no (no CWA!),
- **ELIZABETH:** Grandmother?
  - no (only male, but not necessarily human!)

Retrieval

Given a concept description $C$, determine the set of all (specified) instances of the concept description.

**Example**

We ask for all instances of the concept Male.
For our TBOX/ABox we will get the answer CHARLES, ANDREW, EDWARD, WILLIAM.

- **Reduction:** Compute the set of instances by testing the instance relation for each object!
- **Implementation:** Realization can be used to speed this up!
6 Summary and Outlook

Reasoning services – summary

- Satisfiability of concept descriptions
  - in a given TBox or in an empty TBox
- Subsumption between concept descriptions
  - in a given TBox or in an empty TBox
- Classification
- Satisfiability of an ABox
  - in a given TBox or in an empty TBox
- Instance relations in an ABox
  - in a given TBox or in an empty TBox
- Realization
- Retrieval

Outlook

- How to determine subsumption between two concept descriptions (in the empty TBox)?
- How to determine instance relations/ABox satisfiability?
- How to implement the mentioned reductions efficiently?
- Does normalization and unfolding introduce another source of computational complexity?