UNIVERSITY OF TRENTO - Italy
Dipartimento di Ingegneria e Scienza dell'Informazione

Know dive

Logics

Intensional representation

Sept 13,2023

Representation language

Definition (Language, intensional representation) Let L = La ∪ Lc be a language. Then its **intensional representation** is

$$\mathrm{L}i = < \mathrm{L}ia, \mathrm{L}ic >$$

where Lia is the **language of atomic formulas, intensionally defined** and Lic is the **language of complex formulas, intensionally defined**.

Language of atomic formulas

Definition (Language of atomic formulas, intensional representation) Let Li =< Lia, Lic > be a language intensionally defined. Then the **intensional representation**of Lia is

$$\text{L}ia =< \text{A}a, \{\text{FR}\}a >$$

where: Aa is the **alphabet** of La and {FR}a is the set of **formation rules for** Lea with

$$\text{A}a =< \text{E}, \{\text{C}\}, \{\text{R}\} >$$

$$\text{L}ea = \{w : w \in \text{C}(\{\text{FR}\}a, \text{A}a)\}$$

where: E is a set of **(names of) entities**, {C} is a set of **concepts**, where a conceptis a **name of a class**, {P} and a set of **properties**, where a property is a **name of a relation** and, finally, C({FR}a,Aa) is the transitive closure of {FR}a on Aa.

Example - Language of atomic formulas

Consider the language which allows for atomic complex formulas of shape C1 ⊓ C2 where Ci is a concept. The BNF generating this language consists of the following two formation rules:

<awff> ::= <concept>

<awff> ::= <awff> ⊓ <awff>

where <concept> is non-terminal symbol which stands for any element C of the alphabet. ⊓ is a terminal symbol which, as such, cannot be further decomposed.

Formation rules

Definition 9.3 (Formation rule) We restrict ourselves to languages with context-free grammars. Accordingly, we take {FR}a = {Ra}, where each formation rule Ra has form

<expression> ::= --expression--

where, following BNF notation1:

- <expression> is a *nonterminal* expression. Nonterminals are enclosed within <>;
- Symbols that do not appear on the left side of a rule are called *terminals*;
- -expression-- consists of one or more sequences of either terminal or nonterminal symbols;
- ::= allows for <expression> to be replaced with a sequence occurring in --expression--;
- Sequences in --expression-- are separated by the bar ”—”, indicating choice in the substitution.

Transitive closure

Observation (Transitive closure) C({FR}c, La) is the minimal set of formulas which can be obtained by recursively applying the rules of {FR}c to their own results, starting from La.

Atomic formulas are black boxes for {FR}c in the sense that the rules in {FR}c can compose them into complex formulas but cannot change their internal structure.

Interpretation function

Definition (Interpretation function, intensional representation) Let

$$\mathrm{L} = \mathrm{L}a \cup \mathrm{L}c$$

be a language with

$$\mathrm{L}a = \mathrm{L}A \cup \mathrm{L}AC.$$

Let the interpretation function I : La → D be defined as

$$\mathrm{I} = \mathrm{I}A \circ \mathrm{I}AC,$$

with IAC : La → LA and IA : LA → D. Then, the **intensional representation** of I is

$$\mathrm{I}i =< \mathrm{L}a, \{\mathrm{FR}\}\mathrm{I} >$$

where {FR}I is the set of **formation rules for** Ie with

$$\mathrm{I}e = \{< w, \mathrm{f} >: w \in \mathrm{L}a, \mathrm{f} \in \mathrm{D}, < w, \mathrm{f} >\in \mathrm{C}(\{\mathrm{FR}\}\mathrm{I}, \mathrm{L}a)\}$$

where C({FR}I, La) is the transitive closure of {FR}I over La.

Interpretation function

Definition (Interpretation function, intensional representation) Let

$$\mathrm{L} = \mathrm{L}a \cup \mathrm{L}c$$

be a language with

$$\mathrm{L}a = \mathrm{L}A \cup \mathrm{L}AC.$$

Let the interpretation function $\mathrm{I} : \mathrm{L}a \to \mathrm{D}$ be defined as

$$\mathrm{I} = \mathrm{I}A \circ \mathrm{I}AC,$$

with $\mathrm{I}AC : \mathrm{L}a \to \mathrm{L}A$ and $\mathrm{I}A : \mathrm{L}A \to \mathrm{D}$. Then, the **intensional representation** of I is

$$\mathrm{I}i =< \mathrm{L}a, \{\mathrm{FR}\}\mathrm{I} >$$

where {FR}I is the set of **formation rules for** $\mathrm{I}e$ with

$$\mathrm{I}e = \{< w, \mathrm{f} >: w \in \mathrm{L}a, \mathrm{f} \in \mathrm{D}, < w, \mathrm{f} >\in \mathrm{C}(\{\mathrm{FR}\}\mathrm{I}, \mathrm{L}a)\}$$

where $\mathrm{C}(\{\mathrm{FR}\}\mathrm{I}, \mathrm{L}a)$ is the transitive closure of {FR}I over $\mathrm{L}a$.

Example – Interpretation function

Consider the set of formulas of the form C1 ⊓ C2, where Ci can be an atomic assertion as well as a complex assertion. Consider the formation rules generating them.

We formalize the intuition that C1⊓C2 denotes the intersection of the interpretations of two atomic assertions as follows:

IAC (<concept>) ::= IA(<concept>)

IAC (<awff> ⊓ <awff>) ::= IAC (<awff>) ∩ IAC (<awff>)

which can be used to implemented the following sequence of rewrites

I(C1 ⊓ C2) = IAC (C1 ⊓ C2) = IA(C1) ∩ IA(C2) = C1 ∩ C2

Example – Interpretation function (cont.)

Thus, for instance,

$$\begin{aligned}
&\mathrm{I}(\,(\text{person} \sqcap \text{woman}) \sqcap \text{dog}) = \\
&\mathrm{I}AC\,(\,(\text{person} \sqcap \text{woman}) \sqcap \text{dog}) = \\
&\mathrm{I}AC\,(\text{person} \sqcap \text{woman}) \cap \mathrm{I}A(\text{dog}) = \\
&\mathrm{I}A(\text{person}) \cap \mathrm{I}A(\text{woman})) \cap \text{dog} = \\
&(\text{person} \cap \text{woman}) \cap \text{dog} = \\
&\text{woman} \cap \text{dog} = \emptyset
\end{aligned}$$

where we have assumed to know that women are persons and dogs are disjoint from persons.

Language of complex formulas

Definition (Language of complex formulas, formation rules) Let Li =<Lia, Lic > be a language intensionally defined. Then the **intensional representation** of Lic is

$$\mathrm{L}ic =< \mathrm{L}ea, \{\mathrm{FR}\}c >$$

where $\{\mathrm{FR}\}c$ is the set of **formation rules for** Lec with

$$\mathrm{L}ec = \{w : w \in \mathrm{C}(\{\mathrm{FR}\}c, \mathrm{L}ea)\}$$

where: Lea is as from above and, C($\{\mathrm{FR}\}c$, Lea) is the transitive closure of $\{\mathrm{FR}\}c$ on Lea (see above).

Example - Language of complex formulas

Consider the language defined in Example 8.6 which allows for atomic complex formulas of shape $A1$ xor $A2$ where Ai is any formula. The BNF generating this language consists of the following two formation rules:

<wff> ::= <awff>

<wff> ::= <wff> xor <wff>

where <awff> is a non-terminal symbol which can be grounded into any atomic formula in La.

Entailment relation

Definition (Entailment relation, intensional representation)
Let M ⊆ D be a model and T ⊆ L a theory. Let the entailment relation be defined as |= ⊆ M × T. Then, the **intensional representation** of |= is

$$|=_i = \langle D, L\{FR\} \rangle$$

where {FR} is the set of **formation rules of** |=e, with

$$|=_e = \{\langle f, w \rangle : f \in M, w \in L, \langle f, w \rangle \in C(\{FR\}, D, L)\}$$

where C({FR},D, L) is the transitive closure of {FR} over <D,L>.

Entailment relation - example

Take L = La ∪ Lc , where the formation rules for Lc, are as follows

<wff> ::= <awff>
<wff> ::= <wff> xor <wff>

Then we define the entailment relation with the following recognition rules:

M |= <awff> ::= I(<awff>)
M |= <wff1> xor <wff2> ::=
M |= <wff1> and M not|= <wff2> |
M not|= <wff1> and M |= <wff2>

We have the following examples (where a, ai are atomic formulas).

M |= a	if I(a) ∈ M
M not|= a	if I(a) ∉ M
M |= $a1$ xor $a2$	if [I($a1$) ∈ M and I($a2$) ∉ M | I($a2$) ∈ M and I($a1$) ∉ M]
M |= {$w1$, $w2$}	if M |= $w1$ and M |= $w2$

Logics

Definition (Logic, intensional representation) Let L=< L, D, I, |=>, be a logic defined for the same domain of interpretation Di of ˆW i . Then, the **intensional representation** Li of L is defined as:

$$\mathrm{L}i =< \mathrm{L}i\,,\ \mathrm{D}i\,,\ \mathrm{I}_i\,,\ |=_i>, \qquad \text{with } \mathrm{L}i =< \mathrm{L}ia,\ \mathrm{L}ic >$$

and

$$\mathrm{L}ia\ =<\ \mathrm{A}a,\ \{\mathrm{FR}\}a >$$
$$\mathrm{L}ic\ =<\ \mathrm{L}ea,\ \{\mathrm{FR}\}c >$$
$$\mathrm{I}_i\ =<\ \mathrm{L}ea,\ \{\mathrm{FR}\}\mathrm{I}\ >$$
$$|=_i\ =<\ \mathrm{D},\mathrm{L}e,\ \{\mathrm{FR}\}\ >$$

Li , Lia, Lic, Ii , |=i are the **stencils** used to generate a logic

Logics, models and theories – The practice

1. Select a Logic (crucial representation choice)

$$\mathcal{L} = < L, D, I, \models >,$$

2. Agree on L, I (… and therefore D)

3. Agree on $\models$ (… and therefore reasoning principles)

4. Construct TA = $\{a\} \subseteq LA$

5. The model M = {f} ⊆ D is automatically defined

NOTE: Agreement is on linguistic representation, based on a shared understanding of what language means, and on reasoning mechanism (shared understanding?)

NOTE 2: agreement must be formalized

Reasoning problems

Reasoning Problem (Model checking) Given T and M, check whether M |= T

Reasoning Problem (Satisfiability) Given T , check whether there exists M such that M |= T

Reasoning Problem (Validity) Given T , check whether for all M, M |= T

Reasoning Problem (Unsatisfiability) Given T , check whether there is no M such that M |= T

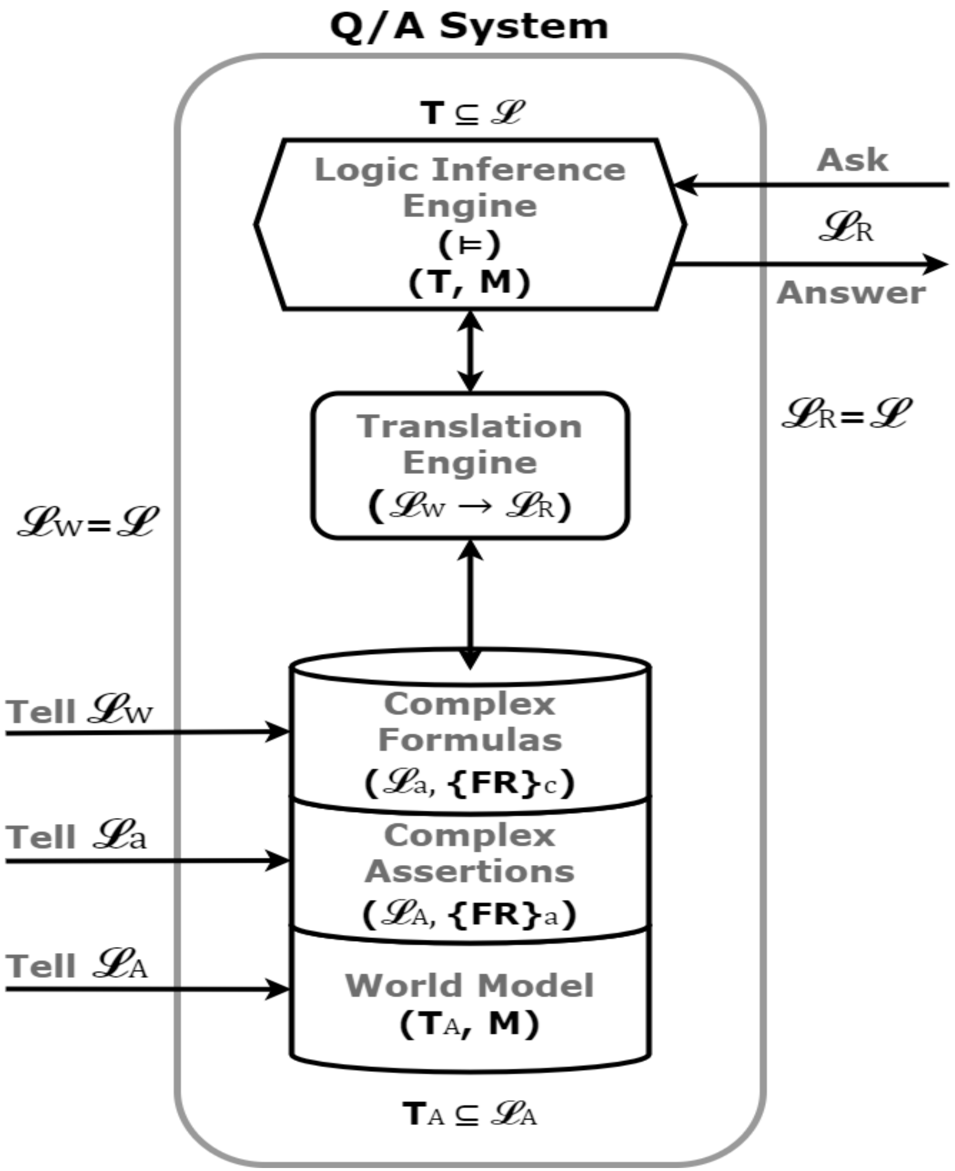
Reasoning Problem (Logical consequence) Given T1, T2 and a set of reference models {M}, check whether

$$T1 \mid=_{\{M\}} T2$$

Reasoning Problem (Logical equivalence) Given T1, T2 and a set of reference models {M}, check whether

$$T1 \mid=_{\{M\}} T2 \text{ and } T2 \mid=_{\{M\}} T1$$

Using a Logic



Expressivity vs. Efficiency

Observation (Logic, selection trade-offs) Any logics can be characterized by two main parameters:

- **Expressivity**, that is, the level of detail at which the problem is expressed, depending on the syntax of the language of the logic;
- **Computational efficiency**, that is how much it costs, in terms of space and time, to reason and answer queries in that language.

Expressivity vs. Efficiency (cont.)

- More expressivity allows for a more refined and precise modeling of the problem but it also generates longer and more complicated formulas.
- The modeler must find the right trade-off between *expressiveness* and *computational complexity*.
- Here the choice of the representation language L =< La, Lc > is crucial. The computational complexity of both La and Lc ranges in fact from polynomial to exponential and beyond.
- There is also an issue of *(un)decidability*, namely the possibility for the reasoner, on certain queries, to get into an infinite loop, never terminate and, therefore, never return an answer.

Logics

Intensional representation

Sept 13,2023