UNIVERSITY OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

Know dive

World models

From assertional to definitional languages

Sept 29,2023

Assertional languages - limitations

Observation 8.1 (Assertional languages, limitations). The description of domains and models using *only* assertions is very limited. One would like to have more flexible ways to describe them.

What assertional languages do not represent – (1) definitions

Definitional languages are motivated by the fact there is information that cannot be easily and intuitively in an assertional language.

- *Definitions characterizing our knowledge of the world* (cats are animals, bachelor is un unmarried man, car is a synonym of automobile, the brother of my father is my uncle, cars are vehicles with four wheels carrying people)
- *Definitions characterizing the world* (Bruno is my uncle, which I know because it is the brother of my father, but this is not written anywhere .., commonsense reasoning)
- *Descriptions of general properties of the worlds* (all cars are produced by a car maker, quite different from maker(car1, BMW), maker(car2, VW))

From assertions to definitions (reprise)

Intution (Assertion). An assertion α has one of the following four (five) forms

$C(e)$, $Pn(e1, \ldots, en)$, $C1 \leq C2$, $C1 \equiv C2$, $Pn(C1, \ldots, Cn)$

where each assertion denotes a fact.

Intuition (definition). Construct a definition by substituting a primitive concept or property with a complex concept or property, for instance *C2* in $C1 \leq C2$ or $C1 \equiv C2$.

Atomic and complex assertions - example

If C1 and C2 are concepts, then, C1 ⊓ C2 (interpreted as set intersection), is also a concept description, where Ci can be an atomic assertion as well as a complex assertion.

Examples of assertions in this language are (";" is used to separate different formulas):

"(being) a person" ; "(having) blond hair" ; "(having) a dog" ;
"blond hair ⊓ " a person" ;

… to be used in definitions

"a blond person" ≡ "blond hair" ⊓ "a person"
"a car" ≤ "a vehicle" ⊓ "four wheels"

Atomic formulas – modeling definitions

Definition 8.3 (Atomic assertions, complex assertions, atomic formulas)
Given a language L = La ∪ Lc, La is defined as

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

where: LA is an assertional language, that we also call a **language of atomic assertions**, and LAC is a **language of complex assertions**. La is a language of **atomic formulas.**

Observation (Atomic formulas) Assertions are atomic formulas, but some atomic formulas are not assertions. The key property of atomic formulas and assertions is that they are interpreted by an interpretation function.

In other words the meaning of atomic formulas, like that of assertions, can be computed directly from the domain.

Language of atomic formulas (Definitional language)

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

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

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

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

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

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.

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.

Example – Definitional language

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

Definitions:

<concept> ≡ <awff>

Complex formulas:

<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.

Transitive closure

Observation (Transitive closure) A transitive closure is the minimal set of formulas which can be obtained by recursively applying the formation rules to their own results, starting from initial set of formulas.

$$A ::= B \sqcap C \sqcap C \sqcap C \ldots$$

NOTE: The initial set of formulas are black boxes for the formation rules in the sense that they can compose the initial formulas into into complex formulas but cannot change their internal structure.

From assertions to definitions (continued)

Assertions in assertional languages

$C(e)$, $Pn\,(e1, \ldots, en)$, $C1 \leq C2$, $C1 \equiv C2$, $Pn\,(C1, \ldots, Cn)$

Assertions in definitional languages -Two key differences

- In Definitional languages you can substitute a primitive concept or property with a complex concept or property, for instance *C2* in $C1 \leq C2$ or $C1 \equiv C2$.
- Assertions hold *de facto* (they denote facts), definitions hold *de dicto* (they are just different ways to describe the same fact).

Definitional languages - example

The following are examples of definitional languages:

- All the natural languages, as used by people in their everyday life;
- The language of arithmetics which describes how to define plus and minus from successor, and times and divides from plus and minus . The language of arithmetic is a simplified natural language which allows to mention, among others, numbers, variables, plus, minus, times, and also to compose phrases in more complex phrases;
- Relational database (DB) languages do not extend to definitional languages;
- Entity-relationship (ER) languages do not extend to definitional languages
- KGs do not extend to definitional languages.

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 IACi is

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

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.

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:

$$\mathrm{I}AC\ (\text{<concept>}) ::= \mathrm{I}A(\text{<concept>})$$

$$\mathrm{I}AC\ (\text{<awff>} \sqcap \text{<awff>}) ::= \mathrm{I}AC\ (\text{<awff>}) \cap \mathrm{I}AC\ (\text{<awff>})$$

which can be used to implemented the following sequence of rewrites

$$\mathrm{I}(C1 \sqcap C2) = \mathrm{I}AC\ (C1 \sqcap C2) = \mathrm{I}A(C1) \cap \mathrm{I}A(C2) = C1 \cap C2$$

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 (in the reference domain of interpretation) women are persons and dogs are disjoint from persons.

World model, intensional representation

Definition 6.10 (World Model, intensional representation) Given a **World Model**

$$W = < L_a, D, I >,$$

its **intensional representation** W_i is defined as

$$W_i =< L_{ia} , D_i , I_i>$$

World models, models and theories – The practice

1. Select the world model (crucial representation choice)

$$W_i =< L_{ia}, D_i , I_{ia} >$$

2. Agree on L_{ia}, I_i (… and therefore D)

3. Construct $T = \{a\} \subseteq L_a$ (assertions and definitions)

4. The model $M = \{f\} \subseteq D$ is automatically defined

NOTE: Agreement is only on linguistic representation, based on a shared understanding of what language means

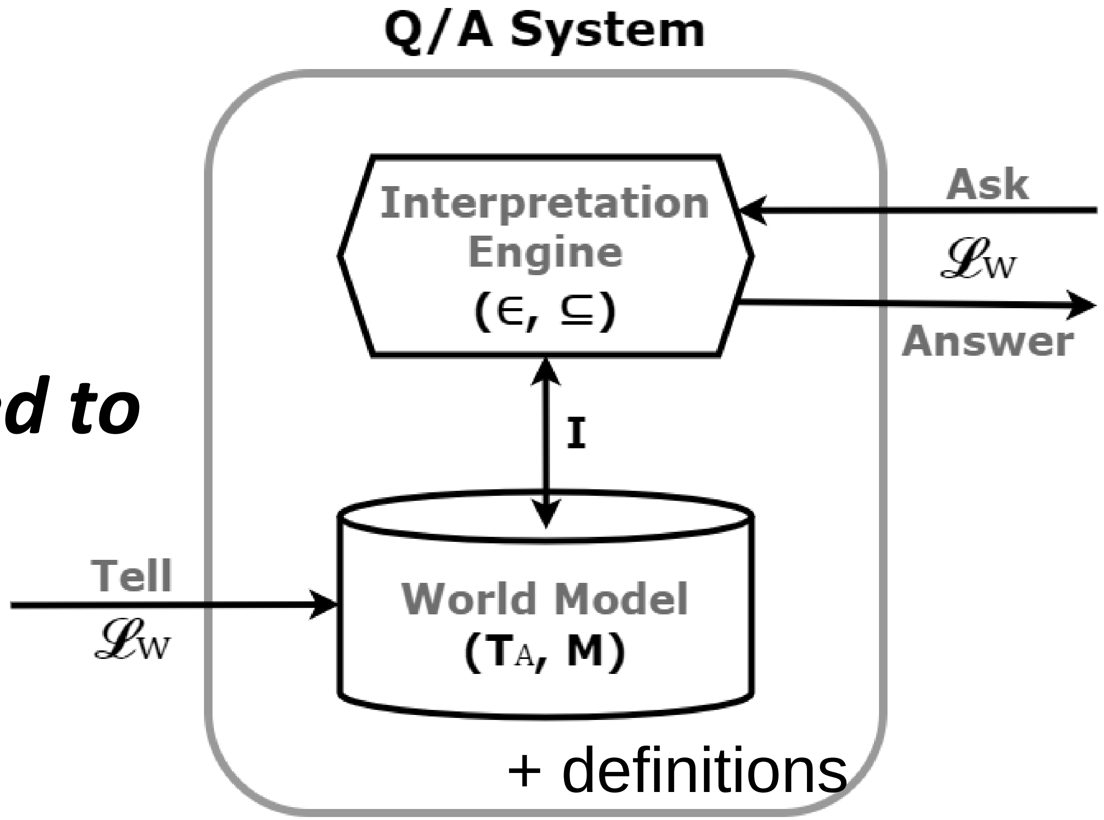
Using a world model

BUT:

M |= a

cannot be reduced to

IA(a) ∈ M



Which questions and answers?

Reasoning problems!

Reasoning problems (with respect a world model)

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

World models

From assertional to definitional languages

Sept 29,2023