

Yesterday: "Symbolic Reasoning"

Background: Philosophical logics, mathematical logics, model theory aspects: human reasoning about properties of the logic).

Each logic, and thus also First-Order Logic provides a framework that can be used for symbolic reasoning:

FOL Formulas are strings, FOL reasoning are algorithms that work on their parse trees.

=> symbolic reasoning: all about Syntax, not Semantics

Formulas are evaluated wrt. first-order-logic structures/interpretations

Syntax: the symbols used for writing formulas:

- * logical symbols: \wedge, \exists, \dots
- * variables: x, y, \dots
- * depending on the application: predicate symbols and function symbols, "signature" Σ

for mondial: $\Sigma = \{Country, City, name, hasCapital, \dots\}$

FOL Structure: $\mathcal{S} = (I, \mathcal{D})$

\mathcal{D} is the domain ... the things in the real world.

I maps the symbols from Σ to the domain ...

Example: our "real-world-application" contains a (green) frog, and strings and numbers:

$\mathcal{D} = \{ \text{🐸} \} \cup \text{Strings} \cup \text{Numbers} \dots$

Signature to talk about the frog and its properties: (1-ary and 2-ary predicates and constant symbols)

$\Sigma = \{ \text{Frog}/1, \text{Green}/1, \text{name}/2, \text{bob}/c0 \}$

Interpret the symbols in OUR structure/model \mathcal{S} (=current situation):

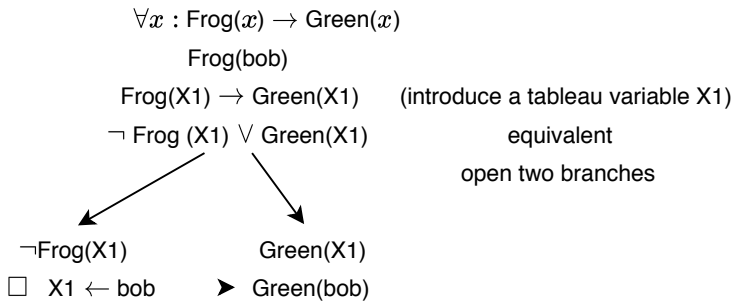
- $I(\text{bob}) = (\text{🐸})$ (an element from \mathcal{D})
- $I(\text{name}) = \{ (\text{🐸}, \text{"Bob"}), \dots \}$ (a set of 2-tuples over \mathcal{D})
- $I(\text{Frog}) = \{ (\text{🐸}), \dots \}$ (a set of 1-tuples over \mathcal{D})

Knowledge base \mathcal{K} : all frogs are green.

$\forall x : \text{Frog}(x) \rightarrow \text{Green}(x)$

Our \mathcal{S} must be a model of \mathcal{K} :

Tableau calculus: what can we derive?



=> conclusion by reasoning: bob must be green in our \mathcal{S}

$\Rightarrow I(\text{Green}) \supseteq I(\text{bob})$

$I(\text{Green}) \supseteq \{ (\text{🐸}) \}$

I practically is a database, containing unary and binary tables:

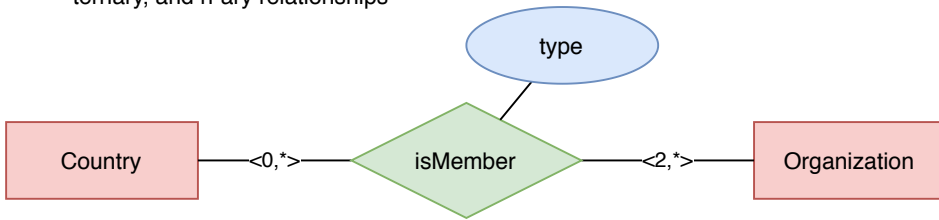
(note: DB is only on the syntax level, so bob <-> 🐸)

name	
bob	"Bob"
:	:

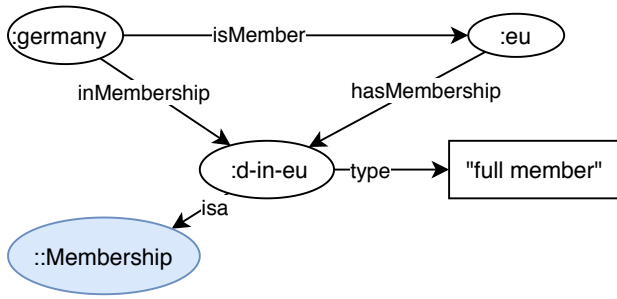
Frog
bob

(the constant bob/c0 is like an object identifier)

RDF (and also UML class diagrams): only binary relationships
ER: relationships may have attributes;
ternary, and n-ary relationships



RDF: Reification of the "Membership" (relationship) instance



Note: the same holds when modeling relationships with attributes in UML class diagrams. UML provides "association classes" for such cases.

Reification has also to be used for ternary (or n -ary in general) relationships in UML, and in RDF. True ternary relationships are rare. Those who did the BSc here may remember the first exercise sheet of the "Introduction to Databases" lecture.

The reference solutions for it (in German) can be found at <https://www.dbis.informatik.uni-goettingen.de/Teaching/DB-WS1920/DBBlatt1ML.pdf>.