The language of the first-order logic (FOL)

Three things define a declarative language:

- syntax – what groups of symbols are valid and in what order

  “the car that I drive”

  “drive the car I that”

- semantics – what the well-formed sentences mean – some expressions may not mean anything

  “blue holidays runs”

- pragmatics – how meaningful expressions are used

  “there is someone behind you”

These slides are mainly based on Ronald J. Brachman, Hector J. Levesque. Knowledge representation and reasoning, Morgan Kaufmann, 2004.

The syntax in FOL

There are two types of symbols: logical and nonlogical.

The logical symbols (like the reserved words in a programming language):

a) Punctuation: ‘(‘, ‘)’, ‘.’

b) Connectives:

- $\neg$, $\wedge$, $\vee$ (in descending order of priority)
- quantifiers $\exists$, $\forall$
- logical equality = (is a special symbol, not a predicate)

c) Variables: an infinite set of symbols (denoted by x, y, z with/without subscripts and superscripts

The syntax in FOL

The nonlogical symbols (like the identifiers in a programming language)

- they have an application-dependent meaning or use
- they have arity

a) Function symbols – start with a lower-case letter; written in mixed case

- examples: bestFriend
  a, b, c, f, g, h with/without sub/superscripts
- by convention – if arity is zero, we use symbols a, b, c – they are called constants.

b) Predicate symbols - start with an upper-case letter; written in mixed case

- examples: OlderThan
  P, Q, R with/without sub/superscripts
- Predicate symbols of arity 0 are called propositional symbols.

The syntax in FOL

In FOL there are two types of valid syntactic expressions:

1) Terms
   a) Every variable is a term;
   b) If $t_1,\ldots,t_n$ are terms and f is a function symbol with n arguments, then $f(t_1,\ldots,t_n)$ is a term

2) Formulas
   a) If $t_1,\ldots,t_n$ are terms and P is a predicate symbol with n arguments, then $P(t_1,\ldots,t_n)$ is a formula
   b) If t_1 and t_2 are terms, then $t_1=t_2$ is a formula
   c) If α and β are formulas and x is a variable, then $\neg\alpha$, $\alpha\wedge\beta$, $\alpha\vee\beta$, $\forall x.\alpha$ and $\exists x.\alpha$ are formulas

called atomic formulas (or atoms)

The propositional subset of FOL is the language with no terms or quantifiers, but only propositional symbols are used:

$$(P\wedge\neg Q)\vee R$$

The syntax in FOL

- Abbreviations:
  - $\alpha \supset \beta$ for $(\neg\alpha \lor \beta)$ (also denoted by $\rightarrow$ or $\Longrightarrow$)
  - $\alpha \equiv \beta$ for $(\alpha \supset \beta) \land (\beta \supset \alpha)$ (also denoted by $\Leftrightarrow$)

- A variable occurrence is bound in a formula if it lies within the scope of a quantifier. Otherwise, it is free.

$$\forall y.\ P(x) \land \exists x\ [P(y) \lor Q(x)]$$

free

bound

bound

- Obs. Some authors consider the occurrence of the variable just after the quantifiers neither free nor bound.

- Definition. A sentence in FOL is any formula without free variables

$$\forall y.\ \exists x\ [P(y) \lor Q(x)]$$

The semantics in FOL

- The meaning of a sentence derives from the interpretation of the nonlogical symbols involved.

- Since nonlogical symbols are application-dependent, we are looking for a clear specification of the meaning of a sentences as a function of the interpretation of the predicate and function symbols.

- For example, the meaning of “DemocraticCountry” can be specified by “objects” that represent those countries that are considered to be democratic. There may be different interpretations (some people consider some countries democratic, others don’t). In terms of FOL, we are not interested to say what “DemocraticCountry” means according to the dictionary.

Interpretations in FOL

An interpretation $\mathcal{I}$ is a pair <D,I>, where

- D is a non-empty set of objects, called the domain of the interpretation (it can be anything)
- I is the interpretation mapping, that assigns a meaning to the predicate and function symbols

- If P is a predicate symbol of arity n, then I[P] is a *n*-ary relation over D:

$$I[P] \subseteq \underbrace{D \times \ldots \times D}_{n \text{ times}}$$

- If f is a function symbol of arity n, then I[f] is a *n*-ary function over D:

$$I[f] \in [\underbrace{D \times \ldots \times D}_{n \text{ times}} \rightarrow D]$$

Examples

D={d_1, d_2, d_3, anna, mary, george, john, …}

- Dog is a unary predicate symbol
  I[Dog]={d_1, d_2, d_3} – the set of dogs in this interpretation

- OlderThan is a binary predicate symbol
  I[OlderThan]={ (anna, mary), (anna, george)}

- bestFriend is a unary function symbol
  I[bestFriend]: $\mathrm{D} \to \mathrm{D}$
  I[bestFriend](anna)=mary

- firstChildOf is a binary function symbol
  I[firstChildOf]: $\mathrm{D}\times\mathrm{D} \to \mathrm{D}$
  I[firstChildOf](marry,george)=john

- motherOfThreeChildren is a 3-ary function symbol
  I[motherOfThreeChildren]: $\mathrm{D}\times\mathrm{D}\times\mathrm{D} \to \mathrm{D}$
  I[motherOfThreeChildren](john,marry,george)=anna

Obs. johnSmith is a constant I[johnSmith]=john $\in \mathrm{D}$

Interpretations in FOL

- A useful alternative to interpret predicates symbols is in terms of their characteristic function. Thus, for P a predicate of arity n, we view I[P] as an *n*-ary function to {0,1}:

$$\mathrm{I}[P] \in [\underbrace{D \times \ldots \times D}_{n \text{ times}} \rightarrow \{0,1\}]$$

- The two specifications are related as following: a tuple of objects is considered to be in the relation over D iff the characteristic function over those objects has value 1 (e.g., $\mathrm{I}[\mathrm{Dog}](d_1)=1$).

- If P is a predicate of arity 0, I[P] is either 1 or 0 (true/false).

- For the propositional subset of FOL, the domain D can be ignored and the interpretation can be seen as a mapping I from the propositional symbols to either 0 or 1.