First-order Logic[1]

Lecture 4

[1] The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello

First order languages

Summary

- ♢ Motivation [R&N, 8.1]
- ♢ Syntax [R&N, 8.2]
- ♢ Semantics [R&N, 8.2]
- ♢ Representation in FOL [R&N, 8.3]

Features of propositional logic

Propositional logic allows to represent information that is

- ♢ **partial**
- ♢ **disjunctive**
- ♢ **negated**

(unlike most data structures and databases)

Features of propositional logic

Propositional logic is **declarative**:

- syntax
- semantics
- inference

Syntactic expressions correspond to assertions and **inference** allows to derive new facts

Features of propositional logic

Propositional logic is **compositional**:
meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$

Meaning in propositional logic is **context-independent**
(unlike natural language, where meaning depends on context)

take the book next to the keyboard

Modeling the Wumpus world: breeze and stench

for each Location $L_{i,j}$:

$\diamondsuit$ breeze is perceived in the locations adjacent to a pit, e.g.

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$\diamondsuit$ stench is perceived in the locations adjacent to a wumpus, e.g.

$$S_{1,1} \Leftrightarrow (W_{1,2} \vee W_{2,1})$$

Modeling the Wumpus world: wumpus

♢ there is at least one wumpus

$W_{1,1} \vee W_{1,2} \vee W_{1,2} \vee \ldots \vee W_{4,4}$

♢ there is at most one wumpus, for **each pair of locations**:

$\neg W_{1,1} \vee \neg W_{1,2}$

$\ldots$

$\neg W_{4,3} \vee \neg W_{4,4}$

Modeling the Wumpus world: state

State= Location (Pose) + Arrow + Wumpus

Fluents:
$L^0_{1,1}, FacingEast^0, HaveArrow^0, WumpusAlive^0$

Fluents have a superscript denoting the time t, as their value changes over time.

Modeling the Wumpus world: agent perceptions

Assumption: percepts refer to the current location,
e.g. Stench is perceived at time t: $Stench^t$

Turning perceptions into knowledge about the world:

◇ If the agent is in location x, y at time t, then perceiving breeze is equivalent to asserting $B_{x,y}$:

$$L_{x,y}^t \Rightarrow (Breeze^t \Leftrightarrow B_{x,y})$$
$$L_{x,y}^t \Rightarrow (Stench^t \Leftrightarrow S_{x,y})$$

This relationship must hold for every time step t

Modeling the Wumpus world: transition model

Actions are also labelled with the time of execution:
e.g. $Forward^0$

Effect Axioms: how does the world change as a result of the execution of an action

$$L^0_{1,1} \wedge FacingEast^0 \wedge Forward^0 \;\Rightarrow\; L^1_{2,1}$$

We need similar sentences for:

- $\diamondsuit$ each location,
- $\diamondsuit$ each time step,
- $\diamondsuit$ each action ($Grab, Shoot, Climb, TurnLeft, TurnRight$)

Modeling the Wumpus world: frame problem

For each action:

$$Forward^t \Rightarrow (HaveArrow^t \Leftrightarrow HaveArrow^{t+1})$$
$$Forward^t \Rightarrow (WumpusAlive^t \Leftrightarrow WumpusAlive^{t+1})$$

...

Alternative:

$$HaveArrow^{t+1} \Leftrightarrow (HaveArrow^t \wedge \neg Shoot^t)$$

...

$$\begin{aligned} L_{1,1}^{t+1} \Leftrightarrow\ & (L_{1,1}^t \wedge (\neg Forward^t \vee Bump^{t+1}) \\ & \vee (L_{1,2}^t \wedge (South^t \wedge Forward^t) \\ & \vee (L_{1,2}^t \wedge (West^t \wedge Forward^t) \end{aligned}$$

Modeling the Wumpus world: finally ...

Given the perception at $t = 0$:

$$\neg Stench^0 \wedge \neg Breeze^0 \wedge \neg Glitter^0 \wedge \neg Bump^0 \wedge \neg Scream^0$$

Action: $Forward^0$ can proven safe!

A useful definition:

$$OK^t_{x,y} \;\Leftrightarrow\; (\neg P_{x,y} \wedge \neg(W_{x,y} \wedge WumpusAlive^t))$$

Modeling the Wumpus world: next ...

After execution of the action $Forward^0$ one can check in the KB the new location of the agent and the value of the percepts

$$\neg Stench^1 \wedge Breeze^1 \wedge \neg Glitter^1 \wedge \neg Bump^1 \wedge \neg Scream^1$$

Action: $TurnRight^1$

$$\neg Stench^2 \wedge Breeze^2 \wedge \neg Glitter^2 \wedge \neg Bump^2 \wedge \neg Scream^2$$

$\ldots$

Not the right representation tool

The number of rules grows fast ...

- writing of the rules can be done by a program
- number of rules can make reasoning inefficient

Propositional logic has very limited expressive power (unlike natural language)

First-order logic

Whereas propositional logic assumes world contains *facts*, first-order logic (like natural language) assumes the world contains

- Objects: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .
- Relations: (unary) red, round, bogus, prime, multistoried . . .,
  (n-ary) brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .
- Functions: father of, best friend, third inning of, one more than, beginning of . . .

Logics in general

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	degree of truth $\in [0, 1]$	known interval value

The language: logical symbols

A *first order language* $\mathcal{L}$ is built upon the following sets of symbols:

Logical symbols

- propositional connectives: $\neg$,$\wedge$,$\vee$, $\Rightarrow$ and $\Leftrightarrow$;
- propositional constants $\top$ and $\perp$;
- equality $=$ (not always included);
- separators '(', ')' and ',';
- A denumerable set of *individual variable* symbols x_1, x_2, $\ldots$;
- *universal quantification* $\forall$;
- *existential quantification* $\exists$.

The language: parameters

- A denumerable set of *predicate symbols*, each associated with a positive integer n, arity. A predicate with arity n is called n-ary;
- A denumerable set of *function symbols*, each associated with a positive integer n, arity. A function with arity n is called n-ary;
- A denumerable set of *constant symbols*.

Note: *constant symbols* are sometimes regarded as *function symbols* with arity $= 0$.

Examples 1

The *pure predicate language*:

n-ary predicate symbols: $P_1^n, P_2^n, \ldots$;
constant symbols: $c_1, c_2, \ldots$;
no function symbols, no equality.

Example 2

The language of *elementary number theory*:

Equality;
predicate symbols: only the binary predicate $<$;
constant symbols: 0;
function symbols: a unary function symbol s, successor function, (additionally, the binary function symbols $+$ and $\times$, addition and multiplication)

Terms

The set $\textsc{Term}$ of the *terms* of $\mathcal{L}$ is inductively defined as follows:

1. Every constant and variable symbol is a term;
2. If $t_1 \ldots t_n$ are terms and f is a n-ary function symbol, $f(t_1, \ldots, t_n)$ is a term (*functional term*).

Examples: x, c, $f(x, y + c)$,...

Atomic formulae

The set $\textsc{Atom}$ of the *atomic formulae* is inductively defined as follows:

1. $\bot$ and $\top$ are atoms;
2. If t_1 e t_2 are terms then $t_1 = t_2$ is an atom;
3. If $t_1, \ldots, t_n$ are terms and P is a n-ary predicate symbol $P(t_1, \ldots, t_n)$ is an atom.

Examples: $P(x)$, $Q(x, c)$, $R(x, f(x, y + c))$,. . .

Examples: terms and atoms

◊ Terms:

$homeOf(Giovanni)$

$batteryOf(MyHonda)$

$authorOf(Hamlet)$

$x + (2 \times y) \qquad f(x, y, g(z, t + 3))$

◊ Atoms:

$Big(homeOf(Giovanni))$

$Bigger(homeOf(Giovanni), homeOf(Filippo))$

$Low(batteryOf(MyHonda))$

$authorOf(Hamlet) = Shakespeare$

$x + (2 \times y) = 0 \qquad f(x, y, g(z, t + 3)) = f(x, y, w)$

First Order Formulae

The set of *formulae* of $\mathcal{L}$ is inductively defined as follows:

- Every atom is a formula;
- If A is a formula $\neg A$ is a formula;
- If $\circ$ is a binary logical operator, A and B formulae, $A \circ B$ is a formula;
- If A is a formula, x a variable, $\forall x A$ and $\exists x A$ are formulae.

<u>Examples</u>: $P(x)$, $\exists x Q(x, c)$, $\forall z R(x, f(x, y + c))$,...

$\circ = \wedge, \vee, \Rightarrow, \Leftrightarrow$

Examples: formulae

$$Big(homeOf(Giovanni)) \wedge$$
$$Bigger(homeOf(Giovanni), homeOf(Filippo))$$

$$\forall t\ Low(batteryOf(MyHonda), t)$$
$$\exists x\ x = authorOf(Hamlet) \wedge Born(StratfordOnAvon, x)$$
$$\forall x\ x + (2 \times y) = 0$$
$$\neg(f(x, y, g(z, t + 3)) = f(x, y, w))$$

Operator precedence

Precedence among logical operators is defined as follows:

$$\forall, \exists, \neg, \wedge, \vee, \ \Rightarrow \ , \ \Leftrightarrow$$

and, as in PL, all operators are right associative.

$$\forall x P(x) \ \Rightarrow \ \exists y \exists z Q(y, z) \wedge \neg \exists x R(x)$$

would be:

$$(\forall x P(x)) \ \Rightarrow \ (\exists y (\exists z Q(y, z))) \wedge \neg (\exists x R(x)).$$

but we follow R&N: the parentheses enclosing the scope of a quantifier are omitted, when the scope is the whole formula: i.e.: $\forall\, x \ \ P(x) \Rightarrow Q(x)$

Note: in any case the inner occurrence of x is bound to the innermost existential quantifier

Notation variants

Syntax	**RN**	**Others**
Negation (not)	$\neg P$	$\sim P \quad \overline{P}$
Conjunction (and)	$P \wedge Q$	$P \& Q \quad P \cdot Q \quad PQ \quad P, Q$
Disjunction (or)	$P \vee Q$	$P \mid Q \quad P; Q \quad P + Q$
Implication (if)	$P \ \Rightarrow \ Q$	$P \rightarrow Q \quad P \supset Q$
Equivalence (iff)	$P \ \Leftrightarrow \ Q$	$P \equiv Q \quad P \leftrightarrow Q$
Universal (forall)	$\forall x \ \ P(x)$	$(\forall x)P(x) \quad \bigwedge x \ P(x) \quad P(x)$
Existential (exists)	$\exists x \ \ P(x)$	$(\exists x)P(x) \quad \bigvee x \ P(x) \quad P(Sk_i)$
Relation	$R(x, y)$	$(R \ x \ y) \quad Rxy \quad xRy$

Variables in terms atomic formulae

Let $var(t)$ the set of variables of *term* t.

A term/atom is **ground** if it does not contain variables.

An occurrence of a variable x in a formula is **free** if it is not in the scope of a quantifier, *bound* if not free.

Examples:

$P(x), \exists x Q(x, c), \forall z R(x, f(x, y + c)),$

$\forall z R(x, f(x, y+c)) \wedge P(z)$ is $\forall z(R(x, f(x, y+c)) \wedge P(z))$

$\forall z R(x, f(x, y + c)) \wedge \exists z P(z)$

A **sentence** – otherwise called **closed formula** – is a formula without free variables.

Some intuition: representation with existentials

Some medical doctors are arrogants

$a)\quad \exists x(medicalDoctor(x) \wedge arrogant(x))$

$a')\quad \exists x(medicalDoctor(x) \Rightarrow arrogant(x))$ OK no doctors!!

Some worker is a car industry employee

$b)\quad \exists x(worker(x) \wedge carIndustryEmployee(x))$

$b')\quad \exists x(worker(x) \Rightarrow carIndustryEmployee(x))$

Representation with universals

All bakers can make appleCakes

$$c) \quad \forall x(Baker(x) \ \Rightarrow \ cando(x, appleCake))$$
$$c') \quad \forall x(Baker(x) \wedge cando(x, appleCake))$$

all bakers!!

Other examples: nested quantifiers

$\forall x \exists y\ loves(x, y)$ everyone has somebody to love
$\exists x \forall y\ loves(x, y)$ the great lover

Check the use of parameters:

$\forall x \exists y\ loves(y, x)$ somebody loves us
$\exists x \forall y\ loves(y, x)$ the great beloved

Interpretations and models

A *structure for the language* $\mathcal{L}$ is a pair $\mathfrak{A} = \langle D, I \rangle$ where:

- D is a non empty set called *domain* of $\mathfrak{A}$;
- I is a function called *interpretation*. I maps:
  - every constant symbol c into an element $c^I \in D$;
  - every n-ary function symbol f into a function $f^I : D^n \rightarrow D$;
  - every n-ary predicate symbol P into a n-ary relation $P^I \subseteq D^n$.

Note on terminology

($\neq$ RN):

$\diamondsuit$ we introduce the term **structure** to name together both the **domain** (i.e. the set of individual objects) and the **interpretation** defines the meaning of predicates, functions and constants

$\diamondsuit$ we use the term **model** for a structure where a formula is true (NOT the assignment to predicates)

$\diamondsuit$ when we talk about the objects, functions and relations in a structure we use the **superscript** I (i.e. c^I, f^I, P^I)

Examples

$$\forall x \exists y P(x, y)$$

D, the set of human beings
$P^I =$ the set of pairs $\langle A, B \rangle$, such that B is *father* of A
All human beings have a father

D, the set of human beings
$P^{I'}$ the set of pairs $\langle A, B \rangle$, such that B is *mother* of A
All human beings have a mother

D the set of natural numbers
P^J the set of pairs $\langle m, n \rangle$, such that $m < n$
For every nat number there is a greater one

Truth of formulae

The truth of a closed formula ϕ, in a structure $\mathfrak{A}$, is denoted as:

$$\phi \text{ is true in } \mathfrak{A}$$

meaning that the structure $\mathfrak{A}$ satisfies ϕ (or is a model of ϕ).

Truth: definition 1

Let $\mathfrak{A} = \langle D, I \rangle$ a structure for the language $\mathcal{L}$

1. $\top$ is true in $\mathfrak{A}$ and $\perp$ is false in $\mathfrak{A}$;
2. if A is a closed atomic formula $P(t_1, \ldots, t_n)$, then
$$P(t_1, \ldots, t_n) \text{ is true in } \mathfrak{A} \text{ iff } \langle t_1^I \ldots t_n^I \rangle \in P^I;$$
3. if A is a closed atomic formula $t_1 = t_2$ then
$$t_1 = t_2 \text{ is true in } \mathfrak{A} \text{ iff } t_1^I = t_2^I;$$
4. $\neg A$ is true in $\mathfrak{A}$ iff A is false in $\mathfrak{A}$;
5. $A \wedge B$ is true in $\mathfrak{A}$ iff A is true in $\mathfrak{A}$ and B is true in $\mathfrak{A}$;
6. $A \vee B$ is true in $\mathfrak{A}$ iff A is true in $\mathfrak{A}$ or B is true in $\mathfrak{A}$;
7. $(A \Rightarrow B)$ is true in $\mathfrak{A}$ iff A implies B is true in $\mathfrak{A}$;

Truth: definition 2

8. $(A \Leftrightarrow B)$ is true in $\mathfrak{A}$ iff
   A is true in $\mathfrak{A}$ and B is true in $\mathfrak{A}$ or A is false in $\mathfrak{A}$ and B is false in $\mathfrak{A}$;
9. $\forall x A$ is true in $\mathfrak{A}$ iff **for every** $d \in D$ we have $A\{d \rightarrow x\}$ is true in $\mathfrak{A}$;
10. $\exists x A$ is true in $\mathfrak{A}$ iff **there exists a** $d \in D$ s.t. $A\{d \rightarrow x\}$ is true in $\mathfrak{A}$.

Remarks:

- we omit the superscript I and write d instead of d^I.
- $A\{d \rightarrow x\}$ denotes that each occurrence of x is interpreted by the object.
- Same as **extended interpretation** by RN, when name conflicts on quantified variables are avoided.

Examples 1

1. $\exists x(P(x) \wedge Q(x))$

1. Verified in $\mathfrak{A}$ iff there exists a $d \in D$ which makes $(P(x) \wedge Q(x))\{d \rightarrow x\}$ true in $\mathfrak{A}$.

There exists a $d \in D$ such that $d \in P^I \cap Q^I$ (D is by definition non empty). Hence the subsets of D associated by I to P and Q, respectively, are non empty and have a non empty intersection.

Examples 2

2. $\exists x(P(x) \Rightarrow Q(x))$

2. Verified in $\mathfrak{A}$ iff there exists a $d \in D$ which makes $(P(x) \Rightarrow Q(x))\{d \to x\}$ true in $\mathfrak{A}$.

There exists $d \in D$ such that $d \in \overline{P^I} \cup Q^I$. Hence, if I associates P with the empty subset of D, the formula $\exists x(P(x) \Rightarrow Q(x))$ becomes true for every domain element.

Examples 3

3. $\forall x(P(x) \wedge Q(x))$

3. Verified in $\mathfrak{A}$ iff for all $d \in D$ $(P(x) \wedge Q(x))\{d \to x\}$ is true in $\mathfrak{A}$. P^I and Q^I both coincide with D !

4. $\forall x(P(x) \Rightarrow Q(x))$

4. Verified in $\mathfrak{A}$ iff for all $d \in D$ $(P(x) \Rightarrow Q(x))\{d \to x\}$ is true in $\mathfrak{A}$. The extension of $P^I \subseteq Q^I$, P^I can be empty or both P^I and Q^I can be empty.

Models, validity, satisfiability

Let A be a sentence.

$\mathfrak{A}$ is a *model* of A, or A is *true* in $\mathfrak{A}$.

A formula $A \in \mathcal{L}$ is *valid* iff it is true in every structure of $\mathcal{L}$, denoted $\models A$.

A set of formulae Γ is *satisfiable* if there exists a structure $\mathfrak{A}$, such that for every $A \in \Gamma$ A is true in $\mathfrak{A}$.

Validity and satisfiability can NOT be easily checked with the truth tables!

Logical entailment, Equivalence

Let KB a set of formulae and A a closed formula.

Then KB *logically entails* A, written $KB \models A$, iff every model of KB is also a model of A (i.e. for every structure $\mathfrak{A}$ of the language such that KB is true in $\mathfrak{A}$, then A is true in $\mathfrak{A}$).

Two formulae P and Q are *semantically* (or *logically*) *equivalent* (written $P \equiv Q$) if for every structure $\mathfrak{A}$ we have that:

P is true in $\mathfrak{A}$ iff Q is true in $\mathfrak{A}$.

Equivalences for quantifiers

Formulae are semantically equivalent if they differ in

- the name of variables in the scope of quantifiers
  $\forall x P(x) \equiv \forall y P(y)$
- the order of quantifiers of the same kind
  $\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y) \equiv \forall x, y P(x, y)$
- the elimination of quantifiers whose variable does not occur in their scope
  $\forall x P(y) \equiv P(y)$

Semantic Equivalence: negation

1. $\forall x P \equiv \neg \exists x \neg P$
2. $\neg \forall x P \equiv \exists x \neg P$
3. $\exists x P \equiv \neg \forall x \neg P$
4. $\neg \exists x P \equiv \forall x \neg P$.

Semantic Equivalence: and, or

Quantifiers are distributive wrt $\wedge$ and $\vee$, but with restrictions:

1. $\forall x P_1 \wedge P_2 \equiv (\forall x P_1) \wedge \forall x P_2$ although useless!
2. $\exists x (P_1 \vee P_2) \equiv \exists x P_1 \vee \exists x P_2$ although useless!

3. $\forall x (P_1 \vee P_2) \equiv (\forall x P_1) \vee P_2$ (only) if $x \notin var(P_2)$
4. $\exists x (P_1 \wedge P_2) \equiv (\exists x P_1) \wedge P_2$ (only) if $x \notin var(P_2)$.

Semantic Equivalence: implication

Let P_2 a formula where x does not occur free

The **quantifier in the antecedent** changes outside

1. $(\exists x P_1) \Rightarrow P_2 \equiv \forall x(P_1 \Rightarrow P_2)$
2. $(\forall x P_1) \Rightarrow P_2 \equiv \exists x(P_1 \Rightarrow P_2)$

$(\exists x P_1) \Rightarrow P_2$

$\neg(\exists x P_1) \vee P_2$

$(\forall x \neg P_1) \vee P_2$

$\forall x(\neg P_1 \vee P_2)$

$\forall x(P_1 \Rightarrow P_2)$

The **quantifier in the consequent** unchanged outside

3. $P_2 \Rightarrow \exists x P_1 \equiv \exists x(P_2 \Rightarrow P_1)$
4. $P_2 \Rightarrow \forall x P_1 \equiv \forall x(P_2 \Rightarrow P_1)$

KB of family relationship

$$\forall x \forall y (father(y,x) \Rightarrow son(x,y))$$
$$\forall x \forall y (mother(y,x) \Rightarrow son(x,y))$$

$$\forall x \forall y \forall z (father(x,y) \wedge father(y,z) \Rightarrow grandfather(x,z))$$
$$\forall x \forall z (\exists y\, (father(x,y) \wedge father(y,z)) \Rightarrow grandfather(x,z))$$

$$\forall x \forall y \forall z (father(x,y) \wedge mother(y,z) \Rightarrow grandfather(x,z))$$
$$\forall x \forall z (\exists y\, (father(x,y) \wedge mother(y,z)) \Rightarrow grandfather(x,z))$$

FOL Knowledge bases

A **knowledge base** is a representation of the knowledge about the world (problem).

- **intensional knowledge**: general laws on the domain of interest
(e.g.$\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y))$)
- **extensional knowledge**: facts about a specific problem instance (situation)
(e.g.$Parent(daniele, michela)$)

The KB is built with TELL and queried with ASK, which relies on **Inference** in FOL (next class).

Interacting with FOL KBs

Consider a family KB in FOL, specifying that fathers are male parents and that daniele is parent of michela and jacopo:

$$Tell(KB, Male(daniele))$$
$$Ask(KB, \exists a \;\; father(daniele, a))$$

i.e., does the KB entail that Daniele is a father?

Answer: $Yes, \{a/michela\}$ $\leftarrow$ substitution (binding list)
Answer: $Yes, \{a/jacopo\}$

Interacting with FOL KBs

Given a sentence S and a substitution σ,
$S\sigma$ denotes the result of plugging σ into S; e.g.,

$$S = Smarter(x, y), \qquad \sigma = \{x/Hillary, y/Bill\}$$
$$S\sigma = Smarter(Hillary, Bill)$$

$Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$

More on substitution next class