Knowledge Representation and Learning

7. First Order Logic, intuition and syntax

Luciano Serafini

Fondazione Bruno Kessler

April 20, 2023

- First order logic is also called predicate logic
- in FOL proposiitions are not atomic elements
- a proposition is a predication about properties and relations between objects
- The set of objects on which FOL predicates can vary
- universal and existential statements are possible in FOL

Expressivity of propositional logic - I

Question

Try to express in Propositional Logic the following statements:

- Mary is a person
- John is a person
- Mary is mortal
- Mary and John are siblings

A solution

Through four atomic propositions p, q, r, and s:

- p that stands for Mary is a person
- q that stands for John is a person
- r that stands for Mary is mortal
- s that stands for Mary and John are siblings

Expressivity of propositional logic - I

Question

Try to express in Propositional Logic the following statements:

- Mary is a person
- John is a person
- Mary is mortal
- Mary and John are siblings

Another msolution

Through more mnemonic atomic propositions:

- `Mary-is-a-person`
- `John-is-a-person`
- `Mary-is-mortal`
- `Mary-and-John-are-siblings`

Problem with previous solution

- `Mary-is-a-person`
- `John-is-a-person`
- `Mary-is-mortal`
- `Mary-and-John-are-siblings`

How do we link `Mary` of the first sentence to `Mary` of the third sentence? Same with `John`. How do we link `Mary` and `Mary-and-John`?

Expressivity of propositional logic - II

Question

Try to express in Propositional Logic the following statements:

- All persons are mortal;
- There is a person who is a spy.

A solution

We can give all people a name and express this fact through atomic propositions:

- $\texttt{Mary-is-mortal} \wedge \texttt{John-is-mortal} \wedge \texttt{Chris-is-mortal} \wedge \ldots \wedge \texttt{Michael-is-mortal}$
- $\texttt{Mary-is-a-spy} \vee \texttt{John-is-a-spy} \vee \texttt{Chris-is-a-spy} \vee \ldots \vee \texttt{Michael-is-a-spy}$

Problem with previous solution

- $\texttt{Mary-is-mortal} \wedge \texttt{John-is-mortal} \wedge \texttt{Chris-is-mortal} \wedge \ldots \wedge \texttt{Michael-is-mortal}$
- $\texttt{Mary-is-a-spy} \vee \texttt{John-is-a-spy} \vee \texttt{Chris-is-a-spy} \vee \ldots \vee \texttt{Michael-is-a-spy}$

The representation is not compact and generalization patterns are difficult to express.
What is we do not know all the people in our "universe"? How can we express the statement independently from the people in the "universe"?

Expressivity of propositional logic - III

Question

Try to express in Propositional Logic the following statements:

- Every natural number is either even or odd

Constants and Predicates

- Mary is a person
- John is a person
- Mary is mortal
- Mary and John are siblings

In FOL it is possible to build an atomic propositions by applying a predicate to constants

- $Person(mary)$
- $Person(john)$
- $Mortal(mary)$
- $Siblings(mary, john)$

Quantifiers and variables

- Every person is mortal;
- There is a person who is a spy;
- Every natural number is either even or odd;

In FOL it is possible to build propositions by applying universal (existential) quantifiers to variables. This allows to quantify to arbitrary objects of the universe.

- $\forall x. Person(x) \rightarrow Mortal(x)$;
- $\exists x. Person(x) \wedge Spy(x)$;
- $\forall x. (Odd(x) \vee Even(x))$

Functions

- The father of Luca is Italian.

In FOL it is possible to build propositions by applying a function to a constant, and then a predicate to the resulting object.

- $Italian(fatherOf(Luca))$

Syntax of FOL

The alphabet of FOL is composed of two sets of symbols:

Logical symbols

- the logical constant $\bot$
- propositional logical connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\equiv$
- the quantifiers $\forall$, $\exists$
- an infinite set of variable symbols $x_1, x_2, \ldots$
- the equality symbol $=$. (optional)

Non Logical symbols

- a set $c_1, c_2, \ldots$ of constant symbols
- a set $f_1, f_2, \ldots$ of functional symbols each of which is associated with its arity (i.e., number of arguments)
- a set $P_1, P_2, \ldots$ of relational symbols each of which is associated with its arity (i.e., number of arguments)

Non logical symbols - Example

Non logical symbols depends from the domain we want to model. Their must have an intuitive interpretation on such a domain.

Example (Domain of arithmetics)

symbols	type	arity	intuitive interpretation
0	constant	0^*	the smallest natural number
$succ(\cdot)$	function	1	the function that given a number returns its successor
$+(\cdot,\cdot)$	function	2	the function that given two numbers returns the number corresponding to the sum of the two
$<(\cdot,\cdot)$	relation	2	the less then relation between natural numbers

* A constant can be considered as a function with arity equal to 0

Non logical symbols - Example

Example (Domain of arithmetics - extended)

The basic language of arithmetics can be extended with further symbols e.g:

symbols	type	arity	intuitive interpretation
0	constant	0	the smallest natural number
$succ(\cdot)$	function	1	the function that given a number returns its successor
$+(\cdot,\cdot)$	function	2	the function that given two numbers returns the number corresponding to the sum of the two
$*(\cdot,\cdot)$	function	2	the function that given two numbers returns the number corresponding to the product of the two
$<(\cdot,\cdot)$	relation	2	the less then relation between natural numbers
$\leq(\cdot,\cdot)$	relation	2	the less then or equal relation between natural numbers

Non logical symbols - Example

Example (Domain of strings)

symbols	**type**	**arity**	**intuitive interpretation**
ϵ	constant	0	The empty string
"a", "b",	constants	0	The strings containing one single character of the latin alphabet
$concat(\cdot, \cdot)$	function	2	the function that given two strings returns the string which is the concatenation of the two
$subst(\cdot, \cdot, \cdot)$	function	3	The function that replaces all the occurrence of a string with another string in a third one
$<$	relation	2	Alphabetic order on the strings
$substring(\cdot, \cdot)$	relation	2	a relation that states if a string is contained in another string

FOL Terms

Terms

- every constant c_i and every variable x_i is a term;
- if $t_1, \ldots, t_n$ are terms and f_i is a functional symbol of arity equal to n, then $f(t_1, \ldots, t_n)$ is a term

Ground terms

A term is ground if it does not contain individual variables.

- no constants $\Longrightarrow$ No ground terms
- no function symbols $\Longrightarrow$ ground terms $=$ constants
- at least one constant and one function symbol $\Longrightarrow$ infinite set of ground terms

FOL formulas

Definition (Atomic formula)

An atomic formula on a signature Σ is an expression of the form $p(t_1, \ldots, t_n)$ there p is an n-ary predicate of Σ and t_i are Σ terms. If we consider $=$, we have that $t_1 = t_2$ is also an atomic formula

Definition

Formulas

- an atomic formula is a formula;
- if A and B are formulas then $\bot$, $A \wedge B$, $A \rightarrow B$, $A \vee B$, $\neg A$, $A \equiv B$ are formulas
- if A is a formula and x a variable, then $\forall x.A$ and $\exists x.A$ are formulas.

Examples of terms and formulas

Example (Terms)

- x_i,
- c_i,
- $f_i(x_j, c_k)$, and
- $f(g(x, y), h(x, y, z), y)$

Example (formulas)

- $f(a, b) = c$,
- $P(c_1)$,
- $\exists x(A(x) \vee B(y))$, and
- $P(x) \to \exists y.Q(x, y)$.

An example of representation in FOL

Example (Language)

constants	functions (arity)	Predicate (arity)
Aldo	mark (2)	attend (2)
Bruno	best-friend (1)	friend (2)
Carlo		student (1)
MathLogic		course (1)
DataBase		less-than (2)
0, 1, . . . , 10		

Example (Terms)

Intuitive meaning	term
an individual named Aldo	Aldo
the mark 1	1
Bruno's best friend	best-friend(Bruno)
anything	x
Bruno's mark in MathLogic	mark(Bruno,MathLogic)
somebody's mark in DataBase	mark(x,DataBase)
Bruno's best friend mark in MathLogic	mark(best-friend(Bruno),MathLogic)

An example of representation in FOL (cont'd)

Example (Formulas)

Intuitive meaning	**Formula**
Aldo and Bruno are the same person	$Aldo = Bruno$
Carlo is a person and MathLogic is a course	$person(Carlo) \wedge course(MathLogic)$
Aldo attends MathLogic	$attend(Aldo, MathLogic)$
Courses are attended only by students	$\forall x(attend(x, y) \wedge course(y) \rightarrow student(x))$
every course is attended by somebody	$\forall x(course(x) \rightarrow \exists y\ attend(y, x))$
every student attends something	$\forall x(student(x) \rightarrow \exists y\ attend(x, y))$
There is a student who attends all the courses	$\exists x(student(x) \wedge \forall y(course(y) \rightarrow attend(x, y)))$
every course has at least two attenders	$\forall x(course(x) \rightarrow \exists y \exists z(attend(y, x) \wedge attend(z, x) \wedge \neg y = z))$
Aldo's best friend attend the same courses attended by Aldo	$\forall x(attend(Aldo, x) \rightarrow attend(best\text{-}friend(Aldo), x))$
best-friend is symmetric	$\forall x(best\text{-}friend(best\text{-}friend(x)) = x)$
Aldo and his best friend have the same mark in MathLogic	$mark(\text{best-friend}(Aldo), MathLogic) = mark(Aldo, MathLogic)$
A student can attend at most two courses	$\forall x \forall y \forall z \forall w(attend(x, y) \wedge attend(x, z) \wedge attend(x, w) \rightarrow (y = z \vee z = w \vee y = w))$

Common Mistakes

- Use of $\wedge$ with $\forall$

  $\forall x\ (WorksAt(FBK, x) \wedge Smart(x))$ means "Everyone works at FBK and everyone is smart"

  "Everyone working at FBK is smart" is formalized as
  $\forall x\ (WorksAt(FBK, x) \rightarrow Smart(x))$

- Use of $\rightarrow$ with $\exists$

  $\exists x\ (WorksAt(FBK, x) \rightarrow Smart(x))$ mans "There is a person so that if (s)he works at FBK then (s)he is smart" and this is true as soon as there is at last an x who does not work at FBK

  "There is an FBK-working smart person" is formalized as
  $\exists x\ (WorksAt(FBK, x) \wedge Smart(x))$

Representing variations quantifiers in FOL

Example

Represent the statement at least 2 students attend the KR course

$$\exists x_1 \exists x_2 (attend(x_1, KR) \wedge attend(x_2, KR))$$

The above representation is not enough, as x_1 and x_2 are variable and they could denote the same individual, we have to guarantee the fact that x_1 and x_2 denote different person. The correct formalization is:

$$\exists x_1 \exists x_2 (attend(x_1, KR) \wedge attend(x_2, KR) \wedge x_1 \neq x_2)$$

At least n ...

$$\exists x_1 \ldots x_n \left(\bigwedge_{i=1}^{n} \phi(x_i) \wedge \bigwedge_{i \neq j = 1}^{n} x_i \neq x_j \right)$$

Representing variations of quantifiers in FOL

Example

Represent the statement at most 2 students attend the KR course

$$\forall x_1 \forall x_2 \forall x_3 (attend(x_1, KR) \wedge attend(x_2, KR) \wedge attend(x_2, KR) \rightarrow x_1 = x_2 \vee x_2 = x_3 \vee x_1 = x_3)$$

At most n ...

$$\forall x_1 \dots x_{n+1} \left(\bigwedge_{i=1}^{n+1} \phi(x_i) \rightarrow \bigvee_{i \neq j = 1}^{n+1} x_i = x_j \right)$$

Free variables

Intuition

A free occurrence of a variable x is an occurrence of x which is not bounded by a (universal or existential) quantifier.

Definition (Free occurrence)

- any occurrence of x in t_k is free in $P(t_1, \ldots, t_k, \ldots, t_n)$
- any free occurrence of x in ϕ or in ψ is also fee in $\phi \wedge \psi$, $\psi \vee \phi$, $\psi \rightarrow \phi$, and $\neg\phi$
- any free occurrence of x in ϕ, is free in $\forall y.\phi$ and $\exists y.\phi$ if y is distinct from x.

Definition (Ground/Closed Formula)

A formula ϕ is ground if it does not contain any variable. A formula is closed if it does not contain free occurrences of variables.

Free variables

A variable x is free in ϕ (denote by $\phi(x)$) if there is at least a free occurrence of x in ϕ.

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- x is free in $friends(alice, x)$.
- x is free in $P(x) \rightarrow \forall x.Q(x)$ (the occurrence of x in red is free the one in green is not free.

Free variables - intuition

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free
- $\exists x.(Sum(x, 3) = 12)$ no free variables
- $\exists x.(Sum(x, y) = 12)$ y free

Free variable and free terms

Definition (Term free for a variable)

A term t is free for a variable x in formula ϕ, if and only if all the occurrences of x in ϕ do not occur within the scope of a quantifier of some variable occurring in t.

Example

The term x is free for y in $\exists z.hates(y, z)$. We can safely replace y with x obtaining $\exists z.hates(x, z)$ without changing the meaning of the formula. However, the term z is not free for y in $\exists z.hates(y, z)$. In fact y occurs within the scope of a quantifier of z. Thus, we cannot substitute z for y in this sentence without changing the meaning of the sentence as we obtain $\exists z.hates(z, z)$.

Free variables and free terms - example

An occurrence of a variable x can be safely instantiated by a term free for x in a formula ϕ,

If you replace x with a terms which is not free for x in ϕ, you can have unexpected effects:

E.g., replacing x with $\mathit{mother\text{-}of}(y)$ in the formula $\exists y.\mathit{friends}(x, y)$ you obtain the formula

$$\exists y.\mathit{friends}(\mathit{mother\text{-}of}(y), y)$$

Semantics of FOL

FOL interpretation

A first order interpretation for the signature $\Sigma = \langle c_1, c_2, \ldots, f_1, f_2, \ldots, P_1, P_2, \ldots \rangle$ is a pair $\langle \Delta, \mathcal{I} \rangle$ where

- Δ is a non empty set called interpretation domain
- $\mathcal{I}$ is is a function, called interpretation function
  - $\mathcal{I}(c_i) \in \Delta$ (elements of the domain)
  - $\mathcal{I}(f_i) : \Delta^n \rightarrow \Delta$ (n-ary function on the domain)
  - $\mathcal{I}(P_i) \subseteq \Delta^n$ (n-ary relation on the domain)

where n is the arity of f_i and P_i.

We use alternatively the notation $\mathcal{I}(\sigma)$ and $\sigma^{\mathcal{I}}$ to denot the interpretation of the symbol $\sigma \in \Sigma$.

Example of interpretation

Example (Of interpretation)

Symbols

Constants: *alice*, *bob*, *carol*, *robert*
Function: *mother-of* (with arity equal to 1)
Predicate: *friends* (with arity equal to 2)

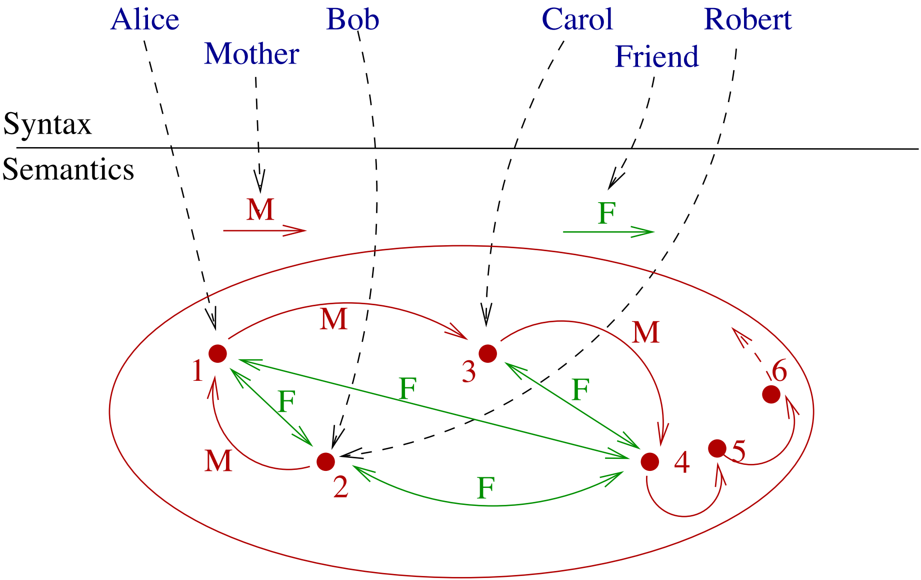
Domain $\Delta = \{1, 2, 3, 4, \dots\}$

Interpretation $\mathcal{I}(alice) = 1$, $\mathcal{I}(bob) = 2$, $\mathcal{I}(carol) = 3$, $\mathcal{I}(robert) = 2$

$$\mathcal{I}(\textit{mother-of}) = M \quad \begin{array}{l} M(1) = 3 \\ M(2) = 1 \\ M(3) = 4 \\ M(n) = n + 1 \text{ for } n \geq 4 \end{array}$$

$$\mathcal{I}(friends) = F = \left\{ \begin{array}{lll} \langle 1, 2 \rangle, & \langle 2, 1 \rangle, & \langle 3, 4 \rangle, \\ \langle 4, 3 \rangle, & \langle 4, 2 \rangle, & \langle 2, 4 \rangle, \\ \langle 4, 1 \rangle, & \langle 1, 4 \rangle, & \langle 4, 4 \rangle \end{array} \right\}$$

Example (cont'd)



Interpretation of terms

Definition (Assignment)

An assignment a is a function from the set of variables to Δ.

$a_{x \mapsto d}$ denotes the assignment that coincides with a on all the variables but x, which is associated to d.

Definition (Interpretation of terms)

The interpretation of a term t w.r.t. the assignment a, in symbols $\mathcal{I}(t)[a]$ is recursively defined as follows:

$$
\begin{aligned}
\mathcal{I}(x_i)[a] &= a(x_i) \\
\mathcal{I}(c_i)[a] &= \mathcal{I}(c_i) \\
\mathcal{I}(f(t_1, \ldots, t_n))[a] &= \mathcal{I}(f)(\mathcal{I}(t_1)[a], \ldots, \mathcal{I}(t_n)[a])
\end{aligned}
$$

FOL Satisfiability of formulas

Definition (Satisfiability of a formula w.r.t. an assignment)

An interpretation $\mathcal{I}$ satisfies a formula ϕ w.r.t. the assignment a according to the following rules:

$$
\begin{array}{rcl}
\mathcal{I} \models t_1 = t_2[a] & \text{iff} & \mathcal{I}(t_1)[a] \text{ is the same element as } \mathcal{I}(t_2)[a] \\
\mathcal{I} \models P(t_1, \dots, t_n)[a] & \text{iff} & \langle \mathcal{I}(t_1)[a], \dots, \mathcal{I}(t_n)[a] \rangle \in \mathcal{I}(P) \\
\mathcal{I} \models \phi \wedge \psi[a] & \text{iff} & \mathcal{I} \models \phi[a] \text{ and } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \phi \vee \psi[a] & \text{iff} & \mathcal{I} \models \phi[a] \text{ or } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \phi \rightarrow \psi[a] & \text{iff} & \mathcal{I} \not\models \phi[a] \text{ or } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \neg\phi[a] & \text{iff} & \mathcal{I} \not\models \phi[a] \\
\mathcal{I} \models \phi \equiv \psi[a] & \text{iff} & \mathcal{I} \models \phi[a] \text{ iff } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \exists x \phi[a] & \text{iff} & \text{there is a } d \in \Delta \text{ such that } \mathcal{I} \models \phi[a_{x \mapsto d}] \\
\mathcal{I} \models \forall x \phi[a] & \text{iff} & \text{for all } d \in \Delta, \mathcal{I} \models \phi[a_{x \mapsto d}]
\end{array}
$$

Example (cont'd)

Exercise 1:

Check the following statements, considering the interpretation $\mathcal{I}$ defined few slides ago:

1. $\mathcal{I} \models Alice = Bob[a]$
2. $\mathcal{I} \models Robert = Bob[a]$
3. $\mathcal{I} \models x = Bob[a_{x \mapsto 2}]$

Example (cont'd)

$$\mathcal{I}(\textit{mother-of}(\textit{alice}))[a] = 3$$

$$\mathcal{I}(\textit{mother-of}(x))[a_{x\mapsto 4}] = 5$$

$$\mathcal{I}(\textit{friends}(x,y)) =$$

$x :=$	$y :=$
1	2
2	1
4	1
1	4
4	2
2	4
4	3
3	4
4	4

$$\mathcal{I}(\textit{friends}(x,x)) =$$

$x :=$
4

$$\mathcal{I}(\textit{friends}(x,y) \wedge x = y) =$$

$x :=$	$y :=$
4	4

$$\mathcal{I}(\exists x \textit{friends}(x,y)) =$$

$y :=$
2
1
4
3

$$\mathcal{I}(\forall x \textit{friends}(x,y)) =$$

$y :=$
4

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ does not contains functional symbols (relational language), there is a strict analogy between first order logics and databases.

- Non logical simbols of $\mathcal{L}$ correspond to database schema (tables)
- Δ corresponds to the set of values which appears in the tables (active domain)
- the interpretation $\mathcal{I}$ corresponds to the tuples that belongs to each relation
- Formulas on $\mathcal{L}$ corresponds to query over the database
- Interpretation of formulas of $\mathcal{L}$ correspond to answers.

Analogy with Databases

FOL	DB
friends	`CREATE TABLE FRIENDS (friend1 : INTEGER friend2 : INTEGER)`
$friends(x, y)$	`SELECT * FROM FRIENDS`
$friends(x, x)$	`SELECT friend1 FROM FRIENDS WHERE friends1 = friends2`
$friends(x, y) \wedge x = y$	`SELECT * FROM FRIENDS WHERE friends1 = friends2`
$\exists x.friends(x, y)$	`SELECT friend2 FROM FRIENDS`
$friends(x, y) \wedge friends(y, z)$	`SELECT * FROM FRIENDS as FRIEND1 FRIENDS as FRIEND2 WHERE FRIENDS1.friends2 = FRIENDS2.friends1`

Satisfiability and Validity

Definition (Model, satisfiability and validity)

An interpretation $\mathcal{I}$ is a model of ϕ under the assignment a, if

$$\mathcal{I} \models \phi[a]$$

A formula ϕ is satisfiable if there is some $\mathcal{I}$ and some assignment a such that $\mathcal{I} \models \phi[a]$.
A formula ϕ is unsatisfiable if it is not satisfiable.
A formula ϕ is valid if every $\mathcal{I}$ and every assignment a $\mathcal{I} \models \phi[a]$

Definition (Logical Consequence)

A formula ϕ is a logical consequence of a set of formulas Γ, in symbols $\Gamma \models \phi$, if for all interpretations $\mathcal{I}$ and for all assignment a

$$\mathcal{I} \models \Gamma[a] \quad \Longrightarrow \quad \mathcal{I} \models \phi[a]$$

where $\mathcal{I} \models \Gamma[a]$ means that $\mathcal{I}$ satisfies all the formulas in Γ under a.

Excercises

Say where these formulas are valid, satisfiable, or unsatisfiable

- $\forall x P(x)$
- $\forall x P(x) \rightarrow \exists y P(y)$
- $\forall x. \forall y. (P(x) \rightarrow P(y))$
- $P(x) \rightarrow \exists y P(y)$
- $P(x) \vee \neg P(y)$
- $P(x) \wedge \neg P(y)$
- $P(x) \rightarrow \forall x. P(x)$
- $\forall x \exists y. Q(x, y) \rightarrow \exists y \forall x Q(x, y)$
- $x = x$
- $\forall x. P(x) \equiv \forall y. P(y)$
- $x = y \rightarrow \forall x. P(x) \equiv \forall y. P(y)$
- $x = y \rightarrow (P(x) \equiv P(y))$
- $P(x) \equiv P(y) \rightarrow x = y$

Solution

$\forall x P(x)$	Satisfiable
$\forall x P(x) \rightarrow \exists y P(y)$	Valid
$\forall x.\forall y.(P(x) \rightarrow P(y))$	Satisfiable
$P(x) \rightarrow \exists y P(y)$	Valid
$P(x) \vee \neg P(y)$	Satisfiable
$P(x) \wedge \neg P(y)$	Satisfiable
$P(x) \rightarrow \forall x.P(x)$	Satisfiable
$\forall x \exists y.Q(x,y) \rightarrow \exists y \forall x Q(x,y)$	Satisfiable
$x = x$	Valid
$\forall x.P(x) \equiv \forall y.P(y)$	Valid
$x = y \rightarrow \forall x.P(x) \equiv \forall y.P(y)$	Valid
$x = y \rightarrow (P(x) \equiv P(y))$	Valid
$P(x) \equiv P(y) \rightarrow x = y$	Satisfiable

Properties of quantifiers

Proposition

The following formulas are valid

- $\forall x(\phi(x) \wedge \psi(x)) \equiv \forall x\phi(x) \wedge \forall x\psi(x)$
- $\exists x(\phi(x) \vee \psi(x)) \equiv \exists x\phi(x) \vee \exists x\psi(x)$
- $\forall x\phi(x) \equiv \neg\exists x\neg\phi(x)$
- $\forall x\exists x\phi(x) \equiv \exists x\phi(x)$
- $\exists x\forall x\phi(x) \equiv \forall x\phi(x)$

Proposition

The following formulas are not valid

- $\forall x(\phi(x) \vee \psi(x)) \equiv \forall x\phi(x) \vee \forall x\psi(x)$
- $\exists x(\phi(x) \wedge \psi(x)) \equiv \exists x\phi(x) \wedge \exists x\psi(x)$
- $\forall x\phi(x) \equiv \exists x\phi(x)$
- $\forall x\exists y\phi(x, y) \equiv \exists y\forall x\phi(x, y)$

Expressing properties in FOL

What is the meaning of the following FOL formulas?

1. $\exists x(bought(Frank, x) \wedge dvd(x))$
2. $\exists x.bought(Frank, x)$
3. $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
4. $(\forall x.bought(Frank, x)) \rightarrow (\forall x.bought(Susan, x))$
5. $\forall x \exists y.bought(x, y)$
6. $\exists x \forall y.bought(x, y)$

1. "Frank bought a dvd."
2. "Frank bought something."
3. "Susan bought everything that Frank bought."
4. "If Frank bought everything, so did Susan."
5. "Everyone bought something."
6. "Someone bought everything."

Expressing properties in FOL

Define an appropriate language and formalize the following sentences using FOL formulas.

1. All Students are smart.
2. There exists a student.
3. There exists a smart student.
4. Every student loves some student.
5. Every student loves some other student.
6. There is a student who is loved by every other student.
7. Bill is a student.
8. Bill takes either Analysis or Geometry (but not both).
9. Bill takes Analysis and Geometry.
10. Bill doesn't take Analysis.
11. No students love Bill.

Expressing properties in FOL

1. $\forall x.(Student(x) \rightarrow Smart(x))$
2. $\exists x.Student(x)$
3. $\exists x.(Student(x) \wedge Smart(x))$
4. $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge Loves(x, y)))$
5. $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge \neg(x = y) \wedge Loves(x, y)))$
6. $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
7. $Student(Bill)$
8. $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$
9. $Takes(Bill, Analysis) \wedge Takes(Bill, Geometry)$
10. $\neg Takes(Bill, Analysis)$
11. $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$

Expressing properties in FOL

For each property write a formula expressing the property, and for each formula writhe the property it formalises.

- Every Man is Mortal
  $\forall x.Man(x) \rightarrow Mortal(x)$
- Every Dog has a Tail
  $\forall x.Dog(x) \rightarrow \exists y(PartOf(x,y) \wedge Tail(y))$
- There are two dogs
  $\exists x, y(Dog(x) \wedge Dog(y) \wedge x \neq y)$
- Not every dog is white
  $\neg\forall x.Dog(x) \rightarrow White(x)$
- $\exists x.Dog(x) \wedge \exists y.Dog(y)$
  There is a dog
- $\forall x, y(Dog(x) \wedge Dog(y) \rightarrow x = y)$
  There is at most one dog

Open and Closed Formulas

- Note that for closed formulas, satisfiability, validity and logical consequence do not depend on the assignment of variables.
- For closed formulas, we therefore omit the assignment and write $\mathcal{I} \models \phi$.
- More in general $\mathcal{I} \models \phi[a]$ if and only if $\mathcal{I} \models \phi[a']$ when $[a]$ and $[a']$ coincide on the variables free in ϕ (they can differ on all the others)

(un)satisfiability/validity of a FOL formula - examples

Example

Decide whether or not $\forall x(P(x) \rightarrow Q(x)) \rightarrow (\forall x P(x) \rightarrow \forall x Q(x))$ is valid.

- The above formula is valid when $\mathcal{I} \models \forall x(P(x) \rightarrow Q(x)) \rightarrow (\forall x P(x) \rightarrow \forall x Q(x))[a]$ for all assignment a. Which is equivalent to say that
- if $\mathcal{I} \models \forall x(P(x) \rightarrow Q(x))[a]$ then $\mathcal{I} \models (\forall x P(x) \rightarrow \forall x Q(x))[a]$; which is the same as:
- if $\mathcal{I} \models \forall x(P(x) \rightarrow Q(x))[a]$ and $\mathcal{I} \models \forall x P(x)[a]$ then $\mathcal{I} \models \forall x Q(x)[a]$.
- To show the previous fact, suppose that:
  (H1) $\mathcal{I} \models \forall x(P(x) \rightarrow Q(x))[a]$, and that
  (H2) $\mathcal{I} \models \forall x P(x)[a]$.
- From the hypothesis (H1), we have that for all $d \in \Delta^{\mathcal{I}}$, $\mathcal{I} \models P(x) \rightarrow Q(x)[a_{x \mapsto d}]$
- from the hypothesis (H2), we have that for all $d \in \Delta^{\mathcal{I}}$, $\mathcal{I} \models P(x)[a_{x \mapsto d}]$
- by the definition of satisfiability of implication we have that for all $d \in \Delta^{\mathcal{I}}$, $\mathcal{I} \models Q(x)[a_{x \mapsto d}]$
- which implies that $\mathcal{I} \models \forall Q(x)[a]$.

(un)satisfiability/validity of a FOL formula - examples

Example

Check if the formula $(\forall x P(x) \rightarrow \forall x Q(x)) \rightarrow \forall x(P(x) \rightarrow Q(x))$ is valid:

- This time we try to show that the formula is not valid.
- For this we have to find an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \forall x P(x) \rightarrow \forall x Q(x)[a]$ but $\mathcal{I} \not\models \forall x(P(x) \rightarrow Q(x))[a]$.
- in order to have that $\mathcal{I} \models \forall x P(x) \rightarrow \forall x Q(x)[a]$, we can choose to falsify the premise of the implication, i.e., to build an interpretation such that $\mathcal{I} \not\models \forall x P(x)[a]$.
- we need an element d in the domain of interpretation $\Delta^{\mathcal{I}}$, such that $\mathcal{I} \not\models P(x)[a_{x \mapsto d}]$.
- In order to have that $\mathcal{I} \not\models \forall x(P(x) \rightarrow Q(x))[a]$, we need an element d' of the domain $\Delta^{\mathcal{I}}$ such that $\mathcal{I} \models P(x)[a_{x \mapsto d'}]$ and $\mathcal{I} \not\models Q(x)[a_{x \mapsto d'}]$.
- at this point we can build the interpretation $\mathcal{I}$ on the domain $\Delta^{\mathcal{I}} = \{d, d'\}$ with $P^{\mathcal{I}} = \{d'\}$ and $Q^{\mathcal{I}} = \emptyset$.

Exercise 2:

Let $\mathcal{L}$ be a first order language on a signatore containing

- the constant symbols a and b,
- the binary function symbol f, and
- the binary predicate symbol P.

Answer to the following questions:

1. Does $\mathcal{L}$ have a finite model? If yes define it, if not explain why.
2. Let $\mathcal{T}$ be a theory containing the following axioms
    1. $\forall y.\neg P(x, x)$ (P is irreflexive)
    2. $\forall xyz.(P(x, y) \wedge P(y, z) \rightarrow P(x, z))$ (P is transitive)
    3. $\forall xy.(P(x, f(x, y)) \wedge P(y, f(x, y))$

    Is $\mathcal{T}$ satisfiable?. If yes can you provide a model for $\mathcal{T}$
3. Does $\mathcal{T}$ have a finite model? If yes, define it; if not, explain why.

Exercise 3:
Suppose that a first order language L contains only the set of constants $\{a, b, c\}$ and no functional symbols, and and the unary predicate symbol P. Say if the following formula is valid, i.e., true in all interpretations. If it is valid give a proof of it's validity; if it is not valid provide a counter-model.

$$P(a) \wedge P(b) \wedge P(c) \rightarrow \forall x P(x)$$

Exercise

Exercise 4:

Transform in FOL the following sentences:

1. The fathers of dogs are dogs.
2. There are at least two students enrolled in every course.
3. No region is part of each of two disjoint regions

Transform in Natural Language the following sentences:

1. $\forall x(Bag(x) \rightarrow \exists y(Coin(y) \wedge Contains(x, y)))$
2. $\exists x(Telephone(x) \wedge \forall y(Secretary(y) \rightarrow \neg Uses(x, y)))$
3. $\exists x(Buyer(x) \wedge Bought(x, TheScream) \wedge \forall y(Buyer(y) \wedge Bought(y, TheScream) \rightarrow x = y))$