First order logic

Syntax and semantics

Luciano Serafini – Maurizio Lenzerini

FBK-IRST, Trento, Italy – Sapienza Università di Roma

A.Y. 2024/25

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

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 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`

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.

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

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

Problem with previous solution

- `Mary-is-mortal` $\wedge$ `John-is-mortal` $\wedge$ `Chris-is-mortal` $\wedge \ldots \wedge$ `Michael-is-mortal`
- `Mary-is-a-spy` $\vee$`John-is-a-spy` $\vee$`Chris-is-a-spy` $\vee \ldots \vee$ `Michael-is-a-spy`

The representation is not compact and generalization patterns are difficult to express.

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 if we do not know all the people in our "universe"? How can we express the statement independently from the people in our "universe"?

Expressivity of propositional logic - III

Question

Try to express in Propositional Logic the following statements:

- Every natural number is either even or odd

Expressivity of propositional logic - III

Question

Try to express in Propositional Logic the following statements:

- Every natural number is either even or odd

A solution

We can use two families of propositions *even*$_i$ and *odd*$_i$ for every $i \geq 1$, and use the set of formulas

$$\{odd_i \vee even_i | i \geq 1\}$$

Problem with previous solution

$$\{odd_i \vee even_i | i \geq 1\}$$

What happens if we want to state this in one single formula? To do this we would need to write an infinite formula like:

$$(odd_1 \vee even_1) \wedge (odd_2 \vee even_2) \wedge \ldots$$

and this cannot be done in propositional logic.

Expressivity of propositional logic -IV

Question

Express the statements:

- the father of Luca is Italian

Solution (Partial)

- `mario-is-father-of-luca` $\rightarrow$ `mario-is-italian`
- `michele-is-father-of-luca` $\rightarrow$ `michele-is-italian`
- ...

Problem with previous solution

- `mario-is-father-of-luca` $\rightarrow$ `mario-is-italian`
- `michele-is-father-of-luca` $\rightarrow$ `michele-is-italian`
- ...

This statement strictly depend from a fixed set of people. What happens if we want to make this statement independently of the set of persons we have in our universe?

Why first order logic?

Because it provides a way of representing information like the following one:

1. Mary is a person;
2. John is a person;
3. Mary is mortal;
4. Mary and John are siblings
5. Every person is mortal;
6. There is a person who is a spy;
7. Every natural number is either even or odd;
8. The father of Luca is Italian

Why first order logic?

Because it provides a way of representing information like the following one:

1. Mary is a person;
2. John is a person;
3. Mary is mortal;
4. Mary and John are siblings
5. Every person is mortal;
6. There is a person who is a spy;
7. Every natural number is either even or odd;
8. The father of Luca is Italian

and also to infer the third one from the first one and the fifth one.

While propositional logic assumes that the world to be formalized is constituted by facts corresponding to propositions, first-order logic (like natural language) assumes that the world is constituted by:

- Individual objects, denoted by Constants: mary, john, 1, 2, 3, red, blue, world war 1, world war 2, 18th Century...
- Functional means to refer to objects, denoted by Functions: the father of, the best friend, the third inning of, ...
- Properties and relations, denoted by Predicates: Mortal, Prime, IsBrotherOf, Bigger than, Inside, IsPartOf, HasColor, Owns, ...

In the following, we define first-order logic with equality.

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 constants $\bot$ (false) and $\top$ (true)
- propositional logical connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\equiv$
- the quantifiers $\forall$, $\exists$
- a denumerable set of variable symbols $x_1, x_2, \ldots$
- the equality predicate symbol $=$ (optional)

Non logical symbols $\langle c_1, c_2, \ldots, f_1, f_2, \ldots, P_1, P_2, \ldots \rangle$

- a denumerable set $c_1, c_2, \ldots$ of constants
- a denumerable set $f_1, f_2, \ldots$ of function symbols each of which is associated with its arity ≥ 1 (i.e., number of arguments)
- a denumerable set $P_1, P_2, \ldots$ of predicate (or relational) symbols each of which is associated with its arity ≥ 0 (i.e., number of arguments)

Non logical symbols - Example

Non logical symbols depend on the domain we want to model. They should have an intended meaning in such a domain.

Example (Domain of arithmetics)

symbol	**type**	**arity**	**intended meaning**
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
$<(\cdot,\cdot)$	predicate	2	the "less than" 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	intended meaning
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
$<(\cdot,\cdot)$	predicate	2	the "less than" predicate between natural numbers
$\leq(\cdot,\cdot)$	predicate	2	the "less than or equal to" predicate between natural numbers

Non logical symbols - Example

Example (Domain of strings)

symbols	**type**	**arity**	**intended meaning**
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
$<$	predicate	2	alphabetic order on the strings
$substring(\cdot, \cdot)$	predicate	2	the relation that states if a string is contained in another string

Terms and formulas of FOL

In the following, we write g/n to indicate that the function or predicate symbol g has arity n.

Terms

- every constant c and every variable x is a term;
- if $t_1, \ldots, t_n$ are terms and f/n is a function symbol, then $f(t_1, \ldots, t_n)$ is a term.

Well formed formulas

- if t_1 and t_2 are terms then $t_1 = t_2$ is a formula;
- If $t_1, \ldots, t_n$ are terms and P/n is a predicate symbol, then $P(t_1, \ldots, t_n)$ is a formula (if $n = 0$, then the formula is written simply as P);
- if A and B are formulas then $\bot$, $\top$, $A \wedge B$, $A \rightarrow B$, $A \vee B$, $\neg A$, $A \equiv B$, (A) are formulas;
- if A is a formula and x a variable, then $\forall x.A$ and $\exists x.A$ are formulas (also written simply as $\forall x\, A$ and $\exists x\, A$).

Examples of terms and formulas

In the following examples we use the function symbols $f_1/2, f_2/3, g/2, h/3$, the variables x, y, z, the constants a, b, c and the predicate symbols $P/1, A/1, B/1, Q/2$.

Example (terms)

- x
- c
- $f_1(x, c)$
- $f_2(g(x, y), h(x, y, z), y)$

Example (formulas)

- $f_1(a, b) = c$
- $P(c)$
- $\exists x(A(x) \vee B(y))$
- $P(x) \rightarrow \exists y.Q(x, y)$

Semantics of FOL

FOL interpretation

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

- Δ is a non empty set called interpretation domain
- $\mathcal{I}$ is a function, called interpretation function such that
  - $\mathcal{I}(c_i) \in \Delta$ for each constant c_i
  - $\mathcal{I}(f_i) : \Delta^n \rightarrow \Delta$ for each function symbol f_i/n
  - $\mathcal{I}(P_i) \subseteq \Delta^n$ for each predicate symbol P_i/n

In other words, $\mathcal{I}$ associates to each constant an element of the domain Δ, to each function symbol f_i/n a total n-ary function on the domain Δ, and to each predicate symbol P_i an n-ary relation on the domain Δ (note that Δ^n denotes the cartesian product $\Delta \times \ldots \times \Delta$ n times).

In the following, we sometime use $c^{\mathcal{I}}$, $f^{\mathcal{I}}$ and $P^{\mathcal{I}}$ instead of $\mathcal{I}(c), \mathcal{I}(f)$ and $\mathcal{I}(P)$, respectively.

Example of interpretation

Example (of interpretation)

Symbols

Constants: *alice*, *bob*, *carol*, *robert*
Function symbol: *mother-of*/1
Predicate symbol: *friends*/2

Interpretation:

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

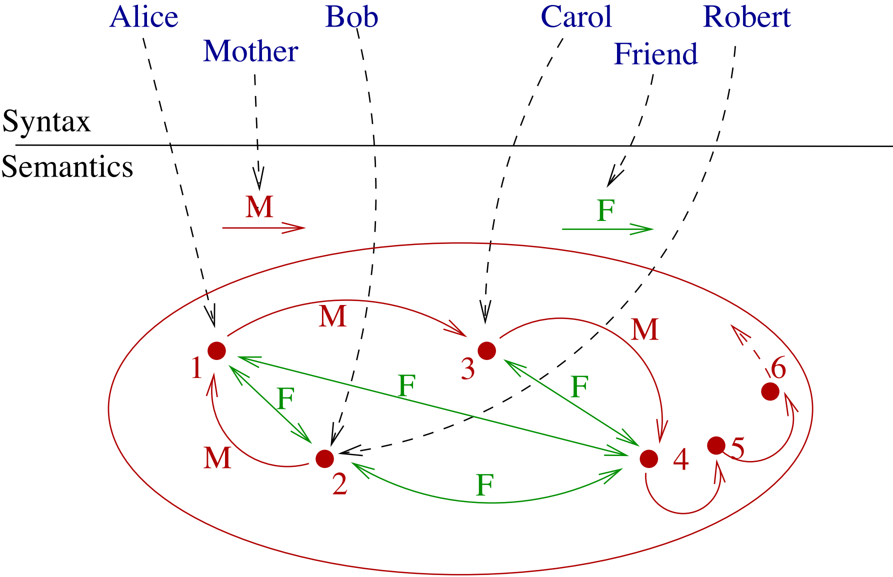
Interpretation function $\mathcal{I}$

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

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

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

Example (cont'd)



Interpretation of terms

Definition (Assignment)

An assignment α for $\mathcal{I}$ is a function from the set of variables to Δ.

If α is an assignment for $\mathcal{I}$, then $\alpha[x/d]$ denotes the assignment for $\mathcal{I}$ that coincides with α 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 α, in symbols $t^{\mathcal{I},\alpha}$ is recursively defined as follows (where x is a variable, c a constant and f/n a function symbol):

$$
\begin{aligned}
x^{\mathcal{I},\alpha} &= \alpha(x) \\
c^{\mathcal{I},\alpha} &= \mathcal{I}(c) \\
f(t_1, \ldots, t_n)^{\mathcal{I},\alpha} &= \mathcal{I}(f)(t_1^{\mathcal{I},\alpha}, \ldots, t_n^{\mathcal{I},\alpha})
\end{aligned}
$$

FOL Satisfaction of formulas

Definition (Satisfaction of a formula by $\mathcal{I}$ w.r.t. α)

The following rules establish when an interpretation $\mathcal{I}$ satisfies a formula ϕ w.r.t. the assignment α (written as $(\mathcal{I}, \alpha) \models \phi$):

$$
\begin{array}{rcl}
(\mathcal{I}, \alpha) \models \top & \text{and} & (\mathcal{I}, \alpha) \not\models \bot \\
(\mathcal{I}, \alpha) \models t_1 = t_2 & \text{if} & t_1^{\mathcal{I},\alpha} \text{ is the same as } t_2^{\mathcal{I},\alpha} \\
(\mathcal{I}, \alpha) \models P(t_1, \dots, t_n) & \text{if} & \langle t_1^{\mathcal{I},\alpha}, \dots, t_n^{\mathcal{I},\alpha} \rangle \in \mathcal{I}(P) \\
(\mathcal{I}, \alpha) \models \phi \wedge \psi & \text{if} & (\mathcal{I}, \alpha) \models \phi \text{ and } (\mathcal{I}, \alpha) \models \psi \\
(\mathcal{I}, \alpha) \models \phi \vee \psi & \text{if} & (\mathcal{I}, \alpha) \models \phi \text{ or } (\mathcal{I}, \alpha) \models \psi \\
(\mathcal{I}, \alpha) \models \phi \rightarrow \psi & \text{if} & (\mathcal{I}, \alpha) \not\models \phi \text{ or } (\mathcal{I}, \alpha) \models \psi \\
(\mathcal{I}, \alpha) \models \neg\phi & \text{if} & (\mathcal{I}, \alpha) \not\models \phi \\
(\mathcal{I}, \alpha) \models \phi \equiv \psi & \text{if} & (\mathcal{I}, \alpha) \models \phi \text{ if and only if } (\mathcal{I}, \alpha) \models \psi \\
(\mathcal{I}, \alpha) \models (\phi) & \text{if} & (\mathcal{I}, \alpha) \models \phi \\
(\mathcal{I}, \alpha) \models \exists x \phi & \text{if} & (\mathcal{I}, \alpha[x/d]) \models \phi \text{ for some } d \in \Delta \\
(\mathcal{I}, \alpha) \models \forall x \phi & \text{if} & (\mathcal{I}, \alpha[x/d]) \models \phi \text{ for all } d \in \Delta
\end{array}
$$

Exercise

Let α be such that $\alpha(x) = 2$, and let $\mathcal{I}$ be the interpretation function defined few slides ago on the domain $\Delta = \{1, 2, 3, 4, \dots\}$. Check whether the following are true:

1. $(\mathcal{I}, \alpha) \models \mathit{Alice} = \mathit{Bob}$
2. $(\mathcal{I}, \alpha) \models \mathit{Robert} = \mathit{Bob} \wedge \mathit{friends}(\mathit{Carol}, \mathit{mother\text{-}of}(\mathit{Carol}))$
3. $(\mathcal{I}, \alpha[x/2]) \models \neg(x = \mathit{Bob})$
4. $(\mathcal{I}, \alpha) \models \exists x \, \neg \mathit{friends}(x, \mathit{mother\text{-}of}(x))$
5. $(\mathcal{I}, \alpha) \models \forall x \, \exists y \, \mathit{friends}(x, y)$
6. $(\mathcal{I}, \alpha) \models \forall x \, \exists y \, \neg \mathit{friends}(x, y)$
7. $(\mathcal{I}, \alpha) \models \forall x \, \forall y \, \mathit{friends}(x, y) \rightarrow \mathit{friends}(y, x)$

Free variables

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

Definition (Inductive definition of free occurrence)

- any occurrence of x in t_k is free in $P(t_1, \dots, t_k, \dots, t_n)$
- any free occurrence of x in ϕ or in ψ is also free in $\phi \wedge \psi$, $\psi \vee \phi$, $\psi \rightarrow \phi$, $\psi \equiv \phi$ 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.

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

Definition (Free variable)

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 - examples

Intuitively..

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

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\, (friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$ no free variables

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ $\quad x$ free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ $\quad x$ not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ $\quad x$ free
- $\forall y. friends(Bob, y)$ $\quad$ no free variables
- $sum(x, 3) = 12$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\, (friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$ no free variables
- $sum(x, 3) = 12$ x free

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$ no free variables
- $sum(x, 3) = 12$ x free
- $\exists x.(sum(x, 3) = 12)$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$ no free variables
- $sum(x, 3) = 12$ x free
- $\exists x.(sum(x, 3) = 12)$ no free variables

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\, (friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x free
- $\forall y.friends(Bob, y)$ no free variables
- $sum(x, 3) = 12$ x free
- $\exists x.(sum(x, 3) = 12)$ no free variables
- $\exists x.(sum(x, y) = 12)$

Free variables - examples

Intuitively..

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

- $friends(Alice, x)$ x free
- $\forall x\,(friends(Alice, x) \wedge friends(Bob, x))$ x not free
- $(\forall x\, friends(Alice, x)) \wedge friends(Bob, x)$ x 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

Open and closed formulas

Definition (Ground/Closed/Open Formula)

A formula ϕ is ground if it does not contain any variable.
A formula is open if it contains at least one free variable, closed otherwise.
Obviously, all ground formulas are closed.

It is not difficult to see that when we interpret a closed formula in an interpretation, we do not actually need any assignment. Indeed, it can be shown that if α_1 and α_2 coincide on the variables that are free in ϕ, then $(\mathcal{I}, \alpha_1) \models \phi$ if and only if $(\mathcal{I}, \alpha_2) \models \phi$. For this reason, in the following, whenever ϕ is a closed formula, we write

$$\mathcal{I} \models \phi$$

to mean that $\mathcal{I}$ satisfies ϕ (independently from any assignment).

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
Math		course/1
DataBase		less-than/2
0, 1, . . . , 10		

Example (Terms)

Intended meaning
an individual named Aldo
the mark 1
Bruno's best friend
anything
Bruno's mark in Math
somebody's mark in DataBase
Bruno's best friend mark in Math

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
Math		course/1
DataBase		less-than/2
0, 1, . . . , 10		

Example (Terms)

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

An example of representation in FOL (cont'd)

Example (Formulas)

Intended meaning
Aldo and Bruno are the same person
Carlo is a person and Math is a course
Aldo attends Math
Courses are attended only by students
every course is attended by somebody
every student attends something
there is a student who attends all the courses
every course has at least two attenders
Aldo's best friend attend all the courses attended by Aldo
Aldo and his best friend have the same mark in Math
A student can attend at most two courses

An example of representation in FOL (cont'd)

Example (Formulas)

Intended meaning	**Formula**
Aldo and Bruno are the same person	$Aldo = Bruno$
Carlo is a person and Math is a course	$person(Carlo) \wedge course(Math)$
Aldo attends Math	$attend(Aldo, Math)$
Courses are attended only by students	$\forall x \forall y(attend(x, y) \rightarrow course(y) \rightarrow student(x))$
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 all the courses attended by Aldo	$\forall x(attend(Aldo, x) \rightarrow attend(best\text{-}friend(Aldo), x))$
Aldo and his best friend have the same mark in Math	$mark(best\text{-}friend(Aldo), Math) = mark(Aldo, Math)$
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 $\forall$

$$\forall x \; (WorksAt(FBK, x) \wedge Smart(x))$$

Common Mistakes

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

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

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 $\forall x\ (WorksAt(FBK, x) \rightarrow Smart(x))$

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 $\exists$

  $\exists x\ (WorksAt(FBK, x) \rightarrow Smart(x))$

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

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"

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

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 ...

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

Example

Represent the statement "at most 2 students attend the KR course"

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

At most n ...

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

Models, satisfiability and validity

In the following, if Γ is a set of formulas, then $(\mathcal{I}, \alpha) \models \Gamma$ means that $(\mathcal{I}, \alpha) \models g$ for all $g \in \Gamma$.

Definition (Model, satisfiability and validity)

An interpretation $\mathcal{I}$ is a model of ϕ under the assignment α, if $(\mathcal{I}, \alpha) \models \phi$.
A formula ϕ is satisfiable if there is some $\mathcal{I}$ and some assignment α such that $(\mathcal{I}, \alpha) \models \phi$.
A formula ϕ is unsatisfiable if it is not satisfiable.
A formula ϕ is valid if for every $\mathcal{I}$ and every assignment α, we have $(\mathcal{I}, \alpha) \models \phi$.

Definition (Logical implication)

A formula ϕ is logical implied by a set of formulas Γ, in symbols $\Gamma \models \phi$, if for all interpretations $\mathcal{I}$ and for all assignment α, if $(\mathcal{I}, \alpha) \models \Gamma$ then $(\mathcal{I}, \alpha) \models \phi$.

Definition (Model, satisfiability and validity for closed formulas)

An interpretation $\mathcal{I}$ is a model of a closed formula ϕ if $\mathcal{I} \models \phi$.
A closed formula ϕ is satisfiable if there is some $\mathcal{I}$ such that $\mathcal{I} \models \phi$.
A clsed formula ϕ is unsatisfiable if it is not satisfiable.
A closed formula ϕ is valid if for every $\mathcal{I}$ we have $\mathcal{I} \models \phi$.

Definition (Logical implication)

A closed formula ϕ is logically implied by a set of closed formulas Γ, in symbols $\Gamma \models \phi$, if for all interpretations $\mathcal{I}$, if $\mathcal{I} \models \Gamma$ then $\mathcal{I} \models \phi$.

The notions of theory, set of axioms of a theory, and finitely axiomatizable theory in first-order logic is a straightforward generalization of the corresponding notions in propositional logic.

Properties of first-order logical implication

Proposition

If Γ and Σ are two sets of closed formulas formulas, and A and B two closed formulas, then the following properties hold:

- **Reflexivity** *If $A \in \Gamma$, then $\Gamma \models A$*
- **Ex falso sequitur quodlibet** *If Γ is unsatisfiable, then $\Gamma \models A$ for all formulas A*
- **Monotonicity** *If $\Gamma \models A$ then $\Gamma \cup \Sigma \models A$*
- **Cut** *If $\Gamma \models A$ and $\Sigma \cup \{A\} \models B$ then $\Gamma \cup \Sigma \models B$*
- **Compactness** *If $\Gamma \models A$, then there is a finite subset $\Gamma_0 \subseteq \Gamma$, such that $\Gamma_0 \models A$*
- **Deduction theorem** *If $\Gamma \cup \{A\} \models B$ then $\Gamma \models A \to B$*
- **Deduction principle** *$\Gamma \cup \{A\} \models B$ if and only if $\Gamma \models A \to B$*
- **Refutation principle** *$\Gamma \models A$ if and only if $\Gamma \cup \{\neg A\}$ is unsatisfiable*

Computational properties of first-order logic

Theorem

Checking if a formula in first-order logic is valid is an undecidable problem (although semi-decidable: there is algorithm that terminates and answers "yes" if the formula is valid). The same holds for checking satisfiability and checking logical implication.

The proof of undecidability is by reduction from the halting problem, that is the problem of checking if a given Turing machine halts for a given input. The basic idea is that the Turing maching and the input can be formalized by a first-order logic that is valid if and only if the machine halts for the input.

What about the evaluation of a formula in an interpretation? Note that this question is meaningful only if the interpretation can be represented as a finite structure. Actually, it can be shown that evaluating a formula in a finite interpretation is a PSPACE-complete problem.

Excercises

Tell whether 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. $\exists x \forall y\ bought(x, y)$

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 write the property it formalises.

- Every Man is Mortal
- Every Dog has a Tail
- There are two dogs
- Not every dog is white
- $\exists x.Dog(x) \wedge \exists y.Dog(y)$
- $\forall x \forall y (Dog(x) \wedge Dog(y) \rightarrow x = y)$

Expressing properties in FOL

For each property write a formula expressing the property, and for each formula write 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
  $\exists x \exists y(Dog(x) \wedge Dog(y) \wedge x \neq y)$
- 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
- $\forall x \forall y(Dog(x) \wedge Dog(y) \rightarrow x = y)$
  There is at most one dog