UCS1524 – Logic Programming

Foundations of First Order Logic

Session Meta Data

Author	**Dr. D. Thenmozhi**
Reviewer	
Version Number	1.2
Release Date	20 July 2022

Session Objectives

- Understanding the concept of predicate logic of first order logic (FOL)
- Learning formal representation of FOL, validity, satisfiability and semantic equivalence in PL

Session Outcomes

- At the end of this session, participants will be able to
  - explain the formal representation of statements in FOL and inference in FOL.
  - apply inferencing in FOL

Agenda

- Syntax and Semantics of FOL
- Validity and satisfiability
- Inference rules in FOL
- Semantic equivalence

Pros and cons of propositional logic

- Propositional logic is declarative
- Propositional logic allows partial/disjunctive/negated information
  - (unlike most data structures and databases)
- Propositional logic is compositional:
  - meaning of B1,1 ∧ P1,2 is derived from meaning of B1,1 and of P1,2

- Meaning in propositional logic is context-independent – (unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power
  - (unlike natural language)
  - E.g., cannot say "*“Every elephant is gray”:*
  - “ • except by writing one sentence for each elephant
  - In FOL, can be written as ∀ x (elephant(x) → gray(x))

First-order logic (FOL)

- First-order logic (FOL) models the world in terms of
  - **Objects,** which are things with individual identities
  - **Properties** of objects that distinguish them from other objects
  - **Relations** that hold among sets of objects
  - **Functions,** which are a subset of relations where there is only one "value" for any given "input"
- Examples:
  - Objects: Students, lectures, companies, cars ...
  - Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: father-of, best-friend, second-half, one-more-than ...

Syntax of predicate logic

Variables are expressions of the form x_i with $i = 1, 2, 3 \ldots$.
Predicate symbols are expressions of the form P_i^k, where $i = 1, 2, 3 \ldots$ and $k = 0, 1, 2 \ldots$.
Function symbols are expressions of the form f_i^k, where $i = 1, 2, 3 \ldots$ and $k = 0, 1, 2 \ldots$.
We call i the (identification) index and k the arity of the symbol.
Terms are inductively defined as follows:

(1) Variables are terms.

(2) Function symbols of arity 0 are terms.

(3) If f is a function symbol with arity $k \geq 1$ and $t_1, \ldots, t_k$ are terms then $f(t_1, \ldots, t_k)$ is a term.

Function symbols of arity 0 are called constants.

Syntax of predicate logic

- **Constant symbols,** which represent individuals in the world
  - Mary
  - 3
  - Green
- **Function symbols,** which map individuals to individuals
  - father-of(Mary) = John
  - color-of(Sky) = Blue
- **Predicate symbols,** which map individuals to truth values
  - greater(5,3)
  - green(Grass)
  - color(Grass, Green)

Syntax of predicate logic

Formulas (of predicate logic) are inductively defined as follows:.

Well formed formulas (wff) are called as formulas

(1) Predicate symbols of arity 0 are formulas.

(2) If P is a predicate symbol of arity $k \geq 1$ and $t_1, \ldots, t_k$ are terms then $P(t_1, \ldots, t_k)$ is a formula.

(3) If F is a formula, then $\neg F$ is also a formula.

(4) If F and G are formulas, then $(F \wedge G)$ and $(F \vee G)$ are also formulas.

(5) If x is a variable and F is a formula, then $\exists x\ F$ and $\forall x\ F$ are also formulas. The symbols $\exists$ and $\forall$ are called the existential and the universal quantifier, respectively.

Formulas of the form P for some predicate symbol of arity 0 or of the form $P(t_1, \ldots, t_k)$ are called atomic formulas. The syntax tree and the subformulas of a formula are defined as usual.

A BNF for predicate logic

```
S := <Sentence> ;
<Sentence> := <AtomicSentence> |
          <Sentence> <Connective> <Sentence> |
          <Quantifier> <Variable>,... <Sentence> |
          "NOT" <Sentence> |
          "(" <Sentence> ")";
<AtomicSentence> := <Predicate> "(" <Term>, ... ")" |
                    <Term> "=" <Term>;
<Term> := <Function> "(" <Term>, ... ")" |
          <Constant> |
          <Variable>;
<Connective> := "AND" | "OR" | "IMPLIES" | "EQUIVALENT";
<Quantifier> := "EXISTS" | "FORALL" ;
<Constant> := "A" | "X1" | "John" | ... ;
<Variable> := "a" | "x" | "s" | ... ;
<Predicate> := "Before" | "HasColor" | "Raining" | ... ;
<Function> := "Mother" | "LeftLegOf" | ... ;
```

Quantifiers

- **Universal quantification**
  - $(\forall x)P(x)$ means that P holds for **all** values of x in the domain associated with that variable
  - E.g., $(\forall x)$ dolphin(x) $\rightarrow$ mammal(x)
- **Existential quantification**
  - $(\exists\, x)P(x)$ means that P holds for **some** value of x in the domain associated with that variable
  - E.g., $(\exists\, x)$ mammal(x) $\wedge$ lays-eggs(x)
  - Permits one to make a statement about some object without naming it

Quantifiers

- Universal quantifiers are often used with "implies" to form "rules":

  $(\forall x)$ student(x) $\rightarrow$ smart(x) means "All students are smart"

- Universal quantification is *rarely* used to make blanket statements about every individual in the world:

  $(\forall x)$student(x)$\wedge$smart(x) means "Everyone in the world is a student and is smart"

- Existential quantifiers are usually used with "and" to specify a list of properties about an individual:

  $(\exists x)$ student(x) $\wedge$ smart(x) means "There is a student who is smart"

- A common mistake is to represent this English sentence as the FOL sentence:

  $(\exists x)$ student(x) $\rightarrow$ smart(x)

  – But what happens when there is a person who is *not* a student?

Quantifier Scope

- Switching the order of universal quantifiers *does not* change the meaning:
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x)\ P(x,y)$
- Similarly, you can switch the order of existential quantifiers:
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x)\ P(x,y)$
- Switching the order of universals and existentials *does* change meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

Connections between All and Exists

We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:

$$(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$$

$$\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$$

$$(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$$

$$(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$$

Translating English to FOL

Every gardener likes the sun.

$\forall x$ gardener(x) $\rightarrow$ likes(x,Sun)

You can fool some of the people all of the time.

$\exists x\ \forall t$ person(x) $\wedge$time(t) $\rightarrow$ can-fool(x,t)

You can fool all of the people some of the time.

$\forall x\ \exists t$ (person(x) $\rightarrow$ time(t) $\wedge$can-fool(x,t))

$\forall x$ (person(x) $\rightarrow$ $\exists t$ (time(t) $\wedge$can-fool(x,t)))

Equivalent

There are exactly two purple mushrooms.

$\exists x\ \exists y$ mushroom(x) $\wedge$ purple(x) $\wedge$ mushroom(y) $\wedge$ purple(y) ^ $\neg$(x=y) $\wedge$ $\forall z$ (mushroom(z) $\wedge$ purple(z)) $\rightarrow$ ((x=z) $\vee$ (y=z))

Clinton is not tall.

$\neg$tall(Clinton)

Free and bounded variables, closed formulas

A variable x occurs in a formula F if it appears in some term of F.

An occurrence of a variable in a formula is either free or bounded.

An occurrence of x in F is bounded if it belongs to some subformula of F of the form $\exists x G$ or $\forall x G$; the smallest such subformula is the scope of the occurrence. Otherwise the occurrence is free.

A formula without any free occurrence of any variable is closed.

The matrix of a formula F is the formula obtained by removing from F every occurrence of the quantifiers $\exists$ and $\forall$, together with the (occurrence of a) variable following them. The matrix of F is denoted by F^*.

We introduce the usual abbreviations $(F \to G)$ and $(F \leftrightarrow G)$ as in propositional logic.

Free and bounded variables, closed formulas

Example: $F = (\exists x_1 P_5^2(x_1, f_2^1(x_2)) \vee \neg\forall x_2 P_4^2(x_2, f_7^2(f_4^0, f_5^1(x_3))))$ is a formula. All the subformulas of F are:

F

$\exists x_1 P_5^2(x_1, f_2^1(x_2))$

$P_5^2(x_1, f_2^1(x_2))$

$\neg\forall x_2 P_4^2(x_2, f_7^2(f_4^0, f_5^1(x_3)))$

$\forall x_2 P_4^2(x_2, f_7^2(f_4^0, f_5^1(x_3)))$

$P_4^2(x_2, f_7^2(f_4^0, f_5^1(x_3)))$

All the terms that occur in F are:

x_1

x_2

$f_2^1(x_2)$

$f_7^2(f_4^0, f_5^1(x_3))$

f_4^0

$f_5^1(x_3)$

x_3

All occurences of x_1 in F are bound. The first occurence of x_2 is free, all others are bound. Further, x_3 occurs free in F. Hence, the formula F is not closed. The term f_4^0 is an example for a constant. The matrix of F is the formula

$$F^* = (P_5^2(x_1, f_2^1(x_2)) \vee \neg P_4^2(x_2, f_7^2(f_4^0, f_5^1(x_3))))$$

Semantics of predicate logic: structures

A structure is a pair $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$, where $U_{\mathcal{A}}$ is an arbitrary, nonempty set called the ground set or universe of $\mathcal{A}$, and $I_{\mathcal{A}}$ is a partial function that maps

- predicate symbols of arity $k \geq 1$ to predicates over $U_{\mathcal{A}}$ of arity k (i.e., to functions of type $U_{\mathcal{A}}^k \to \{0,1\}$ or, equivalently, to subsets of $U_{\mathcal{A}}^k$),
- predicate symbols of arity 0 to either 0 or 1
- function symbols of arity $k \geq 1$ to functions over $U_{\mathcal{A}}$ of arity k (i.e., to functions of type $U_{\mathcal{A}}^k \to U_{\mathcal{A}}$),
- constants f of arity 0 to elements of the universe $U_{\mathcal{A}}$, and
- variables x to elements of the universe $U_{\mathcal{A}}$.

Structures - Example

Example: $F = \forall x P(x, f(x)) \wedge Q(g(a, z))$ is a formula. Here, P is a binary and Q a unary predicate, f is unary, g a binary, and a a 0-ary function symbol. The variable z is free in F. An example for a structure $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ which is suitable for F is the following.

$$
\begin{aligned}
U_\mathcal{A} &= \{0,1,2,3,\ldots\} = \mathbb{N}, \\
I_\mathcal{A}(P) &= P^\mathcal{A} = \{(m,n) \mid m,n \in U_\mathcal{A} \text{ and } m < n\}, \\
I_\mathcal{A}(Q) &= Q^\mathcal{A} := \{n \in U_\mathcal{A} \mid n \text{ is prime }\} \\
I_\mathcal{A}(f) &= f^\mathcal{A} = \text{ the successor function on } U_\mathcal{A}, \\
&\quad \text{hence } f^\mathcal{A}(n) = n+1, \\
I_\mathcal{A}(g) &= g^\mathcal{A} = \text{ the addition function on } U_\mathcal{A}, \\
&\quad \text{hence } g^\mathcal{A}(m,n) = m+n, \\
I_\mathcal{A}(a) &= a^\mathcal{A} = 2, \\
I_\mathcal{A}(z) &= z^\mathcal{A} = 3.
\end{aligned}
$$

In this structure F is obviously "true" because every natural number is smaller than its successor, and the sum of 2 and 3 is a prime number.

Structures

Let F be a formula and let $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$ be a structure.
$\mathcal{A}$ is suitable for F if all predicate and function symbols occurring in F and all variables occurring free in F belong to the domain of $I_{\mathcal{A}}$.

Let F be a formula and let $\mathcal{A}$ be a structure suitable for F. For every term t that can be constructed from variables and function symbols that appear in F, we define the value of t in the structure $\mathcal{A}$, denoted by $\mathcal{A}(t)$. The definition is inductive:

(1) If $t = x$ for some variable x, then $\mathcal{A}(t) = x^{\mathcal{A}}$.

(2) If $t = f(t_1, \ldots, t_k)$ for some function symbol f of arity k and terms $t_1, \ldots, t_k$, then $\mathcal{A}(t) = f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$.

(3) If $t = a$ for some constant a, then $\mathcal{A}(t) = a^{\mathcal{A}}$.

Evaluation of a term in a structure

We define inductively the (truth-)value of a formula F in the structure A, denoted by A(F):

- If $F = P(t_1, \ldots, t_k)$ for some predicate symbol P of arity k and terms $t_1, \ldots, t_k$ then

$$\mathcal{A}(F) = \begin{cases} 1 & \text{if } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in P^{\mathcal{A}} \\ 0 & \text{otherwise} \end{cases}$$

- If $F = \neg G$ for some formula G then

$$\mathcal{A}(F) = \begin{cases} 1 & \text{if } \mathcal{A}(G) = 0 \\ 0 & \text{otherwise} \end{cases}$$

- If $F = (G \wedge H)$ for some formulas G and H then

$$\mathcal{A}(F) = \begin{cases} 1 & \text{if } \mathcal{A}(G) = 1 \text{ and } \mathcal{A}(H) = 1 \\ 0 & \text{otherwise} \end{cases}$$

- If $F = (G \vee H)$ for some formulas G and H then

$$\mathcal{A}(F) = \begin{cases} 1 & \text{if } \mathcal{A}(G) = 1 \text{ or } \mathcal{A}(H) = 1 \\ 0 & \text{otherwise} \end{cases}$$

- If $F = \forall x\ G$ for some formula G and variable x then

$$\mathcal{A}(F) = \begin{cases} 1 & \text{if for every } d \in U_{\mathcal{A}}: \ \mathcal{A}_{[x/d]}(G) = 1 \\ 0 & \text{otherwise} \end{cases}$$

- If $F = \exists x\ G$ for some formula G and variable x then

$$\mathcal{A}(F) = \begin{cases} 1 \text{ if there exists } d \in U_{\mathcal{A}} \text{ such that: } \mathcal{A}_{[x/d]}(G) = 1 \\ 0 \ \ \text{otherwise} \end{cases}$$

where $\mathcal{A}_{[x/d]}$ denotes the structure $\mathcal{A}'$ that coincides with $\mathcal{A}$ everywhere, but (possibly) in the definition of $x^{\mathcal{A}'}$: it holds $x^{\mathcal{A}'} = d$, whether x belongs to the domain of $I_{\mathcal{A}}$ or not.

Model, validity, satisfiability

We write $\mathcal{A} \models F$ to denote that the structure $\mathcal{A}$ is suitable for the formula F and $\mathcal{A}(F) = 1$ holds. We say that F holds in $\mathcal{A}$ or that $\mathcal{A}$ is a model of F.

If every structure suitable for F is a model of F, then we write $\models F$ and say that F is valid.

If F has at least one model then we say that F is satisfiable.

	Propositional Wffs	**Predicate Wffs**
Truth Values	True or false – depends on the truth value of statement letters	True, false or neither(if the wff has a free variable)
Intrinsic truth	Tautology- true for all truth values of its statements	Valid wff- true for all interpretations
Methodology	Truth table to determine if it is a tautology	No algorithm to determine validity

An expression A is valid if A is true for all interpretations

Model, validity, satisfiability

$(\forall x)P(x) \to (\exists x)P(x)$

- This is valid because if every object of the domain has a certain property, then there exists an object of the domain that has the same property. Therefore, whenever the antecedent is true, so is the consequent, and the implication is therefore true.

$(\forall x)P(x) \to P(a)$

- Valid – quite obvious since *a* is a member of the domain of *X*.

$(\exists x)P(x) \to (\forall x)P(x)$

- Not valid since the property cannot be valid for all objects in the domain if it is valid for some objects of than domain. Can use a mathematical context to check as well. Say $P(x)$ = "x is even", then there exists an integer that is even but not every integer is even.

How about $(\forall x)[P(x) \vee Q(x)] \to (\forall x)P(x) \vee (\forall x)Q(x)$

- Invalid, can prove by mathematical context by taking $P(x) = x$ is even, $Q(x) = x$ is odd. In that case, the hypothesis is true but not the conclusion is false because it is not the case that every integer is even or that every integer is odd.

Example - 1

An expression A is valid if A is true for all interpretations

- Let L = {P,Q; a, b} be a predicate language where the both predicate symbols are binary predicates. Let A = Z_+ (the set of positive integers).
  - We can choose an interpretation to L, for example, such that V (a) := 3, V (b) := 2, V (P(x, y)) := x < y, and V (Q(x, y)) := x >= y.
- Determine the truth value of a formula P(c, b) -> ¬Q(c, a) in the model M.
- We have the interpretation 3 < 2 -> ¬ 2 >= 3 for the formula, and hence it is true in M.

Example - 2

Show that the statement

$$\forall x \, \exists y \, Q(x,y) \rightarrow \exists y \, \forall x \, Q(x,y)$$

is not valid.

Consider the formula in the model $\mathcal{M} = (\mathbb{N}, V)$ ($\mathbb{N} = \{1, 2, \ldots\}$ is the set of natural numbers) where $V(Q) = \{(a,b) \in \mathbb{N} \times \mathbb{N} \mid a < b\}$. Hence, $\mathcal{M} \models \forall x \, \exists y \, Q(x,y)$, because for all natural number, there exists a natural number b, such that $\mathcal{M} \models Q(a,b)$, i.e. $a < b$. On the other hand, $\mathcal{M} \not\models \exists y \, \forall x \, Q(x,y)$ because there does not exist a natural number b, such that $\mathcal{M} \models Q(a,b)$ for all natural numbers a, because the set $\mathbb{N}$ does not have the greatest element. Hence, $\mathcal{M}$ is such a model where our statement is not true. From this it follows that the statement is not valid.

Example - 3

Show that the formula

$$\exists\, x\ (P(x) \wedge \neg Q(x))$$

is satisfiable. Consider the model $\mathcal{M} = (\mathbb{Z}, V)$, such that

$$V(P(x)) = \{x \mid x > 2\} \quad \text{and} \quad V(Q(x)) = \{x \mid x < 10\}.$$

$\mathcal{M} \models \exists\, x\ (P(x) \wedge \neg Q(x))$ iff $\mathcal{M} \models P(a) \wedge \neg Q(a)$ for some $a \in \mathbb{Z}$. This is the case iff $\mathcal{M} \models P(a)$ and $\mathcal{M} \models \neg Q(a)$, i.e. $\mathcal{M} \not\models Q(a)$. For example, the value $a = 15$ satisfies the last condition, because $15 \in V(P)$ (i.e. $15 > 2$) and $15 \notin V(Q)$ (i.e. $15 \not< 10$). Hence, the formula is satisfiable.

Remarks

- F is valid if and only if ¬F is unsatisfiable.
- If all predicate symbols are required to have arity 0, essentially we get the formulas in propositional logic where the predicates play the role of the atomic formulas *in propositional* logic.

$$F = (Q(a) \vee \neg R(f(b), c)) \wedge P(a, b)$$

$$\begin{array}{rcl} Q(a) & \longleftrightarrow & A_1 \\ R(f(b), c) & \longleftrightarrow & A_2 \\ P(a, b) & \longleftrightarrow & A_3 \end{array}$$

$$F' = (A_1 \vee \neg A_2) \wedge A_3$$

- Observe that a formula without occurrences of a quantifier (e.g. The matrix of a given formula) can be transformed into an equivalent formula in CNF or DNF where only the tools from propositional logic are needed.

Quantified inference rules

- Universal instantiation
  - $\forall x\ P(x)\ \therefore\ P(A)$
- Universal generalization
  - $P(A) \wedge P(B) \ldots\ \therefore\ \forall x\ P(x)$
- Existential instantiation
  - $\exists x\ P(x)\ \therefore P(F)$ $\leftarrow$ **skolem constant F**
- Existential generalization
  - $P(A)\ \therefore\ \exists x\ P(x)$

Universal instantiation

- If $(\forall x)\ P(x)$ is true, then $P(C)$ is true, where C is *any* constant in the domain of x
- Example:

  $(\forall x)$ eats(Ziggy, x) $\Rightarrow$ eats(Ziggy, IceCream)

- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only

Existential instantiation

- From $(\exists x)\ P(x)$ infer $P(c)$
- Example:
  - $(\exists x)$ eats(Ziggy, x) $\rightarrow$ eats(Ziggy, Stuff)
- Note that the variable is replaced by a **brand-new constant** not occurring in this or any other sentence in the KB
- Also known as skolemization; constant is a **skolem constant**
- In other words, we don't want to accidentally draw other inferences about it by introducing the constant
- Convenient to use this to reason about the unknown object, rather than constantly manipulating the existential quantifier

Existential generalization

- If P(c) is true, then (∃x) P(x) is inferred.
- Example

  eats(Ziggy, IceCream) ⇒ (∃x) eats(Ziggy, x)
- All instances of the given constant symbol are replaced by the new variable symbol
- Note that the variable symbol cannot already exist anywhere in the expression

Equivalences

Theorem. Let F and G be arbitrary formulas.

(1) $\neg\forall x F \equiv \exists x \neg F$
$\neg\exists x F \equiv \forall x \neg F$

(2) If x does not occur free in G then:
$(\forall x F \wedge G) \equiv \forall x(F \wedge G)$
$(\forall x F \vee G) \equiv \forall x(F \vee G)$
$(\exists x F \wedge G) \equiv \exists x(F \wedge G)$
$(\exists x F \vee G) \equiv \exists x(F \vee G)$

(3) $(\forall x F \wedge \forall x G) \equiv \forall x(F \wedge G)$
$(\exists x F \vee \exists x G) \equiv \exists x(F \vee G)$

(4) $\forall x \forall y F \equiv \forall y \forall x F$
$\exists x \exists y F \equiv \exists y \exists x F$

We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:

$(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$
$\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$
$(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$
$(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$

Summary

- Syntax and Semantics of FOL
  - Predicates, functions, formulas, quantifiers
  - Structures and models
- Validity and satisfiability
- Inference rules in FOL
  - Universal instantiation
  - Universal generalization
  - Existential instantiation
  - Existential generalization
- Semantic equivalence

Check your understanding

Translating English to FOL

- All purple mushrooms are poisonous.
- No purple mushroom is poisonous.
- If anyone can solve the problem, then Hilary can.
- Nobody in the Calculus class is smarter than everyone in the AI class.

Check your understanding

Translating English to FOL

- **All purple mushrooms are poisonous.**
- (Ax) (mushroom(x) ^ purple(x)) => poisonous(x)

- **No purple mushroom is poisonous.**
- ¬(Ex) purple(x) ^ mushroom(x) ^ poisonous(x)
- (Ax) (mushroom(x) ^ purple(x)) => ¬ poisonous(x)

- **If anyone can solve the problem, then Hilary can.**
- $(\exists\ x\ Solves(x, problem)) \Rightarrow Solves(Hilary, problem)$

- **Nobody in the Calculus class is smarter than everyone in the AI class.**
- $\neg[\exists\ x\ TakesCalculus(x) \wedge (\forall\ y\ TakesAI(y) \Rightarrow SmarterThan(x, y))]$

Check your understanding

NF: non-formula F: formula, but not closed C: closed formula

	NF	F	C
$\forall x\ P(a)$			
$\forall x\ \exists y\ (Q(x, y) \vee R(x, y))$			
$\forall x\ (Q(x, x) \rightarrow \exists x\ Q(x, y))$			
$\forall x\ (P(x) \vee \forall x\ Q(x, x))$			
$\forall x\ (P(y) \wedge \forall y\ P(x))$			
$(P(x) \rightarrow \exists x\ Q(x, P(x)))$			
$\forall f\ \exists x\ P(f(x))$			

Check your understanding

NF: non-formula F: formula, but not closed C: closed formula

	NF	F	C
$\forall x\ (\neg\forall y\ Q(x, y) \wedge R(x, y))$			
$\exists x\ R(\forall y, x)$			
$\exists z\ ((Q(z, x) \vee R(y, z)) \rightarrow \exists y\ (R(x, y) \wedge Q(x, z)))$			
$\exists x\ (\neg P(x) \vee P(f(a)))$			
$(P(x) \rightarrow \exists x\ P(x))$			
$\exists x \forall y\ ((P(y) \rightarrow Q(x, y)) \vee \neg P(x))$			
$\forall y (R(f\ (Q(y, y))))$			
$\exists x \forall x\ Q(x, x)$			

Check your understanding

- Let F be the conjunction of 3 formulas : $F = F1 \wedge F2 \wedge F3$.
- Rxy : "x<y"
- $F1 = \forall x \exists y Rxy$
- $F2 = \neg \exists x Rxx$
- $F3 = \forall x\ \forall y\ \forall z\ [R(x,y) \wedge R(y,z) \rightarrow R(x,z)]$

- Check whether the formula is satisfiable in the domain of natural numbers.

Check your understanding

- Check whether the following FOL formulas are valid, satisfiable or unsatisfiable (contradiction) for the domain of Real numbers.

- 1. $\forall x(x^2>=0)$
- 2. $\forall x(x^2>0)$
- 3. $\exists x(x^2+1=0)$
- 4. $\exists x(x^2+x-2=0)$

Real number include natural numbers
whole numbers
integers
rational numbers