/ $\boxed{21.07.2026}$

We will study concepts involving mathematical reasoning. Let us start with a guessing game!!

## Guessing a number!

1. $< 100$ ? Y
2. $< 50$ ? N
3. $< 75$ ? Y
4. $< 62$ ? Y
5. even ? Y
6. $> 56$ ? N

## Guessing a group!

1. even order ? Y
2. abelian ? Y
3. cyclic ? N

## Guessing a graph!

1. Is it a tree? N

2. Is it a cycle? Y

Natural numbers

$0, <, >, +, \cdot, |, =$

$a < b : \exists x \left( (a + x = b) \wedge \neg (x = 0) \right)$

Groups

$*, e, {}^{-1}, =$

$\forall x \forall y \, (x * y = y * x)$

Graphs

E: edge relation.

$\forall x \forall y \, (x E y \to y E x)$

What are the common symbols in all these expressions?

$\neg, \wedge, \vee, \to, \leftrightarrow$ : connectives.

$\forall$, $\exists$ : quantifiers

$=$ : equality (not always)

$x_1, x_2, x_3, \ldots\ldots$ : variables.

( , ) : brackets

Common symbols that we use to talk about mathematics.

Language of natural numbers

$$(\underbrace{0}_{\text{constant symbol}}, \underbrace{S, +, \cdot}_{\text{function symbols}}, \underbrace{<}_{\text{relation symbol}}) \qquad (\underbrace{=}_{\text{equality symbol}})$$

Language of groups

$$(\underbrace{e}_{\text{constant symbol}}, \underbrace{*, {}^{-1}}_{\text{function symbol}}) \qquad (=)$$

Language of graphs.

$(E) \to$ relation symbol (predicate symbol)

# First order language

## Parameters of the language : $(\mathcal{C}, \mathcal{F}, \mathcal{P})$

$\mathcal{C}$ : a countable collection of constant symbols.

$\mathcal{F}$ : a countable collection of function symbols

$\mathcal{P}$ : a countable collection of relation symbols.

To each function symbol and each relation symbol, we associate a number denoting the 'arity' of the corresponding symbol.

- for any $f \in \mathcal{F}$, $\#(f)$ denotes the arity of $f$.
- for any $p \in \mathcal{P}$, $\#(p)$ denotes the arity of $p$.

Thus, we have the alphabet of our first-order language consisting of the common symbols above, and the parameters. Now, we are ready to describe the words and sentences.

## Formulas

- Any primitive formula is a formula.
- If $\varphi$ and $\psi$ are formulas, then so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$
- If $\varphi$ is a formula and $x$ is a variable, then $\forall x\, \varphi$, $\exists x\, \varphi$ are formulas.

**What is a primitive formula?**

Let us consider some examples:

$$\left.2x + 3y, \quad 7 + 2x^2 + 5x^3, \quad 5x + 7y \right\} \text{Terms}$$

$$\left.\begin{array}{l} 2x + 3y = 5x + 7y \\ 7 + 2x^2 + 5x^3 = 0 \end{array}\right\} \text{Formulas}$$

## Terms

- Any variable is a term.
- Any constant symbol is a term
- If $t_1, t_2, \ldots, t_n$ are terms, and $f^n$ is an $n$-ary function symbol, then $f^n t_1 t_2 \cdots t_n$ is a term.

## Primitive formulas.

- If $t_1, t_2, \ldots, t_n$ are terms, and $p^n$ is an $n$-ary predicate/relation symbol, then $p^n t_1 t_2 \ldots t_n$ is a primitive formula.
- If $t_1, t_2$ are terms, then $t_1 = t_2$ is a primitive formula.

## Examples (on numbers)

- $x + y$ (term)
- $x \cdot y$ (term)
- $x \cdot y < y \cdot z$ (formula)
- $\exists x \forall y (x < y)$ (formula)
- $(x = x) \wedge \neg (x = x)$ (formula).

# First-order logic : Syntax.

## Alphabet ($\mathcal{A}$)

- $\mathcal{C} = \{c_1, c_2, \ldots\}$ : a countable collection of constant symbols

- $\mathcal{F} = \{f_1^{i_1}, f_2^{i_2}, \ldots\}$ : a countable collection of function symbols

- $\mathcal{P} = \{p_1^{i_1}, p_2^{i_2}, \ldots\}$ : a countable collection of predicate symbols.

- $V = \{x_1, x_2, \ldots\}$ : a countable collection of variables

- $\equiv$ : equality symbol

- $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ : connectives

- $\forall, \exists$ : quantifiers

- $), ($ : brackets.

Terms ($\mathcal{T}$)

$$t := x_i \mid c_i \mid f_i^{i_j} t_1 t_2 \cdots t_{i_j} , \quad c_i \in \mathcal{C}, \; x_i \in V .$$

Formulas ($\mathcal{F}$)

$$\varphi, \psi := (t_1 \equiv t_2) \mid p_i^{i_j} t_1 t_2 \ldots t_{i_j} \mid (\neg(\varphi)) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid (\varphi \leftrightarrow \psi) \mid (\forall x_i (\varphi)) \mid (\exists x_i (\varphi))$$

$t_k \in \mathcal{T}$, for all $k$.

# First-order logic : Semantics

$D$ : a non-empty set (domain)

$I$ : interprets the parameters of the language

$I(c_i) \in D$, $c_i \in \mathcal{C}$.

$I(f_i^n) : \underbrace{D \times \cdots \times D}_{n \text{ times}} \to D$

$I(p_i^n) \subseteq \underbrace{D \times \cdots \times D}_{n \text{ times}}$.

$\boxed{(D, I) : \text{a structure}}$

Groups :

$\mathcal{C} = \{c\}$, $\mathcal{F} = \{f_1^2, f_2^1\}$, $\mathcal{P} = \Phi$, '$=$'

$D = G$, say. $I(c) = e_G$ $I(f_1^2) : G \times G \to G$

$I(f_2^1) : G \to G$.

## Numbers

$\mathcal{C} = \{c\}$, $\mathcal{F} = \{f_1^1, f_2^2, f_3^2\}$, $\mathcal{P} = \{p_1^2\}$, '$=$'

$D = \mathbb{N}$. $I(c) = 0$ ; $I(f_1^1) = S$ ; $I(f_2^2) = +$ ;

$I(f_3^2) = \cdot$ $I(p_1^2) = <$ .

Now, we need to give specific meaning to the variables in the domain $D$. We do that by considering an assignment function $g : V \to D$. Together, we have:

$$\boxed{(D, I, g) \quad : \quad \text{a model}}$$

Claim: Let $g : V \to D$ be an assignment function. Let $g' : \mathcal{T} \to D$ be defined as follows: (i) $g'(x) = g(x)$ for all $x \in V$;. (ii) $g'(c) = I(c)$ ; (iii) $g'(f_i^n t_1 \cdots t_n) = I(f_i^n)(g'(t_1), \cdots, g'(t_n))$. Then, $g'$ extends $g$ uniquely.

**H.W.** Prove the claim above in details.

Let us denote $g'$ by $g$ from now on.

We are ready to define :

$$\underbrace{(D, I, g)}_{\text{model}} \underbrace{\vDash}_{\text{satisfies}} \underbrace{\varphi}_{\text{formulae}}$$

1. $(D, I, g) \vDash t_1 = t_2$ if $g(t_1) = g(t_2)$
2. $(D, I, g) \vDash p_i^n t_1 \cdots t_n$ if $(g(t_1), \cdots, g(t_n)) \in I(p_i^n)$
3. $(D, I, g) \vDash \neg\varphi$ if it is not the case that $(D, I, g) \vDash \varphi$
4. $(D, I, g) \vDash \varphi \vee \psi$ if $(D, I, g) \vDash \varphi$ or $(D, I, g) \vDash \psi$
5. $(D, I, g) \vDash \varphi \wedge \psi$ if $(D, I, g) \vDash \varphi$ and $(D, I, g) \vDash \psi$
6. $(D, I, g) \vDash \varphi \rightarrow \psi$ if $(D, I, g) \vDash \varphi$ implies $(D, I, g) \vDash \psi$
7. $(D, I, g) \vDash \varphi \leftrightarrow \psi$ if $(D, I, g) \vDash \varphi$ iff $(D, I, g) \vDash \psi$
8. $(D, I, g) \vDash \forall x \varphi$ if for all $d \in D$, $(D, I, g_{[x \to d]}) \vDash \varphi$
9. $(D, I, g) \vDash \exists x \varphi$ if there exists $d \in D$, s.t. $(D, I, g_{[x \to d]}) \vDash \varphi$

How do we define $g_{[x\to d]}$ ?

$g_{[x\to d]} : V \to D$ , defined by,

$$g_{[x\to d]}(y) = \begin{cases} d & \text{if } y = x \\ g(y) & \text{if } y \neq x \end{cases}$$

Notations :

$\mathcal{M} : (D, I, g)$ ; $\mathcal{M}_{[x\to d]} : (D, I, g_{[x\to d]})$

H.W. Prove the following. For any model $\mathcal{M}$, and any formula $\varphi$, $\psi$,

(a) $\mathcal{M} \models \forall x \varphi \leftrightarrow \neg \exists x \neg \varphi$

(b) $\mathcal{M} \models \varphi \to \psi$ iff $\mathcal{M} \not\models \varphi$ or, $\mathcal{M} \models \psi$

(c) $\mathcal{M} \models \varphi \vee \neg \varphi$

(d) $\mathcal{M} \models (\varphi \vee \psi) \leftrightarrow \neg(\neg\varphi \wedge \neg\psi)$

(e) $\mathcal{M} \models (\varphi \leftrightarrow \psi) \leftrightarrow ((\varphi \to \psi) \wedge (\psi \to \varphi))$