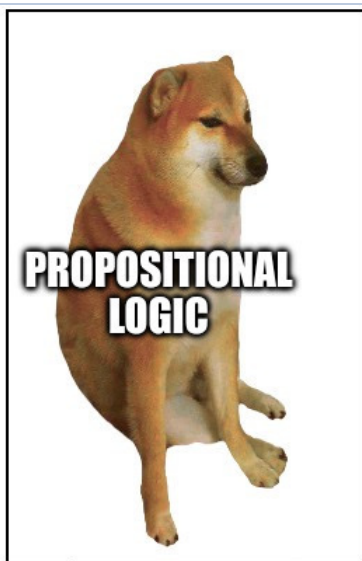


# CS 245: TUT 105 - Tutorial 06

Last time: Midterm Review

- This time:
- First-order logic: Motivation
  - FOL syntax: language, signatures, terms, formulas.
  - FOL Parse trees

Meme(s) of the week:



who are you?

imgflip.com

im you but stronger



# Review:

Today, we graduate from the study of Propositional logic to the study of First Order logic (FOL).

Q. Why do we need FOL?

A. Typically, in math, we like to talk about *things*... functions, relations, objects, sets. We need to be able to create statements describing things like equality of objects, containment, inequalities, and quantified statements about collections of objects. None of these can be done with plain Propositional logic.

E.g.: (from the notes):

To see why we need to go beyond propositional logic in the study of mathematics, consider the following basic argument:

$$\begin{array}{l} 3 + 2 = 5 \\ 5 > 1 \\ \hline \therefore 3 + 2 > 1 \end{array}$$

We can analyze this argument with propositional logic. The three statements " $3 + 2 = 5$ ", " $5 > 1$ ", and " $3 + 2 > 1$ " are each propositions. And all three of these propositions are atomic since we can't break them up into simpler propositions with our logical connectives  $\wedge$ ,  $\vee$ ,  $\neg$ ,  $\rightarrow$ , and  $\leftrightarrow$ . So the only formalization of this argument we can make is by assigning each of these propositions their own variable:

$$\begin{array}{l} p : 3 + 2 = 5 \\ q : 5 > 1 \\ r : 3 + 2 > 1. \end{array}$$

The original argument can be analyzed using these propositional variables. Its premises form the set  $\{p, q\}$  and its conclusion is  $r$ . But  $\{p, q\} \not\models r$ , so we cannot verify the validity of the argument using propositional logic.

What went wrong here?

The problem is that we have to "break up" the atomic propositions above to properly analyze the argument. We have to use the fact that " $3 + 2 = 5$ " and " $5 > 1$ " are not just generic and completely independent statements but rather statements about numbers that involve specific *constants* (here, 2, 3, 5, and 1) from some *domain*, *functions* (like addition) applied to the numbers, and *predicates* (like equality or greater than) about the numbers that are either true or false.

This is exactly what first-order logic will let us do.

Unlike in the propositional case, in FOL there is not a single "language." Instead, we define a class of languages, consisting of:

- Logical connectives:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
  - Quantifier symbols:  $\forall, \exists$
  - Punctuation symbols:  $(, ), ,$
  - Variable symbols:  $u, v, w, x, y, z, x_1, x_2, x_3, x_4, \dots$
  - Equality predicate symbol:  $=$
- ] "Logical Symbols"  
Common to ALL FOLangs

...as well as:

- A set of constant symbols:  $c$
  - A set of function symbols:  $f$
  - A set of predicate/relation symbols:  $R$
- ] "Non-logical" symbols.  
Vary b/t languages.

- arity - the # of arguments to (an interpretation of)  
a function or relation symbol.

E.g. let  $F = \{+, \times, -, \sin, \exp\}$

$C = \{\pi, \ln(2)\}$

$R = \{<, \text{In } \mathbb{Q}\}$

A "common" def'n of the arity fn.  $\text{ary}(\cdot)$  on  $F \cup R$

is:  $\text{ary}(+) = \text{ary}(\times) = \text{ary}(-) = \text{ary}(<) = 2$

$\text{ary}(\sin) = \text{ary}(\exp) = \text{ary}(\text{In } \mathbb{Q}) = 1$  //

- The **signature** of a FOLang is the union of the sets of constant, function, and relation symbols, together with their arity function:

$(C \cup F \cup R, \text{ary})$ .

E.g.'s of signatures:

• Number theory:  $\{0, S, +, \times\}$

• Graph theory:  $\{E\}$

• Group theory:  $\{1, *\}$

• Ring theory:  $\{0, 1, +, \times\}$

Given a signature  $\sigma \dots$

• Our alphabet is  $\Sigma(\sigma) := \{\text{logical symbols}\} \cup \sigma$

• Our language is  $L(\sigma) := \{\text{formulas over alphabet } \Sigma(\sigma)\}$   
defined below!

Terms & Formulas:

Terms are strings representing *elements* or *objects* of the domain. They're defined recursively.

Def'n: A string  $s$  is a term if:

- $s$  is a variable or constant symbol
- $s = f(t_1, \dots, t_n)$  where  $f \in F$ ,  $\text{ary}(f) = n$  and  $t_1, \dots, t_n$  terms
- $s = (t_1 \circ t_2)$ , where  $\circ \in F$  is binary &  $t_1, t_2$  terms.

↑ We include this in the definition so we can write binary operations as "infix," instead of "prefix".

E.g.'s:

$3$

$(x+2)$

$S(S(S(S(0))))$

$+(x_1, x(x_2, 0))$

Non-eq.'s:

$2xxxx$

$3 < x$

$3 = x$

$(3 = x) \vee \neg(3 = x)$

Formulas are statements about terms (objects). They're also defined recursively.

Def'n: A string  $s$  is a formula if

- $s = P(t_1, \dots, t_n)$  for some  $P \in R$  w/  $\text{ary}(P) = n$ ,  $t_1, \dots, t_n$  terms.
- $s = (t_1 \circ t_2)$  for some binary  $\circ \in R$ , terms  $t_1, t_2$ .

•  $s$  is of the form

$(\psi \neg \psi), (\psi \vee \psi), (\psi \rightarrow \psi), \neg \psi, (\psi \leftrightarrow \psi)$

for  $\psi, \psi$  formulas

•  $s = \forall x \psi$  or  $s = \exists x \psi$  for some variable  $x$  & formula  $\psi$ .

Note that this case contains the special logical equality symbol, "=".

E.g.'s:

$(((((3+2)=5) \wedge (5>1)) \rightarrow ((3+2)>1)) \leftrightarrow ((1 < (2+3)) \leftrightarrow ((1 < 5) \vee (5 < 1))))$   
 $((\forall S)S = 0 + S(0))$

$(\forall x \exists y (x+y=0)) \vee (\neg \forall x (x=0))$

Non-e.g.'s:

$Q(x) \forall x$

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



# Practice Problems:

## 1. FOL $\rightarrow$ Natural language translation:

Let  $\sigma = \begin{cases} \mathcal{C} = \mathbb{Z} \cup \{\pi, \sqrt{2}, i, e\} \\ \mathcal{F} = \{+, -, \cdot, \div, (\cdot)\} \\ \mathcal{R} = \{<, >, \mathbb{R}(\cdot), \mathbb{Q}(\cdot), \mathbb{Z}(\cdot), \mathbb{N}(\cdot)\}. \end{cases}$

Binary op.h for exponentiation, e.g.  $(a)^b = a^b$ .

Consider our universe of discourse to be the complex numbers, and give all these symbols the usual interpretation in that setting, as well as the unary predicates

$\mathbb{R}(\cdot), \mathbb{Q}(\cdot), \mathbb{Z}(\cdot), \mathbb{N}(\cdot)$  to indicate that a number is...

... real, rational, integral, natural, resp.

Translate the following sentences in  $L(\sigma)$  into English:

(a)  $\mathbb{N}(0)$

[I'm playing a bit fast and loose with the bracketing in some of these examples — e.g., not bracketing a chain of conjunctions because of associativity, using square brackets — because otherwise these would be impossible to read.]

(b)  $\forall x \forall y ((x+y) = (y+x))$

(c)  $\forall x \forall y \forall z \{ [(x \cdot (y+z)) = ((x \cdot y) + (x \cdot z))] \wedge [((x+y) \cdot z) = ((x \cdot z) + (y \cdot z))] \}$

(d)  $\forall x [\neg (x=0) \rightarrow \exists y ((x \cdot y) = 1)]$

(e)  $\forall x [(((x \cdot x) + 1) = 0) \rightarrow ((x=i) \vee (x=-i))]$

(f)  $\forall p [\mathbb{Z}(p) \wedge (p > 2) \wedge \forall z (\mathbb{Z}(p \div z) \rightarrow (z=1) \vee (z=p))] \rightarrow \neg \mathbb{Z}(p \div 2)$

(g)  $\neg [\exists x \exists y \exists z \exists n (\mathbb{Z}(n) \wedge \mathbb{Z}(x) \wedge \mathbb{Z}(y) \wedge \mathbb{Z}(z) \wedge n > 2 \wedge x > 2 \wedge y > 2 \wedge z > 2 \wedge (((x)^n + (y)^n) = (z)^n)]$

## 2. Natural language $\rightarrow$ FOL translation:

(From Sp. 2025, Tut. 06)

Consider the following specifications:

Domain: "all people in the world"

individual symbols, with their interpretations

a: Ayer and j: Jessica

function symbols, with their interpretations

m(u): returns the mother of u.

relation symbols, with their interpretations

$\approx$

$\neq$

G(u) : "u speaks German"

A(u) : "u is adept in the Python programming language"

S(u, v) : "u and v are siblings"

Propositional Connectives

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ .

Quantifiers

$\forall, \exists$

Punctuation Marks

"(", ")", ",", "

Translate each of the following statements into a **syntactically correct** Predicate (or first-order) logic formula:

(a) Exactly one person speaks German.

**Solution:**

$$\exists x(G(x) \wedge \forall y(G(y) \rightarrow y \approx x))$$

(b) Not all people speak German.

**Solution 1:**

$$(\neg \forall x G(x))$$

or

**Solution 2:**

$$\exists x(\neg G(x))$$

(c) At least one student is adept in the Python programming language.

**Solution 1:**

$$\exists x A(x)$$

or

**Solution 2:**

$$(\neg \forall x(\neg A(x)))$$

(d) All people are adept in the Python programming language.

**Solution 1:**

$$\forall x A(x)$$

or

**Solution 2:**

$$(\neg \exists x(\neg A(x)))$$

(e) If a person speaks German then they adept in the Python programming language.

**Solution:**

$$\forall x(G(x) \rightarrow A(x))$$

(f) There is no person who speaks German and is not adept in the Python programming language.

**Solution:**

$$(\neg \exists x(G(x) \wedge (\neg A(x))))$$

(g) Ayer is Jessica's sibling and is not adept in the Python programming language.

**Solution:**

$$(S(j, a) \wedge (\neg A(a)))$$

(h) Ayer's mother speaks German.

**Solution:**

$$G(m(a))$$

### 3. Parse Tree + Free & Bound var. identification:

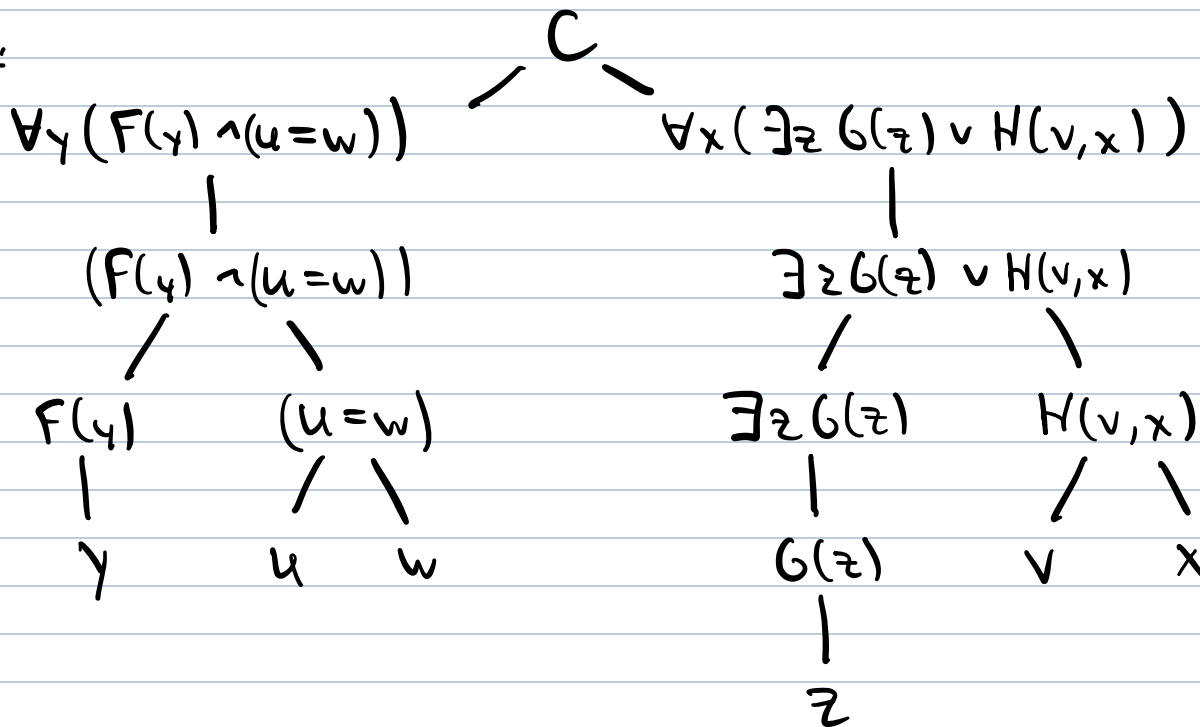
Given the following first order logic formula

(5pts)

$$C \stackrel{\text{def}}{=} (\forall y(F(y) \wedge (u \approx w)) \rightarrow \forall x(\exists z G(z) \vee H(v, x)))$$

(a) Draw the parse tree of C.

Sol.:



(b): Assuming this is a well-formed formula, which type of symbols are F, G, and H, based on context? Function or relation?

A.: Relation.

What is their arity?

A.:  $\text{ary}(F)=\text{ary}(G)=1$ ,  $\text{ary}(H)=2$ .

(c): Which vars are free, which are bound in this formula?

Free:  $u, v, w$

Bound:  $x, y, z$ .



~ Fin. ~

