Examples of first-order formulas for arithmetic (i.e., non-negative integers)

Language (non-logical symbols)

0 - constant symbol

$+$ - addition

$\cdot$ - multiplication

} binary function symbols

$\leq$ - binary predicate (aka, relation)

$=$ - equality

S - successor $\quad S(x) = x+1$. unary function symbol.

"x is even" $\quad \exists y (y + y = x)$

$\exists y (S(S(0)) \cdot y = x)$ $\quad S(S(0)) =$ "2"

"x is prime"

$\neg \exists y \exists z (S(S(y)) \cdot z = x)$ - problems.

1-one

$\neg \exists y \exists z (y \cdot z = x \wedge y \neq 1 \wedge y \neq x) \wedge 2 \leq x$ Works!!

$\neg \exists y \exists z (y \cdot z = x \wedge \neg y = S(0) \wedge \neg y = x) \wedge S(S(0)) \leq x$

There is no number between x and $x+1$

$\neg \exists z (x \leq z \wedge z \leq S(x) \wedge z \neq x \wedge z \neq S(x)))$

About "x"
x is free
y, z, - bound.

<u>Definition</u>: A language L is a set of symbol of the following types along the specification of their arity.

(1) Constant symbols (arity 0) <u>Examples</u> "0", "John"

(2) Function symbols. (arity ≥ 1) <u>Example</u> $S(\cdot)$, $+$, $\circ$, "Mother"

(3) Predicate symbols. (arity ≥ 1) <u>Examples</u> $\leq$, "Likes"

(4) The equality sign $(=)$. - Nearly always included.

These are all "non-logical symbols" (except maybe $=$)

~~The~~ An $\underline{L}$-formula will be an <u>expression</u> (string of symbols) over the following symbols:

1. Symbols of L.
2. Variables, $x_1, x_2, x_3, \ldots$ -range over "objects" or "individuals"
3. Propositional connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
4. Quantifiers
5. Parentheses, commas.

To define L-formulas

1. Define L-terms — 0, x_1, $x_1 + S(x_2)$, $+(x_1, S(x_2))$
2. Define atomic L-formulas — $0 = x_1$, $x_1 + x_2 \leq x_3$
3. Define L-formulas — $\underbrace{\forall x_2 (x_1 + x_2 \leq x_3)} \vee \underbrace{x_3 = 0}$

Defn The L-terms are inductively defined by:

(1) Any constant $c \in L$ is an L-term c.

(2) Any variable x_i is an L-term x_i

(3) If $f \in L$ is a k-ary function symbol, and $t_1, \ldots, t_k$ are L-terms, then $f(t_1, t_2, \ldots, t_k)$ is an L-term.

Example $x_1 + S(0)$

$$\left.\begin{array}{c} x_1 \\ 0 \\ S(0) \\ +(x_1, S(0)) \end{array}\right\} \text{L-terms}$$

Invalid syntax: $(x_1 = x_2) = x_3$

$x_1 = x_2$ – atomic formula

Defn The atomic L-formulas are the expressions of the form

(a) $t_1 = t_2$ where t_1, t_2 are L-terms

(b) $P(t_1, \ldots, t_k)$ where P is a k-ary predicate symbol in L and $t_1, \ldots, t_k$ are L-terms.

Examples $x_1 \leq 0$ $\quad$ $S(x_1) + (x_2 \cdot x_3) = x_4$

$+(S(x_1), \cdot(x_2, x_3)) = x_4$

Defn The L-formulas are inductively defined by:

① Any atomic L-formula is an L-formulas.

② If A, B are L-formulas, then

$\neg A$, $A \vee B$, $A \wedge B$, $A \to B$, $A \leftrightarrow B$ are L-formulas

③ If A is an L-formula, x_i is a variable $(i \geq 1)$, then $\forall x_i A$ and $\exists x_i A$ are L-formulas.

Rules for abbreviating formulas:

① Add or remove parentheses to improve readability

$\forall x_1\, x_1 \le 0$ is a formula, often written $\forall x_1 (x_1 \le 0)$

② Order of precedence:

$\forall, \exists, \neg$ are highest priority (precedence)

$\wedge, \vee$ second highest priority

$\rightarrow, \leftrightarrow$ - lowest priority

Associate from right-to-left.

③ Often write $x, y, z, \ldots$ instead $x_1, x_2, x_3, \ldots$

④ Infix notation for function symbols is allowed: $x_1 + x_2$ instead $+(x, x_2)$

+ for binary predicate symbols

$x_1 \le x_2$ instead of $\le(x_1, x_2)$

⑤ Allow $(\forall x_1)A$ to denote $\forall x_1 A$

⑥ $t \ne t'$ mean $\neg(t = t')$ $\quad \neg t = t'$

Prime(x) $\quad \neg \exists y \exists z (y \cdot z = x \wedge y \neq S(0) \wedge y \neq x) \wedge S(S(0)) \leq x.$

$$\neg \exists y (\exists z (y \cdot z = x) \wedge y \neq S(0) \wedge y \neq x) \wedge S(S(0)) \leq x.$$

y, z - bound (quantified)

x - free (not quantified)

bound occurrences of variables

free occurrences of x.

<u>Intuition</u> Formula expresses a property of x

y, z could be replaced with other variables w/o changing the meaning of the formula.

Inductive definition of free and bound occurrence of a variable, and the quantifier it is bound by:

Let A be a formula, and x_i be an occurrence of x_i in A.

(1) If A is atomic, x_i is a free occurrence.

(2) If A is $\neg B$ or $B \circ C$ ($\circ$ is $\wedge, \vee, \rightarrow, \leftrightarrow$) then x_i is free in A iff it is free in the subformula B or C

~~(3)~~, x_i is bound by the same quantifier in A as it is in B or C - if x_i is a bound occurrence.

(3) If A is $\exists x_i B$, then x_i is a bound occurrence in A.

If x_i is free in B, x_i is bound by the indicated $\exists x$.

If x_i is bound in B, x_i is bound by the same quantifier in A as it is in B.

(3') If A is $\forall x_i B$ - same definition

(4) If A is $\exists x_j B$ or $\forall x_j B$, $j \neq i$, then x_i is bound in A iff x_i is bound B. If it is a bound occurrence, ~~it~~ it is bound by the same quantifier in A as it is in in B.

$$\exists x_1 (x_1 \leq S(x_1) \wedge \forall x_1 (x_1 = x_1))$$

Scope of $\exists x_1$

Scope of $\forall x_2$

"bound by"

"reusing the variable x_1"

means the same as:

$$\exists x (x_1 \leq S(x_1) \wedge \forall x_2 (x_2 = x_2)) \models\!\dashv \exists x_1 (x_1 \leq S(x_1) \wedge \forall x_1 (x_1 = x_1))$$

Defn Let A be a formula and $\exists x_i B$ be a subformula of A.
We call B the scope of the indicated quantifier $\exists x_i$.
(Same definition for scope of $\forall x_i B$ as a subformula)

Theorem Unique Readability holds for first-order formulas.
So the scope of $\exists x_i$ or $\forall x_i$ is well-defined

Then An occurence of x_i in A is bound iff it is in the scope of some $\exists x_i$ or $\forall x_i$ in A.
And if bound, the occurrence of x_i is bound by the quantifier Qx_i which has x_i in its scope and has minimal scope

Defn A sentence is a formula with no free variables.