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

Language (non-logical symbols)

0 - constant symbol
+ - addition
× - multiplication
≤ - binary predicate (aka, relation)
= - equality
S - successor \( S(x) = x + 1 \), unary function symbol

"x is even"
\[ \exists y (y + y = x) \]
\[ \exists y (S(S(0)) \cdot y = x) \]
\[ S(S(0)) = "2" \]

"x is prime"
\[ \neg \exists y \exists z (S(S(y)) \cdot z = x) \text{ - problems} \]
\[ \neg \exists y \exists z (y \cdot z = x \land y \neq 2 \land y \neq x) \land 2 \leq x \text{ - solves} \]
\[ \neg \exists y \exists z (y \cdot z = x \land \neg y = S(0)) \land y = x \land S(S(0)) \leq x \]

There is no number between \( x \) and \( x + 1 \)
\[ \neg \exists z (x \leq z \land z \leq S(x) \land z \neq x \land z \neq S(x)) \text{ - About } \[ "x" \text{ - x is free} \]
\[ y, z, \text{ - bound} \]
**Definition:** A language \( L \) is a set of symbols of the following types along the specification of their arity.

1. Constant symbols (arity 0) **Examples:** "0", "John"  
2. Function symbols (arity \( > 1 \)) **Examples:** \( S(.), +, \times, \) "Mother"  
3. Predicate symbols (arity \( \geq 1 \)) **Examples:** \( \leq, \) "likes"  
4. The equality sign (\( = \)). - Nearly always included.

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

An \( L \)-formula will be an expression (string of symbols) over the following symbols:

1. Symbols of \( L \).
2. Variables, \( x_1, x_2, x_3, \ldots \). (Arge over objects" or "individuals")
3. Propositional connectives: \( \neg, \land, \lor, \rightarrow, \leftrightarrow \)
4. Quantifiers
5. Parentheses, commas.
To define $L$-formulas

1. Define $L$-terms
2. Define atomic $L$-formulas
3. Define $L$-formulas

\[
\begin{align*}
0, & \quad x_1, \quad x_1 + S(x_2) \\
& \quad + (x_1, S(x_2)) \quad (x_1, S(x_2)) \\
& \quad \quad \quad 0 = x_1 \quad x_1 + x_2 \leq x_3 \\
\forall x_2 \ (x_1 + x_2 \leq x_3) \lor x_3 = 0
\end{align*}
\]

Define The $L$-terms are inductively defined by:

1. Any constant $c \in C$ is an $L$-term $c$.
2. Any variable $x_i$ is an $L$-term $x_i$.
3. If $f \in C$ 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: $x_1 + S(0)$

\[
\begin{align*}
x_1, & \quad 0, \quad S(0), \\
+ (x_1, S(0))
\end{align*}
\]

$\{ L$-terms $\}$

Invalid syntax: $x_i = x_j$ is $x_3$

$x_1 = x_2$ - atomic formula
Defn The atomic $L$-formulas are the expressions of the form:

- $t_1 = t_2$ where $t_1, t_2$ are $L$-terms
- $P(t_1, \ldots, t_k)$ where $P$ is a $k$-ary predicate symbol in $L$ and $t_1, \ldots, t_k$ are $L$-terms.

Examples

- $x_1 \leq 0$
- $S(x_1) + (x_2 \cdot x_3) = x_4$
- $+ (S(x_1), o(x_2, x_3)) = x_4$

Defn The $L$-formulas are inductively defined by:

1. Any atomic $L$-formula is an $L$-formula.
2. If $A, B$ are $L$-formulas, then $\neg A, A \vee B, A \land B, A \rightarrow B, A \leftrightarrow B$ are $L$-formulas.
3. If $A$ is an $L$-formula, $x_i$ is a variable ($i > 1$), 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

\( \forall x_1, \exists x_2 \) is a formula, often written \( \forall x_1 (x_1 < 0) \)

2. Order of precedence:

\[ \land, \lor, \rightarrow \text{ are highest priority (precedence)} \]
\[ \land, \lor \text{ second highest priority} \]
\[ \rightarrow \text{ lowest priority} \]

Associate from right-to-left.

3. Often write \( x, y, z, \ldots \) instead of \( x_1, x_2, x_3, \ldots \)

4. Infix notation for function symbols is allowed: \( x_1 + x_2 \in \text{ instead of } (x_1, x_2) \)

   + for binary function symbols

\[ x_1 \leq x_2 \text{ instead of } \leq (x_1, x_2) \]

5. Allow \( (\forall x) A \) to denote \( \forall x A \)

6. \( t \neq t' \) means \( \neg (t = t') \)

   \( t = t' \)
Prime (x) \iff \exists y \exists z (y \cdot z = x \land y \neq s(0) \lor y \neq x) \land s(s(0)) \leq x.

y, z - bound (quantified)

x - free (not quantified)

Intuition: Formula expresses a property of x

y, z could be replaced with other variables

without 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 $B \lor C$ ($\lor = \lor, \lor, \lor$) 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 $C$.

4. 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 $\exists x_i$.
   - If $x_i$ is bound in $B$, $x_i$ is bound by the same quantifier in $A$ as it is in $B$.

5. If $A$ is $\forall x_i B$ — same definition.

6. If $A$ is $\exists x_j B$ or $\forall x_j B$, $j \neq i$, then $x_i$ is bound in $A$.
   - If $x_i$ is bound in $B$. If it is a bound occurrence, $x_i$ is bound by the same quantifier in $A$ as it is in $B$. 

Define Let $A$ be a formula and $\exists x_1 B$ be a subformula of $A$.

We call $B$ the scope of the indicated quantifier $\exists x_1$.

(Using definition $B$ scope of $\exists x_1 B$ as a subformula)

Theorem Unique Readability holds for first-order formulas.

So the scope of $\exists x_1$ or $\forall x_1$ is well-defined.

Then An occurrence of $x_1$ in $A$ is bound if it is in the scope of some $\exists x_1$ or $\forall x_1$ in $A$.

And if bound, the occurrence of $x_1$ is bound by the quantifier $Q x_1$ which has $x_1$ in its scope and has minimal scope.
Defn A sentence is a formula with no free variables.