FIRST-ORDER LOGIC

Pragmatics

Wolfgang Schreiner and Wolfgang Windsteiger
Wolfgang.(Schreiner|Windsteiger)@risc.jku.at
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University (JKU), Linz, Austria
http://www.risc.jku.at

JOHANNES KEPLER
UNIVERSITY LINZ

Pragmatics

We will now investigate the pragmatics (practical use) of first-order logic in two contexts.

- Defining Models
  - Introducing new domains and operations.
  - Unique characterizations of their meaning.
- Specifying Problems
  - Describing expectations for computations.
  - Assumptions on the inputs and guarantees for the outputs.

Highly relevant for computer science and mathematics.

Standard Models

We assume the following "standard models" as given.

Natural Numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$, $\mathbb{N}_n = \{0, \ldots, n-1\}$, $\mathbb{N}_{>0} = \{1, 2, \ldots\}$, etc.

Integer Numbers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

Real Numbers $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$.

- Usual arithmetic operations for all number domains.

Sets $\mathcal{P}(T)$: all sets with elements of set T.

- Element predicate $e \in S$, set builder term $\{t \mid x \in S \wedge \ldots \wedge F\}$.

Products $T_1 \times \ldots \times T_n$: all tuples $(c_1, \ldots, c_n)$ with components from $T_1, \ldots, T_n$.

- For $t = (c_1, \ldots, c_n)$ we have $t.1 = c_1$, $\ldots$, $t.n = c_n$.

Sequences T^*: all finite sequences with values from T; T^ω all infinite sequences.

- $s \in T^* : s = [s(0), s(1), s(2), \ldots, s(n-1)]$, length$(s) = n$.

The "builtin data types" of our models.

Domain Definitions

From the standard domains, we may build new domains.

- A domain definition

$$T := t$$

defines a new domain T from a term t that denotes a set (constructed from previous sets by the application of set builders and/or domain constructors).

$$\begin{aligned}
\mathsf{Nat} &:= \mathbb{N}_{2^{32}} \\
\mathsf{Int} &:= \{i \mid i \in \mathbb{Z} \land -2^{31} \leq i \land i < 2^{31}\} \\
\mathsf{IntArray} &:= \mathsf{Int}^* \\
\mathsf{IntStream} &:= \mathsf{Int}^\omega \\
\mathsf{Primes} &:= \{x \mid x \in \mathbb{N} \land x \geq 2 \land (\forall y \in \mathbb{N} : 1 < y \land y < x \rightarrow \neg(y|x))\}
\end{aligned}$$

Explicit Function Definitions

A new function my be introduced by describing its value.

- An explicit function definition
  $$f\colon T_1 \times \ldots \times T_n \to T$$
  $$f(x_1,\ldots,x_n) := t_x$$
  - introduces a new n-ary function symbol f with
  - a type signature $T_1 \times \ldots \times T_n \to T$ with sets $T_1,\ldots,T_n,T$,
  - a list of variables $x_1,\ldots,x_n$ (the parameters), and
  - a term t_x (the body) whose free variables occur in $x_1,\ldots,x_n$;
  - case $n=0$: the definition of a constant $f : T, f := t$.
- We have to show $(\forall x_1 \in T_1,\ldots,x_n \in T_n\colon t_x \in T)$ and then know
  $$\forall x_1 \in T_1,\ldots,x_n \in T_n\colon f(x_1,\ldots,x_n) = t_x$$

The body t_x may only refer to previously defined functions (no recursion).

Examples

- Definition: Let x and y be natural numbers. Then the *square sum* of x and y is the sum of the squares of x and y.

$$\begin{aligned}&\mathsf{squaresum}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}\\&\mathsf{squaresum}(x,y) := x^2+y^2\end{aligned}$$

- Definition: Let x and y be natural numbers. Then the *squared sum* of x and y is the square of z where z is the sum of x and y.

$$\begin{aligned}&\mathsf{sumsquared}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}\\&\mathsf{sumsquared}(x,y) := \textbf{let } z=x+y \textbf{ in } z^2\end{aligned}$$

- Definition: Let n be a natural number. Then the *square sum set* of n is the set of the square sums of all numbers x and y from 1 to n.

$$\begin{aligned}&\mathsf{squaresumset}\colon \mathbb{N}\to\mathcal{P}(\mathbb{N})\\&\mathsf{squaresumset}(n) := \{\mathsf{squaresum}(x,y) \mid x,y\in\mathbb{N}\wedge 1\le x\le n\wedge 1\le y\le n\}\end{aligned}$$

Predicate Definitions

A new predicate my be introduced by describing its truth value.

- An explicit predicate definition

$$p \subseteq T_1 \times \ldots \times T_n$$

$$p(x_1, \ldots, x_n) :\Leftrightarrow F_x$$

  - introduces a new n-ary predicate symbol p with
  - a type signature $T_1 \times \ldots \times T_n$ with sets $T_1, \ldots, T_n$,
  - a list of variables $x_1, \ldots, x_n$ (the parameters), and
  - a formula F (the body) whose free variables occur in $x_1, \ldots, x_n$;
  - case $n = 0$: the definition of a truth value constant $p :\Leftrightarrow F_x$.
- We then know

$$\forall x_1 \in T_1, \ldots, x_n \in T_n : p(x_1, \ldots, x_n) \leftrightarrow F_x$$

The body F_x may only refer to previously defined predicates (no recursion).

Examples

- Definition: Let x, y be natural numbers. Then x *divides* y (written as $x|y$) if $x \cdot z = y$ for some natural number z.

$$| \subseteq \mathbb{N} \times \mathbb{N}$$
$$x|y :\Leftrightarrow \exists z \in \mathbb{N} : x \cdot z = y$$

- Definition: Let x be a natural number. Then x *is prime* if x is at least two and the only divisors of x are one and x itself.

$$\text{isprime} \subseteq \mathbb{N}$$
$$\text{isprime}(x) :\Leftrightarrow x \geq 2 \wedge \forall y \in \mathbb{N} : y|x \rightarrow y = 1 \vee y = x$$

- Definition: Let p, n be a natural numbers. Then p is a *prime factor* of n, if p is prime and divides n.

$$\text{isprimefactor} \subseteq \mathbb{N} \times \mathbb{N}$$
$$\text{isprimefactor}(p, n) :\Leftrightarrow \text{isprime}(p) \wedge p|n$$

Implicit Function Definitions

A new function may be introduced by a condition on its result value.

- An implicit function definition

$$f : T_1 \times \ldots \times T_n \to T$$

$$f(x_1, \ldots, x_n) := \textbf{such } y : F_{x,y} \text{ (or: } \textbf{the } y : F_{x,y})$$

  - introduces a new n-ary function constant f with
  - a type signature $T_1 \times \ldots \times T_n \to T$ with sets $T_1, \ldots, T_n, T$,
  - a list of variables $x_1, \ldots, x_n$ (the parameters),
  - a variable y (the result variable),
  - a formula $F_{x,y}$ (the result condition) whose free variables occur in $x_1, \ldots, x_n, y$.
- We then know

$$\forall x_1 \in T_1, \ldots, x_n \in T_n : (\exists y \in T : F_{x,y}) \to (\exists y \in T : F_{x,y} \land y = f(x_1, \ldots, x_n))$$

  - If some value satisfies the condition, the function result is such a value.
  - With **the** we claim that the value of f always exists and is unique.

The definition of a function by a formula (rather than a term).

Examples

- Definition: *A root* of real number x is a real number y such that the square of y is x.

$$\begin{aligned}&\text{aRoot}\colon \mathbb{R} \to \mathbb{R}\\ &\text{aRoot}(x) := \textbf{such } y\colon y^2 = x\end{aligned}$$

- Definition: *The root* of non-negative real x is that real y such that the square of y and $y \geq 0$.

$$\begin{aligned}&\text{theRoot}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}\\ &\text{theRoot}(x) := \textbf{the } y\colon y^2 = x \wedge y \geq 0\end{aligned}$$

- Definition: Let $m, n \in \mathbb{N}$ with n positive. Then the *(truncated) quotient* $q \in \mathbb{N}$ of m and n is such that $m = n \cdot q + r$ for some $r \in \mathbb{N}$ with $r < n$.

$$\begin{aligned}&\text{quotient}\colon \mathbb{N} \times \mathbb{N}_{>0} \to \mathbb{N}\\ &\text{quotient}(m,n) := \textbf{the } q\colon \exists r \in \mathbb{N}\colon m = n \cdot q + r \wedge r < n\end{aligned}$$

- Definition: Let x, y be positive natural numbers. The *greatest common divisor* of x and y is the greatest such number that divides both x and y.

$$\begin{aligned}&\text{gcd}\colon \mathbb{N}_{>0} \times \mathbb{N}_{>0} \to \mathbb{N}_{>0}\\ &\text{gcd}(x,y) := \textbf{the } z\colon z|x \wedge z|y \wedge \forall z' \in \mathbb{N}_{>0}\colon z'|x \wedge z'|y \to z' \leq z\end{aligned}$$

Predicates versus Functions

A predicate can give rise to functions in two ways.

- A predicate:

$$\mathsf{isprimefactor} \subseteq \mathbb{N} \times \mathbb{N}$$

$$\mathsf{isprimefactor}(p,n) :\Leftrightarrow \mathsf{isprime}(p) \wedge p|n$$

- An implicitly defined function:

$$\mathsf{someprimefactor} : \mathbb{N} \to \mathbb{N}$$

$$\mathsf{someprimefactor}(n) := \textbf{such } p : \mathsf{isprime}(p) \wedge p|n$$

- An explicitly defined function whose result is a set:

$$\mathsf{allprimefactors} : \mathbb{N} \to \mathcal{P}(\mathbb{N})$$

$$\mathsf{allprimefactors}(n) := \{p \in \mathbb{N} \mid \mathsf{isprime}(p) \wedge p|n\}$$

The preferred style of definition is a matter of taste and purpose.

Specifying Problems

An important role of logic in computer science is to specify problems.

- The specification of a (computational) problem

$$\textbf{Input: } x_1 \in T_1, \ldots, x_n \in T_n \quad \textbf{where } I_x$$

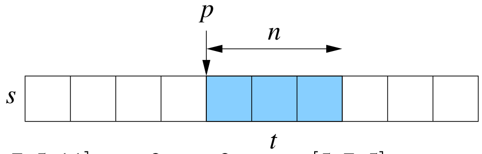
$$\textbf{Output: } y_1 \in U_1, \ldots, y_m \in U_m \quad \textbf{where } O_{x,y}$$

  - consists of a list of input variables $x_1, \ldots, x_n$ with types $T_1, \ldots, T_n$,
  - a formula I_x (the input condition or precondition) whose free variables occur in
$$x_1, \ldots, x_n$$
  - a list of output variables $y_1, \ldots, y_m$ with types $U_1, \ldots, U_m$, and
  - a formula $O_{x,y}$ (the output condition or postcondition) whose free variables occur in
$$x_1, \ldots, x_n, y_1, \ldots, y_m$$

The specification is expressed with the help of auxiliary functions and predicates.

Example

Problem: extract from a finite sequence s of natural numbers a subsequence t of length n starting at position p.



Example: $s = [2,3,5,7,5,11], p = 2, n = 3 \leadsto t = [5,7,5]$

Input: $s \in \mathbb{N}^*, n \in \mathbb{N}, p \in \mathbb{N}$ **where**

$n + p \leq \text{length}(s)$ *(subsequence is in range of array)*

Output: $t \in \mathbb{N}^*$ **where**

$\text{length}(t) = n \;\wedge$ *(length of result sequence)*

$\forall i \in \mathbb{N}_n : t(i) = s(i+p)$ *(content of result sequence)*

The Adequacy of Specifications

Input: x **where** I_x **Output:** y **where** $O_{x,y}$

- Is precondition satisfiable? $(\exists x \colon I_x)$
  Otherwise no input is allowed.
- Is precondition not trivial? $(\exists x \colon \neg I_x)$
  Otherwise every input is allowed, why then the precondition?
- Is postcondition always satisfiable? $(\forall x \colon I_x \rightarrow \exists y \colon O_{x,y})$
  Otherwise no implementation is legal.
- Is postcondition not always trivial? $(\exists x, y \colon I_x \wedge \neg O_{x,y})$
  Otherwise every implementation is legal.
- Is result unique? $(\forall x, y_1, y_2 \colon (I_x \wedge O_{x,y}[y_1/y] \wedge O_{x,y}[y_2/y] \rightarrow y_1 = y_2))$
  Whether this is required, depends on our expectations.

Ask these questions to ensure that specification expresses your intentions.

Example: The Problem of Integer Division

Input: $m \in \mathbb{N}, n \in \mathbb{N}$ **Output:** $q \in \mathbb{N}, r \in \mathbb{N}$ **where** $m = n \cdot q + r$

- The postcondition is always satisfiable but not trivial.
  - For $m = 13, n = 5$, e.g. $q = 2, r = 3$ is legal but $q = 2, r = 4$ is not.
- But the result is not unique.
  - For $m = 13, n = 5$, both $q = 2, r = 3$ and $q = 1, r = 8$ are legal.

Input: $m \in \mathbb{N}, n \in \mathbb{N}$ **Output:** $q \in \mathbb{N}, r \in \mathbb{N}$ **where** $m = n \cdot q + r \wedge r < n$

- Now the postcondition is not always satisfiable.
  - For $m = 13, n = 0$, no output is legal.

Input: $m \in \mathbb{N}, n \in \mathbb{N}$ **where** $n \neq 0$ **Output:** $q \in \mathbb{N}, r \in \mathbb{N}$ **where** $m = n \cdot q + r \wedge r < n$

- The precondition is not trival but satisfiable.
  - $m = 13, n = 0$ is not legal but $m = 13, n = 5$ is.
- The postcondition is always satisfiable and result is unique.
  - For $m = 13, n = 5$, only $q = 2, r = 3$ is legal.

Example: The Problem of Linear Search

Problem: given a finite integer sequence a and an integer x, determine the smallest position p at which x occurs in a ($p = -1$, if x does not occur in a).

Example: $a = [2,3,5,7,5,11], x = 5 \rightsquigarrow p = 2$

Input: $a \in \mathbb{Z}^*, x \in \mathbb{Z}$

Output: $p \in \mathbb{N} \cup \{-1\}$ **where**

let $n = \text{length}(a)$ **in**
if $\exists p \in \mathbb{N}_n : a(p) = x$ *(x occurs in a)*
then $p \in \mathbb{N}_n \wedge a(p) = x \wedge$ *(p is the index of some occurrence of x)*
$(\forall q \in \mathbb{N}_n : a(q) = x \rightarrow p \leq q)$ *(p is the smallest such index)*
else $p = -1$

All inputs are legal; the result always exists and is uniquely determined.

Example: The Problem of Binary Search

Problem: given a finite integer sequence a that is sorted in ascending order and an integer x, determine some position p at which x occurs in a ($p = -1$, if x does not occur in a).

Example: $a = [2,3,5,5,5,7,11], x = 5 \rightsquigarrow p \in \{2,3,4\}$

Input: $a \in \mathbb{Z}^*, x \in \mathbb{Z}$ **where**

let $n = \text{length}(a)$ **in**
$\forall k \in \mathbb{N}_{n-1} : a(k) \le a(k+1)$ *(a is sorted)*

Output: $p \in \mathbb{N} \cup \{-1\}$ **where**

let $n = \text{length}(a)$ **in**
if $\exists p \in \mathbb{N}_n : a(p) = x$ *(x occurs in a)*
then $p \in \mathbb{N}_n \wedge a(p) = x$ *(p is the index of some occurrence of x)*
else $p = -1$

Not all inputs are legal; for every legal input, the result exists but is not unique.

Example: The Problem of Sorting

Problem: given a finite integer sequence a, determine that permutation b of a that is sorted in ascending order.

Example: $a = [5,3,7,2,3] \rightsquigarrow b = [2,3,3,5,7]$

Input: $a \in \mathbb{Z}^*$

Output: $b \in \mathbb{N}^*$ **where**

$$
\begin{array}{ll}
\textbf{let } n = \text{length}(a) \textbf{ in} & \\
\text{length}(b) = n \ \land & \\
(\forall k \in \mathbb{N}_{n-1} : b(k) \le b(k+1)) \ \land & \textit{(b is sorted)} \\
\exists p \in {\mathbb{N}_n}^* : & \textit{(b is a permutation of a)} \\
\quad (\forall k1 \in \mathbb{N}_n, k2 \in \mathbb{N}_n : k1 \neq k2 \rightarrow p(k1) \neq p(k2)) \ \land & \\
\quad (\forall k \in \mathbb{N}_n : a(k) = b(p(k))) &
\end{array}
$$

All inputs are legal; the result always exists and is uniquely determined.

Implementing Problem Specifications

Input: $x_1 \in T_1, \ldots, x_n \in T_n$ **where** I_x

Output: $y_1 \in U_1, \ldots, y_m \in U_m$ **where** $O_{x,y}$

- Specification demands definition of function $f\colon T_1 \times \ldots \times T_n \to U_1 \times \ldots \times U_m$ with property
$$\forall x_1 \in T_1, \ldots, x_n \in T_n : I_x \to \textbf{let } (y_1, \ldots, y_m) = f(x_1, \ldots, x_n) \textbf{ in } O_{x,y}$$
  - For all arguments $x_1, \ldots, x_n$ that satisfy the input condition,
  - the result $(y_1, \ldots, y_m)$ of f satisfies the output condition.
- The specification itself already implicitly defines such a function:
$$f(x_1, \ldots, x_n) := \textbf{such } y_1, \ldots, y_m : I_x \to O_{x,y}$$
- However, actually we want an explicitly defined function (computer program):
$$f(x_1, \ldots, x_n) := t_x$$

A core goal of computer science is to specify problems, to implement the specifications, and to verify the correctness of the implementation (e.g., by formal methods).