Products & Bindings

Contents

Products

What is a product?

A *product* is: an ordered finite collection of values of possibly different types.

Examples:

(1,2)
(1,**true**,"**John**")

Product Type Expressions

$$\text{type\_expr}_1 \times \ldots \times \text{type\_expr}_n, \quad n \geq 2$$

denotes the type consisting of all values

$$(v_1,\ldots,v_n), \quad \text{where } v_i : \text{type\_expr}_i$$

Operators:

$=$
$\neq$

Product Type Expessions

Examples

Bool × **Bool**

denotes the type consisting of the values:

(**true**,**true**)
(**true**,**false**)
(**false**,**true**)
(**false**,**false**)

Nat × **Nat** × **Bool**

denotes the type consisting of the values:

(0,0,**true**)
(0,0,**false**)
(0,1,**true**)
(0,1,**false**)
(1,0,**true**)
(1,0,**false**)
(2,0,**true**)
⋮

Product Value Expressions

$(\text{value\_expr}_1,\ldots,\text{value\_expr}_n)$, $n \geq 2$

syntax	semantics
e_1	v_1
$\vdots$	$\vdots$
e_n	v_n
$(e_1,\ldots,e_n)$	$(v_1,\ldots,v_n)$

Examples:

$(\textbf{true} \vee \textbf{false}, 7 + 2)$ represents $(\textbf{true}, 9)$

Example: A System of Coordinates I

scheme SYSTEM_OF_COORDINATES =
  class
    type
      Position = **Real** × **Real**
    value
      origin : Position = (0.0,0.0),
      distance : Position × Position → **Real**
      distance((x1,y1),(x2,y2)) ≡
        ((x2−x1)↑2.0 + (y2−y1)↑2.0)↑0.5
  end

Example: A System of Coordinates II

scheme SYSTEM_OF_COORDINATES =
  class
    type
      Position = **Real** × **Real**
    value
      origin : Position = (0.0,0.0),
      distance : Position × Position → **Real**
      distance(p1,p2) ≡
        let
          (x1,y1) = p1,
          (x2,y2) = p2
        in
          ((x2−x1)↑2.0 + (y2−y1)↑2.0)↑0.5
        end
  end

(Explicit) Let Expressions

let
  $\text{binding}_1 = \text{expr}_1$,
    ⋮
  $\text{binding}_n = \text{expr}_n$
in
  expr
end

Bindings and Typings

Bindings

Examples:

x
(x,y)
(x,y,z)
((x1,y1),(x2,y2))
(x,(y,z))

Forms:

id
(binding_1, ... , binding_n)

Use:

- typings
- let expressions
- formal function parameters
- ...

Typings

Examples:

x : **Real**
(x,y) : Position
a, b : **Real**

Basic Form:

binding : type_expr

Context condition:
binding must match type_expr

Derived form:

binding_1, ... , binding_n : type_expr

Use:

- value definitions (e.g. **value** x : **Real**)
- quantified expressions (e.g. $\forall$ x : **Real**)
- ...