If You See What I Mean

Fixing and Avoiding Broken Programs by Thinking in Types

Eric Smith

@eric_s_smith

Vermont Code Camp, Sept. 2015

Paradox

This statement is false.

Barber's Paradox

The barber is a man in town who shaves all those, and only those, men in town who do not shave themselves.

Does the barber shave himself?

Russell's Paradox

X = { Sets that don't contain themselves }

Does it contain itself? Yes and No.

Doctrine of Types

APPENDIX B.

THE DOCTRINE OF TYPES.

497. THE doctrine of types is here put forward tentatively, as affording a possible solution of the contradiction; but it requires, in all probability, to be transformed into some subtler shape before it can answer all difficulties. In case, however, it should be found to be a first step towards the truth, I shall endeavour in this Appendix to set forth its main outlines, as well as some problems which it fails to solve.

Every propositional function $\phi(x)$—so it is contended—has, in addition to its range of truth, a range of significance, *i.e.* a range within which x must lie if $\phi(x)$ is to be a proposition at all, whether true or false. This is the first point in the theory of types; the second point is that ranges of significance form *types*, *i.e.* if x belongs to the range of significance of $\phi(x)$, then there is a class of objects, the *type* of x, all of which must also belong to the range of significance of $\phi(x)$, however ϕ may be varied; and the range of significance is always either a single type or a sum of several whole types. The second point is less precise than the first, and the case of numbers introduces difficulties; but in what follows its importance and meaning will, I hope, become plainer.

A *term* or *individual* is any object which is not a range. This is the lowest type of object. If such an object—say a certain point in space—occurs in a proposition, any other individual may *always* be substituted without loss of significance. What we called, in Chapter VI, the class as one, is an individual, provided its members are individuals: the objects of daily life, persons, tables, chairs, apples, etc., are classes as one. (A person

Principles of Mathematics (1903)

Simply Typed Lambda Calculus

Types

$T ::= B \mid T_1 \to T_2$

$B \in \{\text{Bool}, \text{NatNum}, \text{RomNum}, \text{USD}, \text{FrenchVowel}\}$

Terms

$e ::= x \mid \lambda x : T\,.\,e \mid e_1\, e_2$

Plug

The input domain of your function is probably not the types built into the language.

Unnamed Functions

Javascript (ES6)

(a, b) => return a + b;

Java 8

(a, b) -> a + b

PHP

function(a, b) return a + b;

Function Abstraction & Application

Abstraction:

10 * 8
10 * 9
10 * 11
10 * x

f(x) = 10 * x

λx . 10 * x

Application:

f(8) = 10*8
f(9) = 10*9
f(11) = 10*11

(λx . 10 * x) 11

Bound & Free Variables

An instance of a variable is “free” in a lambda expression if it is not “bound” by a lambda.

$\lambda x\,.\;xyz$

$\lambda y\,.\;x(\lambda x\,.\;xx)x$

$\lambda x\,.\;\lambda y\,.\;xy$

Non-Termination

$(\lambda x \,.\, xx)(\lambda x \,.\, xx)$

Tips

- The input domain of your function is probably not the types built into the language.
- If you expect to get an answer from your program it needs to stop.

Wouldn't be nice if your type system helped you with this.

Typing Judgement

Type Environment:

$\Gamma \equiv x_1 : B_1, x_2 : B_2, \ldots\ x_n : B_n$

Typing Judgement:

$\Gamma \vdash t : A$

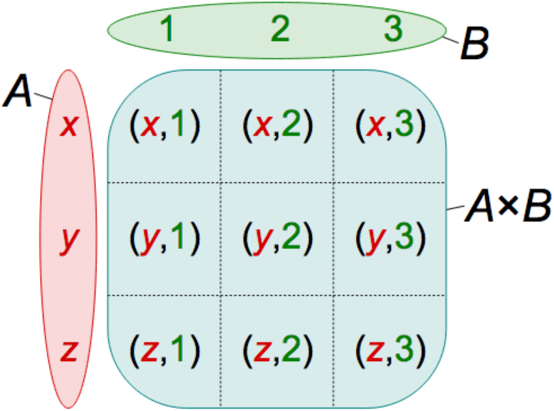
Typing Rule - Function Abstraction

$$\frac{\Gamma, x : A \ \vdash t : B}{\Gamma \vdash \lambda\, x\, .\, t : A \to B} \to\text{-I}$$

Typing Rule - Function Application

$$\frac{\Gamma \vdash t : A \rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash (t\ u) : B} \rightarrow\text{-E}$$

Cartesian Product

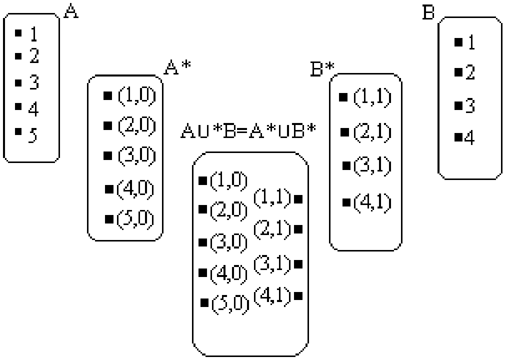


Typing Rule - Product

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash \langle\, t, u \,\rangle : A \times B} \times\text{-I}$$

$$\frac{\Gamma \vdash t : A \times B}{\Gamma \vdash t.fst : A} \times\text{-E} \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash t.snd : B} \times\text{-E}$$

Disjoint Union



Typing Rule - Sum

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{inl}\ t : A + B} +\text{-I} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathsf{inr}\ t : A + B} +\text{-I}$$

$$\frac{\Gamma \vdash t : A + B \qquad \Gamma,\ x : A \vdash t : T \qquad \Gamma,\ y : B \vdash t : T}{\Gamma \vdash \mathsf{case}\ t\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow t_1 \mid \mathsf{inr}\ y \Rightarrow t_2 : T} +\text{-E}$$

Plug

If your language does not have decent support for Sum types, you are fighting an uphill battle.

Agda Boolean

DEMO

Peano Axioms

- Zero is a natural number.
- For every natural number n, the successor function, $S(n)$ is a natural number.

$S(0) = 1$
$S(S(0)) = 2$
$S(S(S(0))) = 3$

Agda Natural Number

DEMO

Agda Record

DEMO

Plug

"How To Design Programs" will give you data driven recipes for writing functions that are based on the Elimination forms of data structures.

Propositional Logic

Propositional Symbols:

$P, Q, ...$

Logical Connectives:

$\vee$ (and)
$\wedge$ (or)
$\supset$ (implies)
$\neg$ (not)

Propositional Logic Examples

P - "Today is Saturday"
Q - "We are in Vermont"

$P \vee Q$ - "Today is Saturday AND we are in Vermont"
$P \wedge Q$ - "Today is Saturday OR we are in Vermont"
$P \supset Q$ - "IF today is Saturday, THEN we are in Vermont"
$\neg P$ - "Today is NOT Saturday"

Intuitionistic vs Classical Logic

Classical logic:

- Propositions are true or they are false (Law of Excluded Middle).
- Truth tables for connectives can be used to determine true/false of complicated formulas.

Intuitistic logic:

- Focus is on evidence & proof
- Proposition is either true or not able to be proven true

Natural Deduction - $\wedge$

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ \wedge\text{-I}$$

$$\frac{\vdash A \wedge B}{\vdash A}\ \wedge\text{-E}_1 \qquad\qquad \frac{\vdash A \wedge B}{\vdash B}\ \wedge\text{-E}_2$$

Natural Deduction - $\supset$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B} \supset\text{-I}$$

$$\frac{\Gamma \vdash A \supset B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B} \supset\text{-E}$$

Natural Deduction - $\vee$

Skip because it's weirder than we have time for.

Curry-Howard Correspondence

Propositions	Types
Proofs	Terms
Proof simplification	Evaluation

Agda Dependent Types

DEMO

Plug

https://www.manning.com/books/type-driven-development-with-begin

50% OFF - mltdidris