Quotient Types by Idempotent Functions in Cedille

Andrew Marmaduke, Christopher Jenkins, Aaron Stump

The University of Iowa

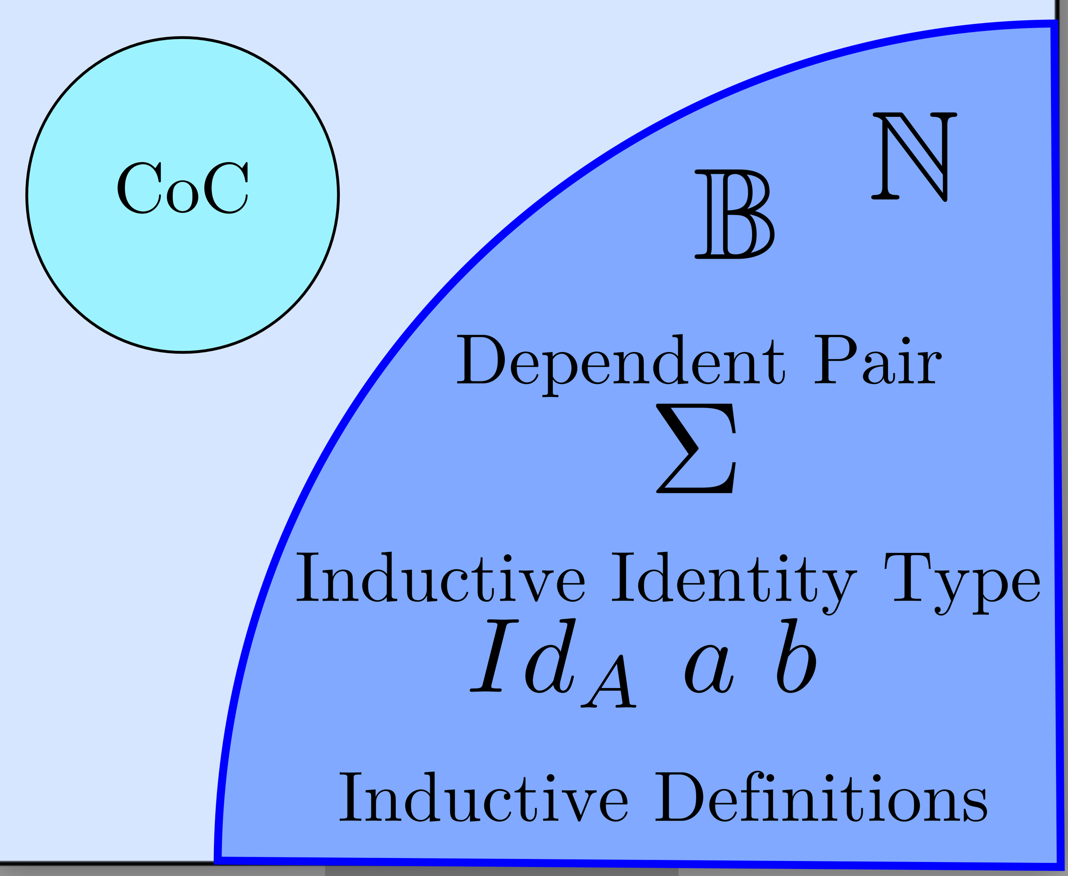
Guide

What is a Type Theory?

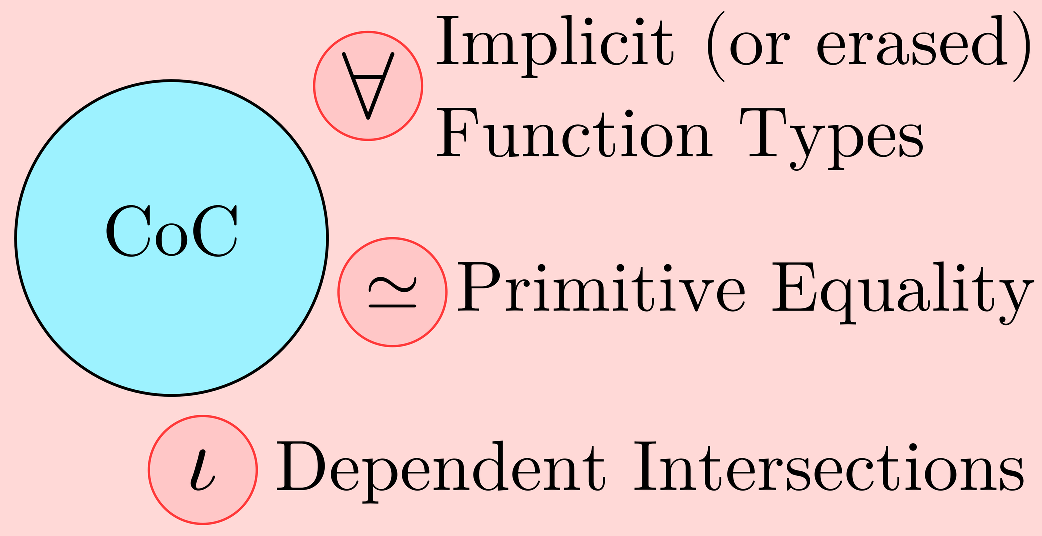
- Foundational theory for logic, math and computer science
- Commonly built on the lambda calculus
- Most common core is the Calculus of Constructions (CoC) (with function types, type quantification, etc)

Traditional Type Theory

(e.g. Agda and Coq)

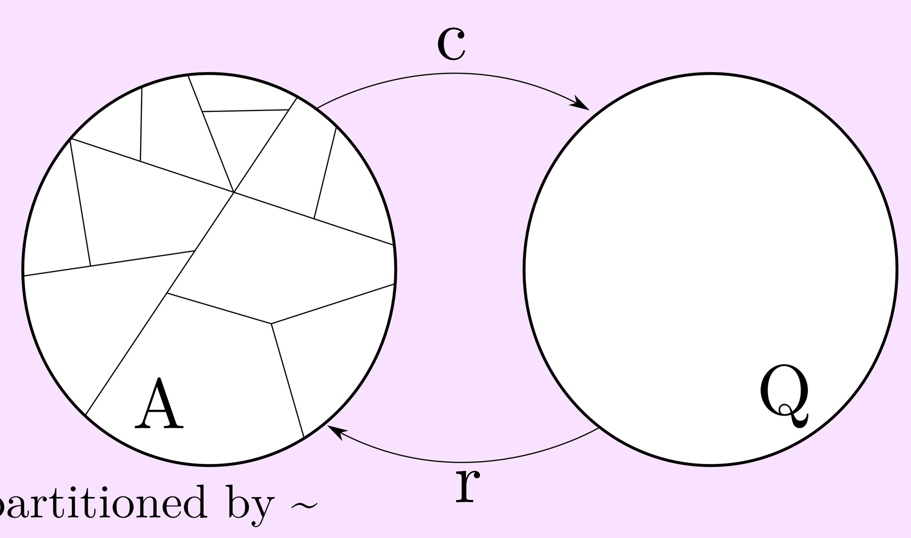


Cedille



Inductive Definitions are derived from these components

What is a Definable Quotient Type?



A partitioned by ~

Given: $\forall a, b \in A.\text{if } a \sim b \text{ then } c(a) = c(b)$
$\forall a \in A.r(c(a)) \sim a$
$\forall q \in Q.c(r(q)) = q$

Then: Q is a quotient of A by ~

Given A and ~

Define: $f : A \to A$ such that

$\forall a, b \in A.\text{if } a \sim b \text{ then } f(a) = f(b)$

$\forall a \in A.f(a) \sim a$

Then $Q = \Sigma a \in A.Id_A\ f(a)\ a$

Example:

$$(\mathbb{N}, \mathbb{N}) \to (\mathbb{N}, \mathbb{N})$$

$f =$ reduce to irreducible fraction

$$\mathbb{Q} = \Sigma(n, d) \in (\mathbb{N}, \mathbb{N}).Id_{(\mathbb{N},\mathbb{N})}\ f((n, d))\ (n, d)$$

Given A

Define $f : A \to A$ such that $\forall a \in A.f(f(a)) = f(a)$

Then $Q = \iota a \in A.f(a) \simeq a$

Equivalent to traditional definition

Key Difference: $\sim$ is defined by f

Example:

$$List\ A \to List\ A$$

$f =$ sort

$$SortedList\ A = \iota \ell \in List\ A.f(\ell) \simeq \ell$$

Discussion of Definable Quotient Types

- Not all quotients are definable in the above sense (e.g. unordered pairs)
- Not all definable quotients can be defined as inductive types
- Quotients give a different perspective on constructing types

---

Quotients give two different meanings for a concept:

Setoid: $(\mathbb{N}, \mathbb{N})$ and $\sim$
Set: $\mathbb{Q}$ and $Id_{\mathbb{Q}}$

Proofs can benefit from both

Projections can get in the way of equalities between views:

$$f((n, m)) = (n, m)$$
$$(n, m) = \pi_1 q$$

where π_1 is the first projection

All the same benefits **and** Q is a subtype of A

$$\forall q \in Q.\exists a \in A.q = a$$

This prevents projections from interfering with equalities between Q and A

$$f((n, m)) = (n, m)$$
$$(n, m) = q$$

Conclusions

1. Quotients by idempotent functions hide the equivalence relation ~
2. Cedille's encoding of quotients by idempotent functions gives you, additionally, subtypes

Related Work

1. Nuo Li, Quotient Types in Type Theory
2. Cyril Cohen, Pragmatic Quotient Types in Coq
3. Homotopy Type Theory for an alternative approach to quotients in a nontraditional theory