A Cubical Implementation of Homotopical Patch Theory

Åsmund Aqissiaq Arild Kløvstad

Universitetet i Bergen

June 23rd, 2022

Overview

Homotopy Type Theory

Version Control Systems

Three Homotopical Patch Theories

1. An Elementary Patch Theory
2. A Patch Theory with laws
3. A Patch Theory with Richer Contexts

Homotopy Type Theory

Version Control Systems

Three Homotopical Patch Theories

1. An Elementary Patch Theory
2. A Patch Theory with laws
3. A Patch Theory with Richer Contexts

Types

- Types and terms

$$\mathsf{naturals} : \mathsf{Type}$$
$$\mathsf{naturals} = \mathbb{N}$$

$$\mathsf{aNumber} : \mathsf{naturals}$$
$$\mathsf{aNumber} = 0$$

- Elimination

$$\mathsf{double} : \mathbb{N} \to \mathbb{N}$$
$$\mathsf{double}\ \mathsf{zero} = \mathsf{zero}$$
$$\mathsf{double}\ (\mathsf{suc}\ n) = \mathsf{suc}\ (\mathsf{suc}\ (\mathsf{double}\ n))$$

- Type families

$$\mathsf{data}\ \mathsf{List}\ (A : \mathsf{Type}) : \mathsf{Type}\ \mathsf{where}$$
$$\quad [\,]\mathsf{l} : \mathsf{List}\ A$$
$$\quad \_{::}\mathsf{l}\_ : A \to \mathsf{List}\ A \to \mathsf{List}\ A$$

Dependent Types I

- A type that *depends* on a term

```
data Vec (A : Type) : ℕ → Type where
  [] : Vec A 0
  _::_ : ∀ {n} → A → Vec A n → Vec A (suc n)
```

- Dependent elimination

```
Vec-induction :
  {A : Type} → {P : ∀ {n} → Vec A n → Type} →
  (P []) →
  (∀ {n} → (a : A) → (as : Vec A n) → P (a :: as)) →
  {n : ℕ} → (v : Vec A n) → P v
```

Dependent Types II

$$\text{Vec-induction}\ \mathit{empty}\ \_\ []\ =\ \mathit{empty}$$
$$\text{Vec-induction}\ \_\ \mathit{cons}\ (a :: v)\ =\ \mathit{cons}\ a\ v$$

- Π-types

$$\Pi : (X : \text{Type}) \to (X \to \text{Type}) \to \text{Type}$$
$$\Pi\ X\ P = (x : X) \to P\ x$$

$$\text{countDown} : \Pi\ \mathbb{N}\ (\text{Vec}\ \mathbb{N})$$
$$\text{countDown}\ \text{zero} = []$$
$$\text{countDown}\ (\text{suc}\ x) = x :: (\text{countDown}\ x)$$

Dependent Types III

- Σ-types

```
record Σ (X : Type) (P : X → Type) : Type where
  field
    fst : X
    snd : P fst

_ : Σ ℕ (Vec ℕ)
_ = record { fst = 2 ; snd = 0 :: (1 :: []) }
```

Identity Types I

- When are two terms the same?

```
data Id {X : Type} (x : X) : X → Type where
  refl : Id x x

_ : Id (double 2) 4
_ = refl
```

Identity Types II

- The J rule (identity induction)

$$
\begin{array}{l}
\mathsf{J} : \{X : \mathsf{Type}\}\ \{x : X\} \to \\
\quad (P : (y : X) \to (\mathsf{Id}\ x\ y) \to \mathsf{Type}) \to \\
\quad (base : P\ x\ \mathsf{refl}) \to \\
\quad \overline{\hphantom{(y : X) \to (p : \mathsf{Id}\ x\ y) \to P\ y\ p \qquad\qquad\qquad}} \\
\quad (y : X) \to (p : \mathsf{Id}\ x\ y) \to P\ y\ p \\
\mathsf{J}\ P\ base\ y\ \mathsf{refl} = base
\end{array}
$$

Groupoids

- Identity types form an equivalence relation (symmetric)
- $\neg$UIP
- Groupoids! [cite]

Higher Inductive Types

Homotopy Type Theory

Version Control Systems

Three Homotopical Patch Theories

1. An Elementary Patch Theory
2. A Patch Theory with laws
3. A Patch Theory with Richer Contexts

Setup

Groupoid Structure

A Patch Theory

Homotopical Patch Theory

Homotopy Type Theory

Version Control Systems

Three Homotopical Patch Theories

1. An Elementary Patch Theory
2. A Patch Theory with laws
3. A Patch Theory with Richer Contexts