The encode-decode method in HoTT, relationally

Fredrik Nordvall Forsberg

joint work with James McKinna

MSP group, Strathclyde

STP @ Dundee, 7 October 2015

Relational reasoning à la Burstall

Burstall's insight: fold-ing lists (1969)

Theorem Given $A, B : \mathcal{U}, b : B, f : A \to B \to B$, define

- $\text{fold}\, f\, b\, [] = b$
- $\text{fold}\, f\, b\, (a :: as) = f\, a\, (\text{fold}\, f\, b\, as)$

Burstall's insight: fold-ing lists (1969)

Theorem Given $A, B : \mathcal{U}, b : B, f : A \to B \to B$, define

- $\mathrm{fold}\, f\, b\, [] = b$
- $\mathrm{fold}\, f\, b\, (a :: as) = f\, a\, (\mathrm{fold}\, f\, b\, as)$

Then for all A, B, b, f as above, and $F : \mathrm{List}\, A \to B \to \mathcal{U}$,

- if $\frac{}{F\, []\, b}$ and $\frac{F\, as\, r}{F\, (a :: as)\, (f\, a\, r)}$,

Burstall's insight: fold-ing lists (1969)

Theorem Given $A, B : \mathcal{U}, b : B, f : A \to B \to B$, define

- $\text{fold}\, f\, b\, [\,] = b$
- $\text{fold}\, f\, b\, (a :: as) = f\, a\, (\text{fold}\, f\, b\, as)$

Then for all A, B, b, f as above, and $F : \text{List}\, A \to B \to \mathcal{U}$,

- if $\dfrac{}{F\, [\,]\, b}$ and $\dfrac{F\, as\, r}{F\, (a :: as)\, (f\, a\, r)}$,
- then for all $as : \text{List}\, A$, we have $F\, as\, (\text{fold}\, f\, b\, as)$.

Proof Induction on $as : \text{List}\, A$.

Induction for functions

- every $f : X \to Y$ gives rise to graph relation $y = f\,x$
- recursive f may be simulated by an inductive $F\,x\,y$

Induction for functions

- every $f : X \to Y$ gives rise to graph relation $y = f\,x$
- recursive f may be simulated by an inductive $F\,x\,y$
  - (partial correctness) **soundness**

$$\mathrm{snd}_f(F) : (\Pi x : X)\,(\Pi y : Y)\,F\,x\,y \to (y = f\,x)$$

(typically: mechanical; proof by induction on F)

Induction for functions

- every $f : X \to Y$ gives rise to graph relation $y = f\,x$
- recursive f may be simulated by an inductive $F\,x\,y$
  - (partial correctness) **soundness**

    $$\mathrm{snd}_f(F) : (\Pi x : X)\,(\Pi y : Y)\,F\,x\,y \to (y = f\,x)$$

    (typically: mechanical; proof by induction on F)
  - (totality) **completeness**

    $$\mathrm{cmp}_f(F) : (\Pi x : X)\,(\Pi y : Y)\,(y = f\,x) \to F\,x\,y$$

    alternatively, by appeal to J

    $$\mathrm{cmp}_f(F) : (\Pi x : X)\,F\,x\,(f\,x)$$

    (typically: not mechanical; proof by induction on the data x)

Abstraction principle

- in proof (elimination): replace induction on lists with induction on graph; definitional equalities encapsulated in instantiation of inductive premises;

- in specification (introduction/definition): reduce fold induction to datatype induction; definitional equalities justify constructors (axioms, inference rules) of graph.

cf.

- Bove-Capretta (1999): termination of non-structural recursion via domain predicates
- Bertot-Magaud (2000): Changement de représentation des données
- McBride-McKinna (2004): The View from the Left

Implemented instances

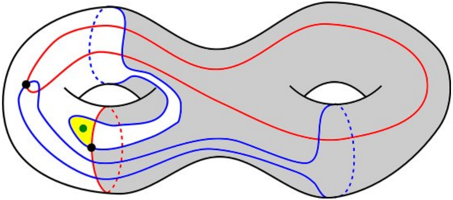
- NQTHM/ACL2: Boyer-Moore "recursion analysis".
- HOL: TFP (Slind); Krauss *et al.*.
- COQ: `Function` (Forest *et al.*), `Program`, `Equations` (Sozeau); esp. for non-structural recursion.
- EPIGRAM: native support for views (soundness built in); have to write programs witnessing views (proofs of completeness) by hand.
- AGDA, IDRIS: (so far) need to proceed entirely by hand.

Idea: extend the technique to implementations of HoTT.

Homotopy Type Theory

Synthetic homotopy theory via Type Theory

- New interpretation of Martin-Löf Type Theory into (abstract) homotopy theory.



- Intuitively:
  - Types $\rightsquigarrow$ spaces.
  - $a : A \rightsquigarrow$ points of A.
  - Identity type $a =_A b \rightsquigarrow$ space of paths from a to b in A.
- Univalence Axiom: equality of types is homotopy equivalence.
- Logical methods capture homotopical concepts; synthetic homotopy theory.
- Getting closer to a well-behaved implementation (CUBICALTT, Coquand *et al.*).

Higher inductive types

- Other logical ideas are also suggested by the homotopy interpretation.
- Higher inductive types: generated by both point and (higher) path constructors.
- E.g. circle $\mathbb{S}^1$ generated by

$$\mathsf{base} : \mathbb{S}^1$$
$$\mathsf{loop} : \mathsf{base} = \mathsf{base}$$

Higher inductive types

- Other logical ideas are also suggested by the homotopy interpretation.

- Higher inductive types: generated by both point and (higher) path constructors.

- E.g. circle $\mathbb{S}^1$ generated by

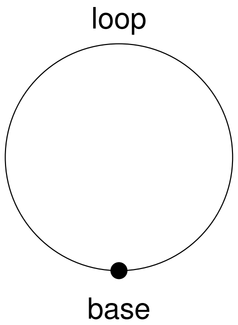
$$\mathsf{base} : \mathbb{S}^1$$
$$\mathsf{loop} : \mathsf{base} = \mathsf{base}$$

base

Higher inductive types

- Other logical ideas are also suggested by the homotopy interpretation.
- Higher inductive types: generated by both point and (higher) path constructors.
- E.g. circle $\mathbb{S}^1$ generated by

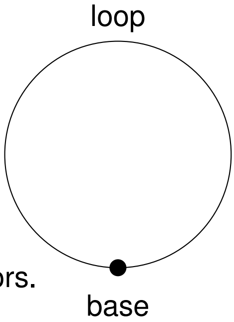
$$\begin{aligned} \mathsf{base} &: \mathbb{S}^1 \\ \mathsf{loop} &: \mathsf{base} = \mathsf{base} \end{aligned}$$



Higher inductive types

- Other logical ideas are also suggested by the homotopy interpretation.
- Higher inductive types: generated by both point and (higher) path constructors.
- E.g. circle $\mathbb{S}^1$ generated by

$$\mathsf{base} : \mathbb{S}^1$$
$$\mathsf{loop} : \mathsf{base} = \mathsf{base}$$



- Eliminator must respect/act on higher constructors.
- Proofs are more subtle; blind approach not very useful.

Proving homotopy equivalences

Proving

$$f : A \simeq B : g$$

becomes: construct inhabitants of

$$(\Pi b : B)\ f\,(g\,b) = b$$

$$(\Pi a : A)\ g\,(f\,a) = a$$

Actual use case: the encode-decode method

$$e_x : P(x) \simeq C(x) : d_x$$

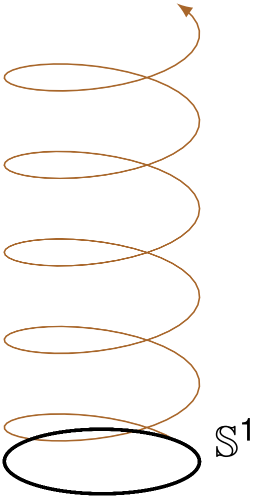
where:

- $x : T$ for HIT T,
- $P(x) \equiv$ path space, defined in terms of equality,
- $C(x) \equiv$ covering space, defined by HIT-recursion and the univalence axiom.

Example for showing $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$

Here:

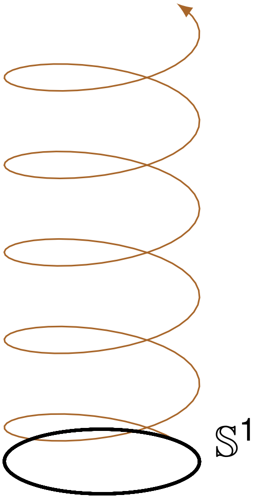
- $T \equiv \mathbb{S}^1$
- $P(x) \equiv \mathsf{base} = x$
- $C(x)$ given by
  - $C(\mathsf{base}) \equiv \mathbb{Z}$
  - $C(\mathsf{loop}) : C(\mathsf{base}) = C(\mathsf{base}) \equiv \mathsf{ua}(\mathsf{succ})$
  - *i.e.* $\mathsf{loop}^{\star}_{C} z \equiv \mathsf{succ}\, z$.



Example for showing $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$

Here:

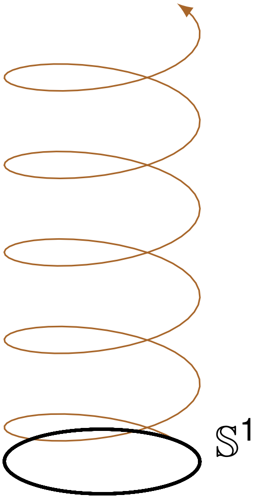
- $T \equiv \mathbb{S}^1$
- $P(x) \equiv \mathsf{base} = x$
- $C(x)$ given by
  - $C(\mathsf{base}) \equiv \mathbb{Z}$
  - $C(\mathsf{loop}) : C(\mathsf{base}) = C(\mathsf{base}) \equiv \mathsf{ua}(\mathsf{succ})$
  - *i.e.* $\mathsf{loop}^{\star}_{C} z \equiv \mathsf{succ}\, z$.
- for $p : P(x)$, $\mathsf{e}_x p \equiv p^{\star}_{C} 0$.



Example for showing $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$

Here:

- $T \equiv \mathbb{S}^1$
- $P(x) \equiv \mathsf{base} = x$
- $C(x)$ given by
  - $C(\mathsf{base}) \equiv \mathbb{Z}$
  - $C(\mathsf{loop}) : C(\mathsf{base}) = C(\mathsf{base}) \equiv \mathsf{ua}(\mathsf{succ})$
  - *i.e.* $\mathsf{loop}^{\star}_{C} z \equiv \mathsf{succ}\, z$.
- for $p : P(x)$, $\mathsf{e}_x p \equiv p^{\star}_{C} 0$.
- for $c : C(x)$, $\mathsf{d}_x c$ given by
  - $\mathsf{d}_{\mathsf{base}} \equiv z \mapsto \mathsf{loop}^z$
  - $\mathsf{d}_{\mathsf{loop}} : \mathsf{loop}^{\star}(\mathsf{d}_{\mathsf{base}}) = \mathsf{d}_{\mathsf{base}}$
  - given by a translation-invariance lemma.

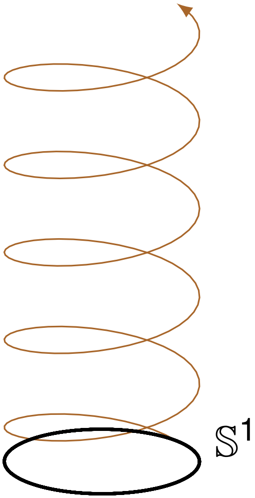


Example for showing $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$

Here:

- $T \equiv \mathbb{S}^1$
- $P(x) \equiv \mathsf{base} = x$
- $C(x)$ given by
  - $C(\mathsf{base}) \equiv \mathbb{Z}$
  - $C(\mathsf{loop}) : C(\mathsf{base}) = C(\mathsf{base}) \equiv \mathsf{ua}(\mathsf{succ})$
  - *i.e.* $\mathsf{loop}^{\star}_{C} z \equiv \mathsf{succ}\, z$.
- for $p : P(x)$, $\mathsf{e}_x p \equiv p^{\star}_{C} 0$.
- for $c : C(x)$, $\mathsf{d}_x c$ given by
  - $\mathsf{d}_{\mathsf{base}} \equiv z \mapsto \mathsf{loop}^z$
  - $\mathsf{d}_{\mathsf{loop}} : \mathsf{loop}^{\star}(\mathsf{d}_{\mathsf{base}}) = \mathsf{d}_{\mathsf{base}}$
  - given by a translation-invariance lemma.
- Prove $e_x = d_x^{-1}$ (tricky!) and conclude:

$$\Omega(\mathbb{S}^1, \mathsf{base}) :\equiv (\mathsf{base} = \mathsf{base}) \equiv P(\mathsf{base}) \simeq C(\mathsf{base}) \equiv \mathbb{Z}$$



Proving $e_x = d_x^{-1}$ in terms of graphs

Introduce graphs, for $x : \mathbb{S}^1$

$$E_x : P(x) \to C(x) \to \mathcal{U}$$
$$D_x : C(x) \to P(x) \to \mathcal{U}$$

with, for all $x : \mathbb{S}^1$

$$\mathrm{snd}_{e_x}(E_x) : (\Pi p : P(x))\,(\Pi c : C(x))\; E_x\,p\,c \to (c = e_x\,p)$$
$$\mathrm{cmp}_{e_x}(E_x) : (\Pi p : P(x))\,(\Pi c : C(x))\; (c = e_x\,p) \to E_x\,p\,c$$
$$\mathrm{snd}_{d_x}(D_x) : (\Pi c : C(x))\,(\Pi p : P(x))\; D_x\,c\,p \to (p = d_x\,c)$$
$$\mathrm{cmp}_{d_x}(D_x) : (\Pi c : C(x))\,(\Pi p : P(x))\; (p = d_x\,c) \to D_x\,c\,p$$

Finally, prove

$$(\dagger)\;(\Pi x : \mathbb{S}^1)\,(\Pi p : P(x))\,(\Pi c : C(x))\; E_x\,p\,c \Leftrightarrow D_x\,c\,p$$

The encode-decode equivalence

The equivalence $e_x : P(x) \simeq C(x) : d_x$ now follows:

$$\mathrm{E}_x\, p\, (\mathrm{e}_x(p)) \qquad \text{by} \quad \mathrm{cmp}_{\mathrm{e}_x}(\mathrm{E}_x)$$

The encode-decode equivalence

The equivalence $e_x : P(x) \simeq C(x) : d_x$ now follows:

$$\begin{array}{lll} \mathsf{E}_x\, p\, (\mathsf{e}_x(p)) & \text{by} & \mathsf{cmp}_{\mathsf{e}_x}(\mathsf{E}_x) \\ \mathsf{D}_x\, (\mathsf{e}_x(p))\, p & \text{by} & (\dagger) \end{array}$$

The encode-decode equivalence

The equivalence $e_x : P(x) \simeq C(x) : d_x$ now follows:

$$\begin{array}{lll} \mathsf{E}_x\, p\, (\mathsf{e}_x(p)) & \text{by} & \mathsf{cmp}_{\mathsf{e}_x}(\mathsf{E}_x) \\ \mathsf{D}_x\, (\mathsf{e}_x(p))\, p & \text{by} & (\dagger) \\ \mathsf{d}_x(\mathsf{e}_x(p)) = p & \text{by} & \mathsf{snd}_{\mathsf{d}_x}(\mathsf{D}_x) \end{array}$$

The encode-decode equivalence

The equivalence $e_x : P(x) \simeq C(x) : d_x$ now follows:

$$\begin{array}{lll} \mathsf{E}_x\, p\, (\mathsf{e}_x(p)) & \text{by} & \mathsf{cmp}_{\mathsf{e}_x}(\mathsf{E}_x) \\ \mathsf{D}_x\, (\mathsf{e}_x(p))\, p & \text{by} & (\dagger) \\ \mathsf{d}_x(\mathsf{e}_x(p)) = p & \text{by} & \mathsf{snd}_{\mathsf{d}_x}(\mathsf{D}_x) \end{array}$$

The other direction is entirely symmetric.

Note No explicit equational reasoning!

Also Note Each step is logical equivalence, homotopy equivalences not needed for argument.

Logical equivalence vs homotopy equivalence

- By soundness and completeness, we get a logical equivalence

$$F\,x\,y \Leftrightarrow (y = f\,x)$$

- Can this be improved to a homotopy equivalence

$$F\,x\,y \simeq (y = f\,x)?$$

Logical equivalence vs homotopy equivalence

- By soundness and completeness, we get a logical equivalence

$$F\,x\,y \Leftrightarrow (y = f\,x)$$

- Can this be improved to a homotopy equivalence

$$F\,x\,y \simeq (y = f\,x)?$$

- Yes, if *snd* and *cmp* are coherent in a suitable way:

$$\mathrm{coh}_f(F) : \mathrm{transport}_{F\,x}(snd\,p)(p) = cmp\,x$$

(*cf.* HoTT Book Issue #718 [for $f = \mathrm{id}$], Rijke/Escardó).

- Can usually be proven for the inductively defined graph.

Logical equivalence vs homotopy equivalence

- By soundness and completeness, we get a logical equivalence

$$F\,x\,y \Leftrightarrow (y = f\,x)$$

- Can this be improved to a homotopy equivalence

$$F\,x\,y \simeq (y = f\,x)?$$

- Yes, if *snd* and *cmp* are coherent in a suitable way:

$$\mathrm{coh}_f(F) : \mathrm{transport}_{F x}(\mathit{snd}\,p)(p) = \mathit{cmp}\,x$$

(*cf.* HoTT Book Issue #718 [for $f = \mathrm{id}$], Rijke/Escardó).

- Can usually be proven for the inductively defined graph.

- However…

this is not what we are doing!

Idea: how to prove (†) (*cf.* Bertot/Magaud)

- (†) is an equivalence of specifications
- $F\ x\ y \simeq (y = f\ x)$ is one way to proceed, not the only one!
- for suitable choices of D, E, (†) becomes easy or even vacuous to prove
  - (easy) by (higher) induction on D, E; not necessarily a homotopy equivalence
  - (vacuous) take $\mathrm{D}_x\ p\ c \equiv \mathrm{E}_x\ c\ p$ (!)
- Difficulty moves into proofs of completeness.

Choices and tradeoffs

	inductive E inductive D	inductive E $D\,c\,p \equiv E\,p\,c$	inductive E HIT D
$\mathrm{snd}_{\mathrm{e}_x}(\mathrm{E}_x)$ $\mathrm{cmp}_{\mathrm{e}_x}(\mathrm{E}_x)$ $\mathrm{coh}_{\mathrm{e}_x}(\mathrm{E}_x)$	mechanical	mechanical	mechanical
$\mathrm{snd}_{\mathrm{d}_x}(\mathrm{D}_x)$	easy	induction	mechanical
$\mathrm{cmp}_{\mathrm{d}_x}(\mathrm{D}_x)$	impossible?	induction + $\mathbb{Z}$ is a set	
$(\dagger) D \Leftrightarrow E$	easy induction	vacuous	hard

$\mathbb{Z}$ is a set!

- Because $\mathbb{Z}$ has decidable equality, it has trivial higher structure by Hedberg's Theorem.
- For all $p, q : x =_{\mathbb{Z}} y$, we have $p = q$.
- In the terminology of HoTT, $\mathbb{Z}$ is a set.
- By soundness, coherence and the univalence axiom, $E_{\mathsf{base}}\, p\, c = (\mathsf{e}_x(p) =_{\mathbb{Z}} c)$.
- Hence also $E_{\mathsf{base}}\, p\, c$ is trivial.
- In particular $\mathsf{loop}^{\star} e = e$ for all $e : E_{\mathsf{base}}\, p\, c$.
- This makes HIT-induction respecting paths vacous!

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathrm{cmp}_{D_x}(\mathsf{d}_x) : (\Pi x : \mathbb{S}^1)\ (\Pi c : C(x))\ E_x\, (\mathsf{d}_x c)\, c$$

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathrm{cmp}_{D_x}(\mathrm{d}_x) : (\Pi x : \mathbb{S}^1)\, (\Pi c : C(x))\, E_x\, (\mathrm{d}_x c)\, c$$

- which by HIT-induction on $\mathbb{S}^1$, and the above observation reduces to

$$(\Pi c : C(\mathrm{base}))\, E_{\mathrm{base}}\, (\mathrm{d}_{\mathrm{base}} c)\, c$$

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathsf{cmp}_{D_x}(\mathsf{d}_x) : (\Pi x : \mathbb{S}^1)\ (\Pi c : C(x))\ E_x\,(\mathsf{d}_x c)\, c$$

- which by HIT-induction on $\mathbb{S}^1$, and the above observation reduces to

$$(\Pi z : \mathbb{Z})\ E_{\mathsf{base}}\,(\mathsf{d}_{\mathsf{base}} z)\, z$$

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathsf{cmp}_{D_x}(\mathsf{d}_x) : (\Pi x : \mathbb{S}^1)\ (\Pi c : C(x))\ E_x\,(\mathsf{d}_x c)\, c$$

- which by HIT-induction on $\mathbb{S}^1$, and the above observation reduces to

$$(\Pi z : \mathbb{Z})\ E_{\mathsf{base}}\,(\mathsf{loop}^z)\, z$$

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathsf{cmp}_{D_x}(\mathsf{d}_x) : (\Pi x : \mathbb{S}^1)\ (\Pi c : C(x))\ E_x\, (\mathsf{d}_x c)\, c$$

- which by HIT-induction on $\mathbb{S}^1$, and the above observation reduces to

$$(\Pi z : \mathbb{Z})\ E_{\mathsf{base}}\, (\mathsf{loop}^z)\, z$$

- by completeness for E, this reduces to

$$(\Pi z : \mathbb{Z})\ (\mathsf{loop}^z)^{\star} 0 = z$$

which is easily proven by (normal) induction on $z : \mathbb{Z}$.

Completeness because $\mathbb{Z}$ is a set

- For $D_x\, c\, p \equiv E_x\, p\, c$, we need

$$\mathsf{cmp}_{D_x}(\mathsf{d}_x) : (\Pi x : \mathbb{S}^1)\ (\Pi c : C(x))\ E_x\, (\mathsf{d}_x c)\, c$$

- which by HIT-induction on $\mathbb{S}^1$, and the above observation reduces to

$$(\Pi z : \mathbb{Z})\ E_{\mathsf{base}}\, (\mathsf{loop}^z)\, z$$

- by completeness for E, this reduces to

$$(\Pi z : \mathbb{Z})\ (\mathsf{loop}^z)^\star 0 = z$$

  which is easily proven by (normal) induction on $z : \mathbb{Z}$.

- No (non-trivial) HIT-induction needed to prove $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$!


Summary

Summary

- Burstall's insight: replace proofs relying on reduction behaviour of functions by proofs by induction over the graph of the function.
- By choosing a clever encoding of the graph, we can get away with less work.
- Work in progress: hopefully scales to more complicated encode-decode proofs.

Summary

- Burst our of funct on.
- By ch ay with less
- Work enco

Thanks!