Encode-decode à la Burstall

James McKinna
jww Fred Nordvall-Forsberg

LFCS, University of Edinburgh

TYPES @ Tallinn
Wednesday, 20 May 2015

an exercise in **using**
(homotopy) type theory

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 $F : [A] \to B \to \mathcal{U}$,

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

Proof Induction on $as : [A]$

Induction for functions

- every $f : X \to Y$ gives rise to **graph relation** $y = f\,x$
- **recursive** f may be simulated by an **inductive** $F\,x\,y$
  - (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: (so far) need to proceed entirely by hand
Idea: extend the technique to implementations of HoTT

HITs in HoTT

- HoTT:
  - (logical) equality now has homotopic/geometric interpretation, as **path space**
  - **univalence axiom**: equality between types is **homotopy equivalence**
- HITs: higher inductive types
  - still inductively defined: **homotopy**-initial algebras (Sojakova)
  - higher constructors yield **new** equalities
  - induction principle (motive) is **constrained** to respect these
- proofs are more subtle; synthetic reasoning in homotopy theory is...

magic

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: 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 univalence axiom

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

Here:

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

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

In terms of graphs

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

- $E_x : \Pi_{p:P(x)} \Pi_{c:C(x)} \mathcal{U}$
- $D_x : \Pi_{c:C(x)} \Pi_{p:P(x)} \mathcal{U}$

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

- $\mathsf{snd}_{e_x}(E_x)$: $\Pi_{p:P(x)} \Pi_{c:C(x)} E_x\, p\, c \to (c = e_x\, p)$
- $\mathsf{cmp}_{e_x}(E_x)$: $\Pi_{p:P(x)} \Pi_{c:C(x)} (c = e_x\, p) \to E_x\, p\, c$
- $\mathsf{snd}_{d_x}(D_x)$: $\Pi_{c:C(x)} \Pi_{p:P(x)} D_x\, c\, p \to (p = d_x\, c)$
- $\mathsf{cmp}_{d_x}(D_x)$: $\Pi_{c:C(x)} \Pi_{p:P(x)} (p = d_x\, c) \to D_x\, c\, p$

and homotopy condition becomes

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

The encode-decode equivalence

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

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

The other direction is entirely symmetric.

NB. No explicit equational reasoning!

HoTT Book Issue #718: Rijke/Escardó

Theorem (*cf.* Thm 7.2.2; **change of base generalisation**)
If

- **soundness**, snd : $\Pi_{x:X}\,\Pi_{y:Y}\,F\,x\,y \to (y = f\,x)$
- **completeness**,
  - (Rijke) cmp : $\Pi_{x:X}\,F\,x\,(f\,x)$
  - (Escardó) cmp : $\Pi_{x:X}\,\Pi_{y:Y}\,(y = f\,x) \to F\,x\,y$

AND

- **coherence**: for $x : X, y : Y, p : F\,x\,y$
  - (Rijke) coh : $\mathrm{transport}_{Fx}(\mathrm{snd}\ p)(p) = \mathrm{cmp}\ x$
  - (Escardó) coh : $\mathrm{cmp}\,(\mathrm{snd}\ p) = p$

Then

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

Proof

- (Rijke) $\Sigma_{y:Y}\,F\,x\,y$ is contractible, with centre $(f\,x, \mathrm{cmp}\ x)$
- (Escardó) $q : (y = f\,x) \mapsto \mathrm{snd}\,(\mathrm{cmp}\ q)$ is idempotent, hence the identity

this is NOT what we are doing. . . because

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

- (†) is an **equivalence of specifications**
- Rijke-Escardó offers **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 $\mathsf{D}_x\, p\, c \equiv \mathsf{E}_x\, c\, p$ (!)
- difficulty moves into proofs of completeness

no royal road to synthetic homotopy theory

Choices and tradeoffs

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

$\mathbb{Z}$ is a set!

Because $\mathbb{Z}$ is a set,

- for any $e : E_{\text{base}}\, p\, c$
- by soundness and coherence, we have $\text{loop}^\star e = e$

Thus

- $\text{cmp}_{D_x}(\text{d}_x)$
- becomes $\Pi_{x:\mathbb{S}^1}\ \Pi_{c:C(x)}\ E_x\,(\text{d}_x c)\, c$
- which by HIT-induction on $\mathbb{S}^1$, and the above observation
- reduces to $\Pi_{c:C(\text{base})}\ E_{\text{base}}\,(\text{d}_{\text{base}} c)\, c$
- that is, $\Pi_{z:\mathbb{Z}}\ E_{\text{base}}\,(\text{d}_{\text{base}} z)\, z$
- by completeness for E, this is $\Pi_{z:\mathbb{Z}}\ (\text{loop}^z)^\star 0 = z$

NB. this is a **different** argument to the standard one, but easy by $\mathbb{Z}$-induction

Conclusions, Questions, Further Work

Mechanisation

- Ornamentation: so far, we have only considered graphs of **folds**; relate to McBride/Dagand algebraic ornaments?
- Inductive families: we have presented these informally, without paying close attention to the meaning and representation of such things in HoTT; explicit equational premises might lead to further insight, further complexity
- Work in progress: yet to complete our Agda formalisations, and there are certainly (many) devils in the detail!
- Cavallo: consider identities of the form
  $\Pi_{x:X}\ \Pi_{b:B(x)}\ f_x\ (g_x\ b) = h_x(b)$
  arising from Mayer-Vietoris *etc.*

Questions?