Induction-induction Part 1

Thorsten Altenkirch[1] Ambrus Kaposi[2] András Kovács[2]
Jakob von Raumer[1]

University of Nottingham, United Kingdom

Eötvös Loránd University, Budapest, Hungary

July 1, 2018

Indexed Inductive Types

- Provers like Lean & Coq are built on *indexed inductive Types* (e. g. Vec)
- Easy to reduce *mutual inductive Types* to this class of Types:

mutual inductive even, odd

with even : $\mathbb{N} \to$ **Prop**
| even_zero : even 0
| even_odd : Π n, odd n $\to$ even (n $+$ 1)

with odd : $\mathbb{N} \to$ **Prop**
| odd_even : Π n, even n $\to$ odd (n $+$ 1)

Indexed Inductive Types

- Provers like Lean & Coq are built on *indexed inductive Types* (e. g. Vec)
- Easy to reduce *mutual inductive Types* to this class of Types:

inductive par : bool $\to \mathbb{N} \to$ **Prop**
| even_zero : par ff 0
| even_odd : Π n, par tt n $\to$ par ff (n + 1)
| odd_even : Π n, par ff n $\to$ par tt (n + 1)

def even := par ff

def odd := par tt

Example: Dense Completion

(due to Fredrik Nordvall Forsberg)
Given a type $A : \mathcal{U}$ and a relation $<: A \to A \to \mathcal{U}$ on it, the *free dense completion* is a new set A' together with a relation $<'$ on A that contains $<$ which is *dense*:
For each $x, y : A'$ with $x <' y$, there is a point mid : A' such that $x <' \mathrm{mid}(x, y) <' y$.

Example: Dense Completion

parameters $(\mathsf{A} : \textbf{Type})\ (\mathsf{lt} : \mathsf{A} \to \mathsf{A} \to \textbf{Type})$

inductive $\mathsf{A}', \mathsf{lt}'$
with $\mathsf{A}' : \textbf{Type}$
$\mid \iota_{\mathsf{A}} : \mathsf{A} \to \mathsf{A}'$
$\mid \mathsf{mid} : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'\ \mathsf{x}\ \mathsf{y}),\ \mathsf{A}'$
with $\mathsf{lt}' : \mathsf{A}' \to \mathsf{A}' \to \textbf{Type}$
$\mid \iota_{\mathsf{lt}} : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}),\ \mathsf{lt}\ \mathsf{x}\ \mathsf{y} \to \mathsf{lt}'\ (\iota_{\mathsf{A}}\ \mathsf{x})\ (\iota_{\mathsf{A}}\ \mathsf{y})$
$\mid \mathsf{mid}_l : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'\ \mathsf{x}\ \mathsf{y}),\ \mathsf{lt}'\ \mathsf{x}\ (\mathsf{mid}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})$
$\mid \mathsf{mid}_r : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'\ \mathsf{x}\ \mathsf{y}),\ \mathsf{lt}'\ (\mathsf{mid}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})\ \mathsf{y}$

Example: Dense Completion

Consider the *category of algebras* for the type:

```
structure Alg :=
  (A'    : Type u)
  (lt'   : A' → A' → Type u)
  (ι_A   : A → A')
  (mid   : Π x y, lt' x y → A')
  (ι_lt  : Π a b, lt a b → lt' (ι_A a) (ι_A b))
  (mid_l : Π x y p, lt' x (mid x y p))
  (mid_r : Π x y p, lt' (mid x y p) y)

structure Alg_m (M N : Alg) :=
  (A' : M.A' → N.A')
  (lt' : Π x y, M.lt' x y → N.lt' (A' x) (A' y))
  (ι_A : Π a, A' (M.ι_A a) = N.ι_A a)
  ...
```

We want to construct an initial object in this category to get the correct elimination principle!

Example: Dense Completion

We will construct this algebra and prove its initiality in eight steps:

1. Erase the typing relation
2. Construct the initial pre-algebra
3. Construct a welltypedness predicate
4. Define the inital algebra
5. Relate objects of the pre-algebra to arbitrary algebras
6. Show right-uniqueness of the relation
7. Show left-totality of the relation
8. Extract the eliminators from the relation

Example: Dense Completion

Step 1: Erasure of the typing relation:

structure Pre $:=$
  $(\mathsf{A}' : \textbf{Type}\ \mathsf{u})$
  $(\mathsf{lt}' : \textbf{Type}\ \mathsf{u})$
  $(\iota_{\mathsf{A}} : \mathsf{A} \to \mathsf{A}')$
  $(\mathsf{mid} : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'),\ \mathsf{A}')$
  $(\iota_{\mathsf{lt}} : \Pi\ (\mathsf{a}\ \mathsf{b} : \mathsf{A})\ (\mathsf{p} : \mathsf{lt}\ \mathsf{a}\ \mathsf{b}),\ \mathsf{lt}')$
  $(\mathsf{mid}_l : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'),\ \mathsf{lt}')$
  $(\mathsf{mid}_r : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'),\ \mathsf{lt}')$

These *pre-algebras* form a category as well.

Example: Dense Completion

Step 2: The initial object of these is just a mutual inductive definition!

mutual inductive A', lt'

with A' : **Type** u
| ι_{A} : $\mathsf{A} \to \mathsf{A}'$
| mid : Π (x y : A') (p : lt'), A'

with lt' : **Type** u
| ι_{lt} : Π (a b : A) (p : lt a b), lt'
| mid_l : Π (x y : A') (p : lt'), lt'
| mid_r : Π (x y : A') (p : lt'), lt'

But the typing information of lt' is missing.

Example: Dense Completion

Step 3: Adding a welltypedness predicate:

mutual inductive $\mathsf{w}_{\mathsf{A'}}$, $\mathsf{w}_{\mathsf{lt'}}$

with $\mathsf{w}_{\mathsf{A'}} : \mathsf{A}' \to$ **Prop**
$|\ \iota_{\mathsf{A}} : \Pi\ \mathsf{a},\ \mathsf{w}_{\mathsf{A'}}\ (\mathsf{A}'.\iota_{\mathsf{A}}\ \mathsf{a})$
$|\ \mathsf{mid} : \Pi\ \{\mathsf{x}\ \mathsf{y}\ \mathsf{p}\},\ \mathsf{w}_{\mathsf{A'}}\ \mathsf{x} \to \mathsf{w}_{\mathsf{A'}}\ \mathsf{y} \to \mathsf{w}_{\mathsf{lt'}}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p} \to \mathsf{w}_{\mathsf{A'}}\ (\mathsf{A}'.\mathsf{mid}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})$

with $\mathsf{w}_{\mathsf{lt'}} : \mathsf{A}' \to \mathsf{A}' \to \mathsf{lt}' \to$ **Prop**
$|\ \iota_{\mathsf{lt}} : \Pi\ \mathsf{a}\ \mathsf{b}\ \mathsf{p},\ \mathsf{w}_{\mathsf{lt'}}\ (\mathsf{A}'.\iota_{\mathsf{A}}\ \mathsf{a})\ (\mathsf{A}'.\iota_{\mathsf{A}}\ \mathsf{b})\ (\mathsf{lt}'.\iota_{\mathsf{lt}}\ \mathsf{a}\ \mathsf{b}\ \mathsf{p})$
$|\ \mathsf{mid}_l : \Pi\ \{\mathsf{x}\ \mathsf{y}\ \mathsf{p}\},\ \mathsf{w}_{\mathsf{A'}}\ \mathsf{x} \to \mathsf{w}_{\mathsf{A'}}\ \mathsf{y} \to \mathsf{w}_{\mathsf{lt'}}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p}$
$\quad \to \mathsf{w}_{\mathsf{lt'}}\ \mathsf{x}\ (\mathsf{A}'.\mathsf{mid}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})\ (\mathsf{lt}'.\mathsf{mid}_l\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})$
...

Example: Dense Completion

Step 4: Define the initial object of the typed algebras as a subtype of the initial untyped algebra:

def S : Alg $:=$
{ $\mathsf{A}' := \Sigma'$ (x : A'), $\mathsf{w}_{\mathsf{A}'}$ x ,
  $\mathsf{lt}' := \lambda$ x y, Σ' p, $\mathsf{w}_{\mathsf{lt}'}$ x.1 y.1 p,
  $\iota_{\mathsf{A}} := \lambda$ x, $\langle \mathsf{A}'.\iota_{\mathsf{A}}$ x, $\mathsf{w}_{\mathsf{A}'}.\iota_{\mathsf{A}}$ x$\rangle$,
  mid $:= \lambda$ a b p, $\langle \mathsf{A}'$.mid a.1 b.1 p.1, $\mathsf{w}_{\mathsf{A}'}$.mid a.2 b.2 p.2$\rangle$,
  ... }

Are we done yet? Have to define the eliminator!

Example: Dense Completion

Step 5: For any algebra M : Alg, inductively define a relation that relates objects of the untyped S' : Alg to those of M:

mutual inductive $\mathsf{r}_{\mathsf{A}'}$, $\mathsf{r}_{\mathsf{lt}'}$

with $\mathsf{r}_{\mathsf{A}'} : \mathsf{A}' \to \mathsf{M.A}' \to$ **Prop**
$| \ \iota_{\mathsf{A}} : \Pi\ \mathsf{x},\ \mathsf{r}_{\mathsf{A}'}\ (\mathsf{A}'.\iota_{\mathsf{A}}\ \mathsf{x})\ (\mathsf{M}.\iota_{\mathsf{A}}\ \mathsf{x})$
$| \ \mathsf{mid} : \Pi\ \mathsf{x}\ \mathsf{y}\ \mathsf{p}\ \alpha\ \beta\ \pi\ (\mathsf{xr} : \mathsf{r}_{\mathsf{A}'}\ \mathsf{x}\ \alpha)\ (\mathsf{yr} : \mathsf{r}_{\mathsf{A}'}\ \mathsf{y}\ \beta)\ (\mathsf{pr} : \mathsf{r}_{\mathsf{lt}'}\ \mathsf{p}\ \pi),$
$\quad \mathsf{r}_{\mathsf{A}'}\ (\mathsf{A}'.\mathsf{mid}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p})\ (\mathsf{M.mid}\ \alpha\ \beta\ \pi)$

with $\mathsf{r}_{\mathsf{lt}'} : \Pi\ (\mathsf{x} : \mathsf{lt}')\ \{\alpha\ \beta : \mathsf{M.A}'\}\ (\pi : \mathsf{M.lt}'\ \alpha\ \beta),$ **Prop**
$| \ \iota_{\mathsf{lt}} : \Pi\ \mathsf{a}\ \mathsf{b}\ \mathsf{p},\ \mathsf{r}_{\mathsf{lt}'}\ (\mathsf{lt}'.\iota_{\mathsf{lt}}\ \mathsf{a}\ \mathsf{b}\ \mathsf{p})\ (\mathsf{M}.\iota_{\mathsf{lt}}\ \mathsf{a}\ \mathsf{b}\ \mathsf{p})$
...

Example: Dense Completion

Step 6: Show, by induction on the untyped object, that the relation is right unique:

mutual def $\mathsf{r\_unique}_{\mathsf{A}'}$, $\mathsf{r\_unique}_{\mathsf{lt}'}$

with $\mathsf{r\_unique}_{\mathsf{A}'}$: Π $\{\mathsf{x} : \mathsf{A}'\}$ $\{\alpha\ \alpha' : \mathsf{M.A}'\}$
$(\mathsf{r}\alpha : \mathsf{r}_{\mathsf{A}'}\ \mathsf{x}\ \alpha)$ $(\mathsf{r}\alpha' : \mathsf{r}_{\mathsf{A}'}\ \mathsf{x}\ \alpha')$, $\alpha = \alpha'$
...

with $\mathsf{r\_unique}_{\mathsf{lt}'}$: Π $\{\mathsf{p} : \mathsf{lt}'\}$ ... $\{\pi : \mathsf{M.lt}'\ \alpha\ \beta\}$ $\{\pi' : \mathsf{M.lt}'\ \alpha'\ \beta'\}$
$(\mathsf{r}\pi : \mathsf{r}_{\mathsf{lt}'}\ \mathsf{p}\ \pi)$ $(\mathsf{r}\pi' : \mathsf{r}_{\mathsf{lt}'}\ \mathsf{p}\ \pi')$, $\pi = \pi'$
...

Example: Dense Completion

Step 7: Show that the relation is left total on welltyped objects and thus a function:

mutual def $\mathsf{r\_ex}_{\mathsf{A'}}$, $\mathsf{r\_ex}_{\mathsf{lt'}}$

with $\mathsf{r\_ex}_{\mathsf{A'}} : \Pi\ (\mathsf{x} : \mathsf{A}'),\ \mathsf{w}_{\mathsf{A'}}\ \mathsf{x}$
$\quad \to \Sigma'\ \alpha,\ \mathsf{r}_{\mathsf{A'}}\ \mathsf{x}\ \alpha$
...

with $\mathsf{r\_ex}_{\mathsf{lt'}} : \Pi\ (\mathsf{x}\ \mathsf{y} : \mathsf{A}')\ (\mathsf{p} : \mathsf{lt}'),\ \mathsf{w}_{\mathsf{A'}}\ \mathsf{x} \to \mathsf{w}_{\mathsf{A'}}\ \mathsf{y} \to \mathsf{w}_{\mathsf{lt'}}\ \mathsf{x}\ \mathsf{y}\ \mathsf{p}$
$\quad \to \Sigma'\ \alpha\ \beta\ (\pi : \mathsf{M.lt}'\ \alpha\ \beta),\ \mathsf{r}_{\mathsf{A'}}\ \mathsf{x}\ \alpha \wedge \mathsf{r}_{\mathsf{A'}}\ \mathsf{y}\ \beta \wedge \mathsf{r}_{\mathsf{lt'}}\ \mathsf{p}\ \pi$
...

Example: Dense Completion

Step 8: From this, extract this actual eliminator, e. g.:

```
parameter (M : Alg)

def A′.elim : A′ → M.A′
| ⟨x, xw⟩ := (r_ex_A′ x xw).1

def lt′.elim : Π {x y}, lt′ x y → M.lt′ (A′.elim x) (A′.elim y) :=
...

theorem A′.mid.elim (x y : A′) (p : lt′ x y) :
  A′.elim (S.mid x y p) = M.mid (A′.elim x) (A′.elim y) (lt′.elim p) :=
...
```

Inductive-Inductive Types

Define inductively:

- A number of sorts $A, B, \ldots$ where later sorts can depend on positively on earlier sorts (e. g. $A : \mathcal{U}$, $B : A \to \mathcal{U}$),
- inductively, by giving constructors, such that constructors can positively refer to constructors of any sort.

Other important examples

- Syntax of a type theory (T.A., Ambrus Kaposi):

$$\begin{aligned} \mathsf{Con} &: \mathcal{U}, \\ \mathsf{Ty} &: \mathsf{Con} \to \mathcal{U} \\ \mathsf{Tm} &: \prod_{\Gamma:\mathsf{Con}} \mathsf{Ty}(\Gamma) \to \mathcal{U} \end{aligned}$$

- Cauchy real numbers (cf. HoTT book):

$$\begin{aligned} \mathbb{R} &: \mathcal{U}, \\ \_ \sim_{\_} \_ &: \mathbb{R} \to \mathbb{Q} \to \mathbb{R} \to \mathcal{U} \end{aligned}$$

Specifying IITs

- What is a correct specification of an inductive-inductive Type?
- Go with the approach seen in the talk “Constructing inductive-inductive types using a domain-specific type theory” by András Kovács
- Define type erasure and welltypeness predicates as transformations of contexts

Lean

- The theorem prover Lean features a meta language used to define tactics and other user defined commands.
- We started to implement the approach.
- Done so far: Automatic generation of the initial object itself and of the relation needed to construct the eliminator.
- The kernel API for inductive types is likely to improve in future versions of the prover.
- Goal: Full automation of the construction, outside of the kernel, but mostly hidden from the user.