Higher and/or dependent lenses

Nils Anders Danielsson

AIM XXII, Leuven, 2015-09-16

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement n° 247219. It has perhaps also received financial support from the Swedish Research Council, but I do not think that I am required to include a corresponding project number here. This presentation does not necessarily reflect the views of the ERC or the EU. The EU is not liable for any use of the presented information.

Introduction

$$
\begin{array}{l}
\textbf{record } R_1 : \mathit{Set} \textbf{ where} \\
\quad \textbf{field} \\
\qquad x : \mathit{Bool} \\
\qquad f : \mathit{Bool} \to \mathit{Bool} \\
\textbf{record } R_2 : \mathit{Set} \textbf{ where} \\
\quad \textbf{field} \\
\qquad r_1 : R_1 \\
\textbf{record } R_3 : \mathit{Set} \textbf{ where} \\
\quad \textbf{field} \\
\qquad r_2 : R_2
\end{array}
$$

Introduction

$$
\begin{aligned}
&\mathit{set\text{-}x}_1 \;:\; R_1 \;\rightarrow\; \mathit{Bool} \;\rightarrow\; R_1 \\
&\mathit{set\text{-}x}_1\; r\; x \;=\; \textbf{record}\; r\,\{x \;=\; x\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{set\text{-}x}_2 \;:\; R_2 \;\rightarrow\; \mathit{Bool} \;\rightarrow\; R_2 \\
&\mathit{set\text{-}x}_2\; r\; x \;=\; \textbf{record}\; r \\
&\qquad\qquad\qquad \{r_1 \;=\; \textbf{record}\;(R_2.r_1\; r) \\
&\qquad\qquad\qquad\qquad\qquad \{x \;=\; x\}\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{set\text{-}x}_3 \;:\; R_3 \;\rightarrow\; \mathit{Bool} \;\rightarrow\; R_3 \\
&\mathit{set\text{-}x}_3\; r\; x \;= \\
&\quad \textbf{record}\; r \\
&\qquad \{r_2 \;=\; \textbf{record}\;(R_3.r_2\; r) \\
&\qquad\qquad\qquad \{r_1 \;=\; \textbf{record}\;(R_2.r_1\;(R_3.r_2\; r)) \\
&\qquad\qquad\qquad\qquad\qquad \{x \;=\; x\}\}\}
\end{aligned}
$$

Introduction

With lenses:

$$
\begin{aligned}
x &: \mathit{Lens}\ R_1\ \mathit{Bool} \\
r_1 &: \mathit{Lens}\ R_2\ R_1 \\
r_2 &: \mathit{Lens}\ R_3\ R_2
\end{aligned}
$$

$$
\begin{aligned}
\mathit{set}\text{-}x_1 &: R_1 \rightarrow \mathit{Bool} \rightarrow R_1 \\
\mathit{set}\text{-}x_1 &= \mathit{set}\ x \\
\mathit{set}\text{-}x_2 &: R_2 \rightarrow \mathit{Bool} \rightarrow R_2 \\
\mathit{set}\text{-}x_2 &= \mathit{set}\ (x \circ r_1) \\
\mathit{set}\text{-}x_3 &: R_3 \rightarrow \mathit{Bool} \rightarrow R_3 \\
\mathit{set}\text{-}x_3 &= \mathit{set}\ (x \circ r_1 \circ r_2)
\end{aligned}
$$

Introduction

In this talk:

- What happens if we view lenses
  through the lens of homotopy type theory?
- What if we have dependent record types?

Note: Work in progress.

Preliminaries

H-levels

$$\mathit{H\text{-}level} : \mathbb{N} \to \mathit{Set} \to \mathit{Set}$$

$$\begin{aligned} \mathit{Is\text{-}proposition} &= \mathit{H\text{-}level}\ \mathit{1} \\ \mathit{Is\text{-}set} &= \mathit{H\text{-}level}\ \mathit{2} \end{aligned}$$

$$\mathit{Is\text{-}proposition}\ A \quad \Leftrightarrow \quad (x\ y : A) \to x \equiv y$$

$$\begin{aligned} &\mathit{Is\text{-}set}\ A \quad \Leftrightarrow \\ &\quad (x\ y : A) \to \mathit{Is\text{-}proposition}\,(x \equiv y) \end{aligned}$$

Propositional truncation

$$\|\_\| \ : \ Set \ \rightarrow \ Set$$
$$Is\text{-}proposition \, \| \ A \ \|$$
$$|\_| \quad : \ A \ \rightarrow \ \| \ A \ \|$$

Non-dependent eliminator:

$$Is\text{-}proposition \ B \ \rightarrow$$
$$(A \ \rightarrow \ B) \ \rightarrow$$
$$\| \ A \ \| \ \rightarrow \ B$$

Assumptions

Used in various proofs/definitions:

- The propositional truncation.
- Extensionality (used silently).
- The univalence axiom (UA).
- The K rule (K).

Assumptions

Used in various proofs/definitions:

- The propositional truncation.
- Extensionality (used silently).
- The univalence axiom (UA).
- The K rule (K).

TODO: Are K and $\|\_\|$ mutually consistent?

Equivalences

Equivalences:

$$\_\simeq\_ \; : \; Set \to Set \to Set$$

$A \simeq B$ is logically equivalent to
"A is in bijective correspondence with B".

Split surjections

Split surjections (functions with right inverses):

$$\_\twoheadrightarrow\_ \ : Set \rightarrow Set \rightarrow Set$$

Higher lenses

Traditional definition

Very well-behaved lenses:

$$
\begin{aligned}
&TLens : Set \to Set \to Set\\
&TLens\ A\ B =\\
&\quad (get : A \to B) \times\\
&\quad (set : A \to B \to A) \times\\
&\quad (\forall\ a\ b.\quad get\ (set\ a\ b) \equiv b) \times\\
&\quad (\forall\ a.\quad set\ a\ (get\ a) \equiv a) \times\\
&\quad (\forall\ a\ b_1\ b_2.\ set\ (set\ a\ b_1)\ b_2 \equiv set\ a\ b_2)
\end{aligned}
$$

Traditional definition

Can define id, $\_\circ\_$, can prove

$$\begin{aligned} id \circ l &\equiv l, \\ l \circ id &\equiv l, \\ l_1 \circ (l_2 \circ l_3) &\equiv (l_1 \circ l_2) \circ l_3, \end{aligned}$$

without assuming that domains or codomains are sets.

However, the last proof is rather long (at least my proof).

First definition using equivalences

A well-known fact (for *set-theoretic* presentations of lenses):

$$Lens\ A\ B \ \rightarrow\ \exists\ R\ :\ Set.\ A \leftrightarrow R \times\ B$$

Can we use this to define what a lens is?

$$Lens'\ A\ B\ =\ (R\ :\ Set) \times (A\ \simeq\ R \times\ B)$$

Getter, setter

Recall:

$$Lens' \; A \; B \; = \; (R \; : \; Set) \times (A \; \simeq \; R \times B)$$

Getter:

$$get\,(\_, eq) \; a \; = \; snd\,(to \; eq \; a)$$

Setter:

$$set\,(\_, eq) \; a \; b \; = \; from \; eq\,(fst\,(to \; eq \; a), b)$$

First definition using equivalences

Recall:

$$Lens'\ A\ B\ =\ (R\ :\ Set) \times (A\ \simeq\ R \times B)$$

Too big:

$$TLens\ \bot\ \bot\ \simeq\ \top$$
$$Lens'\ \ \bot\ \bot\ \simeq\ Set$$

Higher lenses

Due to Paolo Capriotti:

$$
\begin{aligned}
&HLens\ A\ B\ = \\
&\quad (get\ :\ A\ \rightarrow\ B) \times \\
&\quad (H\ \ :\ \parallel B \parallel\ \rightarrow\ Set) \times \\
&\quad (\lambda\ b.\ (a\ :\ A) \times (get\ a\ \equiv\ b))\ \equiv\ (\lambda\ b.\ H \mid b \mid)
\end{aligned}
$$

Higher lenses

Andrea Vezzosi and I found the following definition:

$$
\begin{aligned}
&ILens\ A\ B\ = \\
&\quad (R\ :\ Set) \times \\
&\quad (A\ \simeq\ R \times B) \times \\
&\quad (R\ \rightarrow\ \|\ B\ \|)
\end{aligned}
$$

- If B is empty, then R is empty.
- Equivalent to $HLens$ (assuming UA).
- We can still define get and set.

Identity

Recall:

$$ILens\ A\ A \stackrel{\text{def}}{=} (R\ :\ Set) \times (A\ \simeq\ R \times A) \times (R\ \rightarrow\ \|\ A\ \|)$$

For $ILens\ A\ A$:

$$A\ \simeq\ \|\ A\ \| \times A$$

Composition

Assume $A \simeq R_1 \times B$, $B \simeq R_2 \times C$. We get:

$$\begin{array}{ll} A & \simeq \\ R_1 \times B & \simeq \\ R_1 \times (R_2 \times C) & \simeq \\ (R_1 \times R_2) \times C & \end{array}$$

Also:

$$\begin{array}{ll} (R_1 \times R_2) & \rightarrow \\ R_2 & \rightarrow \\ \| C \| & \end{array}$$

or

$$\begin{array}{ll} (R_1 \times R_2) & \rightarrow \\ R_1 & \rightarrow \\ \| B \| & \rightarrow \\ \| C \| & \end{array}$$

Composition

Can prove

$$
\begin{aligned}
UA &\rightarrow id \circ l \equiv l, \\
UA &\rightarrow l \circ id \equiv l, \\
UA &\rightarrow l_1 \circ (l_2 \circ l_3) \equiv (l_1 \circ l_2) \circ l_3.
\end{aligned}
$$

The proofs are straightforward.

Relation between $ILens$ and $TLens$

Easy:

$$ILens\ A\ B \ \rightarrow\ TLens\ A\ B$$

If the domain is a set:

$$Is\text{-}set\ A\ \rightarrow\ ILens\ A\ B \twoheadrightarrow TLens\ A\ B$$
$$UA\ \rightarrow\ Is\text{-}set\ A\ \rightarrow\ ILens\ A\ B\ \simeq\ TLens\ A\ B$$

When defining an $ILens$ from a $TLens$:

$$R\ =\ (f\ :\ B\ \rightarrow\ A) \times (\forall\ b\ b'.\ set\ (f\ b)\ b'\ \equiv\ f\ b')$$

Relation between $ILens$ and $TLens$

If the codomain is a proposition,
then an $ILens$ is just a get function:

$$UA \rightarrow \textit{Is-proposition}\ B \rightarrow$$
$$ILens\ A\ B \simeq (A \rightarrow B)$$

Relation between $ILens$ and $TLens$

If the codomain is a proposition,
then an $ILens$ is just a get function:

$$UA \rightarrow \textit{Is-proposition}\ B \rightarrow$$
$$ILens\ A\ B \simeq (A \rightarrow B)$$

This is not necessarily the case for $TLens$es:

$$\textit{Is-proposition}\ B \rightarrow$$
$$TLens\ A\ B \simeq (A \rightarrow B) \times ((a : A) \rightarrow a \equiv a)$$

Relation between $ILens$ and $TLens$

If the codomain is $\top$,
then an $ILens$ is $\top$:

$$UA \rightarrow$$
$$ILens\ A\ \top \simeq \top$$

This is not necessarily the case for $TLens$es:

$$TLens\ A\ \top \simeq ((a : A) \rightarrow a \equiv a)$$

Relation between $ILens$ and $TLens$

Kraus and Sattler have shown

$$UA \rightarrow \neg\, \mathit{Is\text{-}proposition}\,((a : A) \rightarrow a \equiv a),$$

where $A = (B : Set) \times (B \equiv B)$.

Relation between $ILens$ and $TLens$

Kraus and Sattler have shown

$$UA \to \neg\, \textit{Is-proposition}\,((a : A) \to a \equiv a),$$

where $A = (B : Set) \times (B \equiv B)$.

We get:

$$\begin{aligned} UA \to \neg\, (ILens\ \ ((B : Set) \times (B \equiv B))\ \top \twoheadrightarrow \\ TLens\,((B : Set) \times (B \equiv B))\ \top) \end{aligned}$$

Relation between $ILens$ and $TLens$

I don't know if we can prove

$$TLens\ A\ B \rightarrow ILens\ A\ B$$

or

$$\neg\,(TLens\ A\ B \rightarrow ILens\ A\ B).$$

Relation between $ILens$ and $TLens$

Both definitions satisfy:

$$Lens\ A\ B \ \rightarrow\ A \ \rightarrow\ H\text{-}level\ n\ A \ \rightarrow\ H\text{-}level\ n\ B$$

All h-levels are closed under $TLens$:

$$H\text{-}level\ n\ A \ \rightarrow\ H\text{-}level\ n\ B \ \rightarrow \\ H\text{-}level\ n\ (TLens\ A\ B)$$

For $ILens$ I have (so far?) only managed to prove:

$$UA \ \rightarrow\ H\text{-}level\ n\ A \ \rightarrow \\ H\text{-}level\ (1 + n)\ (ILens\ A\ B)$$

No first projection lens

For both definitions one can find A, B such that

$$\neg \; Lens \, (\Sigma \; A \; B) \; A.$$

Example: $A \; = \; Bool$, $B \; a \; = \; a \; \equiv \; true$.

Dependent lenses

Second projection lens?

What if we want to define a lens corresponding to the second projection?

$$Lens\ :\ (A\ :\ Set)\ \rightarrow\ (A\ \rightarrow\ Set)\ \rightarrow\ Set$$

$$\begin{aligned}&second\text{-}projection\ :\\&\quad Lens\,(\Sigma\ A\ B)\ (\lambda\ (a,\_).\ B\ a)\end{aligned}$$

Example

A dependent record type:

$$
\begin{array}{l}
\textbf{record}\ R\ :\ Set\ \textbf{where} \\
\quad \textbf{field} \\
\qquad \begin{array}{lcl}
x & : & Bool \\
f & : & Bool \to Bool \\
f{\equiv}id & : & \forall\ y.\ f\ y \equiv y
\end{array}
\end{array}
$$

Should be possible to define:

$$
\begin{array}{lcl}
x & : & Lens\ R\ (\lambda\ \_.\ Bool) \\
f & : & Lens\ R\ (\lambda\ \_.\ (f\ :\ Bool \to Bool) \times \\
 & & \qquad\qquad\quad \forall\ y.\ f\ y \equiv y) \\
f{\equiv}id & : & Lens\ R\ (\lambda\ r.\ \forall\ y.\ R_1.f\ r\ y \equiv y)
\end{array}
$$

Dependent lenses

Preliminary definition:

$$
\begin{array}{l}
\mathit{Lens} : (A : \mathit{Set}) \to (A \to \mathit{Set}) \to \mathit{Set} \\
\mathit{Lens}\ A\ B = \\
\quad \begin{array}{ll}
(R & : \mathit{Set}) \times \\
(B' & : R \to \mathit{Set}) \times \\
(\mathit{eq} & : A \simeq \Sigma\, R\, B') \times \\
(\mathit{inhabited} & : (r : R) \to \|\, B'\ r \,\|) \times \\
\mathbf{let} & \mathit{remainder} : A \to R \\
 & \mathit{remainder}\ a = \mathit{fst}\ (\mathit{to}\ \mathit{eq}\ a) \\
\mathbf{in} & \\
(\mathit{variant} & : \forall\ a.\ B'\ (\mathit{remainder}\ a) \equiv B\ a)
\end{array}
\end{array}
$$

Dependent lenses

Equivalently:

$$
\begin{array}{l}
\mathit{Lens} : (A : \mathit{Set}) \to (A \to \mathit{Set}) \to \mathit{Set} \\
\mathit{Lens}\ A\ B = \\
\quad \begin{array}{ll}
(R & : \mathit{Set}) \times \\
(B' & : R \to \mathit{Set}) \times \\
(\mathit{eq} & : A \simeq \Sigma\, R\, B') \times \\
(\mathit{inhabited} & : (r : R) \to \|\, B'\ r \,\|) \times \\
(\mathit{variant} & : (r : R)\ (b' : B'\ r) \to \\
 & \ B'\ r \equiv B\,(\mathit{from}\ \mathit{eq}\,(r, b')))
\end{array}
\end{array}
$$

Getter

$$
\begin{aligned}
& \mathit{Lens}\ A\ B\ = \\
& \quad (R && : \mathit{Set}) \times \\
& \quad (B' && : R \to \mathit{Set}) \times \\
& \quad (\mathit{eq} && : A \simeq \Sigma\, R\, B') \times \\
& \quad (\mathit{inhabited} && : (r : R) \to \|\, B'\, r \,\|) \times \\
& \quad \textbf{let} && \mathit{remainder} : A \to R \\
& && \mathit{remainder}\ a = \mathit{fst}\ (\mathit{to}\ \mathit{eq}\ a) \\
& \quad \textbf{in} && \\
& \quad (\mathit{variant} && : \forall\ a.\ B'\ (\mathit{remainder}\ a) \equiv B\ a)
\end{aligned}
$$

$$
\begin{aligned}
& \mathit{get} : (a : A) \to B\ a \\
& \mathit{get}\ a = \mathit{to}\ (\mathit{variant}\ a)\ (\mathit{snd}\ (\mathit{to}\ \mathit{eq}\ a))
\end{aligned}
$$

Setter

$$
\begin{aligned}
&\mathit{Lens}\ A\ B\ =\\
&\quad (R \qquad\qquad\quad : \mathit{Set}) \times\\
&\quad (B' \qquad\qquad\ \ : R \to \mathit{Set}) \times\\
&\quad (\mathit{eq} \qquad\qquad\ : A \simeq \Sigma\, R\, B') \times\\
&\quad (\mathit{inhabited} : (r : R) \to \|\, B'\, r\, \|) \times\\
&\quad \textbf{let} \qquad\qquad\quad \mathit{remainder} : A \to R\\
&\qquad\qquad\qquad\qquad\ \ \mathit{remainder}\ a = \mathit{fst}\ (\mathit{to}\ \mathit{eq}\ a)\\
&\quad \textbf{in}\\
&\quad (\mathit{variant} \qquad : \forall\ a.\ B'\ (\mathit{remainder}\ a) \equiv B\ a)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{set} : (a : A) \to B\ a \to A\\
&\mathit{set}\ a\ b = \mathit{from}\ \mathit{eq}\ (\mathit{remainder}\ a, \mathit{from}\ (\mathit{variant}\ a)\ b)
\end{aligned}
$$

Lens laws

$$remainder\,(set\ a\ b) \equiv remainder\ a$$

$$unchanged : B\,(set\ a\ b) \equiv B\ a$$

$$\begin{aligned}
set\ a\,(get\ a) &\equiv a \\
get\,(set\ a\ b) &\equiv from\ unchanged\ b \\
set\,(set\ a\ b_1)\ b_2 &\equiv set\ a\,(to\ unchanged\ b_2)
\end{aligned}$$

Propositional codomain

If the codomain is a family of propositions, then a $Lens$ is just a get function:

$$UA \ \to \ (\forall\ a.\ \mathit{Is\text{-}proposition}\ (B\ a)) \ \to$$
$$\mathit{Lens}\ A\ B \ \simeq \ ((a\ :\ A) \ \to \ B\ a)$$

Composition

Can we define a composition operator?

$$
\begin{aligned}
&\_\circ\_ \ : \\
&\quad \{A : Set\}\ \{B : A \to Set\} \\
&\quad \{C : (a : A) \to B\ a \to Set\} \\
&\quad (l_1 : Lens\ A\ B) \\
&\quad (l_2 : \forall\ a.\ Lens\ (B\ a)\ (C\ a)) \to \\
&\quad Lens\ A\ (\lambda\ a.\ C\ a\ (get\ l_1\ a))
\end{aligned}
$$

A requirement on the resulting get function:

$$get\ a \equiv get\ (l_2\ a)\ (get\ l_1\ a)$$

Composition

If we can prove that non-dependent dependent lenses are isomorphic to $ILens$es, then the answer is no.

Assuming K:

$$
\begin{aligned}
& K \rightarrow \\
& (eq : Lens\ A\ (\lambda\ \_.\ B) \simeq ILens\ A\ B) \times \\
& \quad \forall\ l\ a \rightarrow get\ l\ a \equiv get\ (to\ eq\ l)\ a
\end{aligned}
$$

(What if we have UA instead?)

Composition

Counterexample:

$$
\begin{aligned}
l_1 &: \mathit{Lens}\ \mathit{Bool}\ (\lambda\ \_.\ \mathit{Bool}) \\
l_1 &= \mathit{id}
\end{aligned}
$$

$$
\begin{aligned}
l_2 &: \mathit{Bool} \rightarrow \mathit{Lens}\ \mathit{Bool}\ (\lambda\ \_.\ \mathit{Bool}) \\
l_2\ \mathit{true} &= \mathit{id} \\
l_2\ \mathit{false} &= \mathit{swap}
\end{aligned}
$$

$$\forall\ b.\ \mathit{get}\ b \equiv \mathit{true}$$

$$
\begin{aligned}
&\mathit{true} && \equiv \\
&\mathit{get}\ (\mathit{set}\ \mathit{true}\ \mathit{false}) && \equiv \\
&\mathit{false}
\end{aligned}
$$

Composition

A variant that is inhabited:

$$
\begin{array}{l}
\_\circ\_ \;: \\
\quad \{A : \mathit{Set}\}\ \{B\ C : A \to \mathit{Set}\} \to \\
\quad (l_1 : \mathit{Lens}\ A\ B) \to \\
\quad (l_2 : (r : R\ l_1) \to \\
\qquad\qquad \mathit{Lens}\ (B'\ l_1\ r) \\
\qquad\qquad\qquad (\lambda\ b'.\ C\ (\mathit{from}\ (\mathit{eq}\ l_1)\ (r, b')))) \to \\
\quad \mathit{Lens}\ A\ C
\end{array}
$$

Has worked well in the examples I have tried.

Discussion

- Higher lenses.
  - Perhaps the definition of $HLens/ILens$ is OK.
  - Some open questions.
- Dependent lenses.
  - Perhaps one can find a better definition.
  - The present definition might be OK in the presence of K.
  - Impossible to define composition.