Verification of Redecoration for Infinite Triangular Matrices in Coq

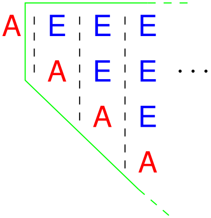
Celia Picard

joint work with Ralph Matthes

13/09/2012

Introduction

The finite case: finite triangles seen as triangular matrices



- Type E fixed throughout
- Special type A for elements of the diagonal
- Inductive family $Tri_{fin}\ A$

Introduction

The finite case: definitions

Triangles and trapeziums

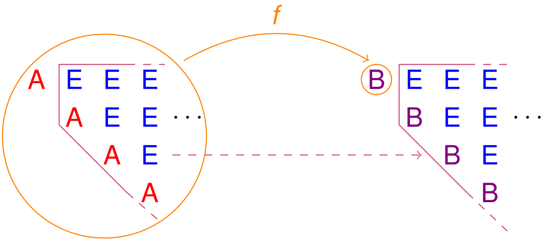
$$
\begin{array}{ccccc}
\mathbf{E} & E & E & E & \\
A & \mathbf{E} & E & E & \\
 & A & \mathbf{E} & E & \cdots \\
 & & A & \mathbf{E} & \\
 & & & A &
\end{array}
\quad
\begin{array}{c}
\xrightarrow{\text{triangle}} \\
\xleftarrow[\text{trapezium}]{}
\end{array}
\quad
\begin{array}{ccccc}
\mathbf{E} \times A & E & E & E & \\
 & \mathbf{E} \times A & E & E & \cdots \\
 & & \mathbf{E} \times A & E & \\
 & & & \mathbf{E} \times A &
\end{array}
$$

Formal definition

$$\frac{a : A}{sg_{fin}\, a : Tri_{fin}\, A} \qquad \frac{a : A \qquad t : Tri_{fin}(E \times A)}{constr_{fin}\, a\, t : Tri_{fin}\, A}$$

Introduction

The finite case: redecoration



Verifications of redecoration:

- against a list-based model
- comonad laws

All details in:

[A. Abel, R. Matthes, and T. Uustalu. Iteration and coiteration schemes for higher-order and nested datatypes. Theoretical Computer Science 2005]

[R. Matthes and M. Strecker. Verification of the redecoration algorithm for triangular matrices. TYPES 2007]

Outline

1. Reference Representation with a Coinductive Family
2. Another Conception of Triangles
3. Conclusion

Outline

1. Reference Representation with a Coinductive Family
   - Definitions
   - Redecoration
   - Properties of redecoration
2. Another Conception of Triangles
3. Conclusion

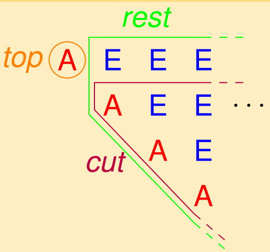
Reference Representation with a Coinductive Family

Definitions

Definition of the infinite triangles

$$\frac{a : A \qquad t : \mathit{Tri}(E \times A)}{\mathit{constr}\ a\ t : \mathit{Tri}\ A} \qquad\qquad \mathit{Trap}\ A = \mathit{Tri}(E \times A)$$

Projections and function *cut*



$\mathit{top} : \forall A.\ \mathit{Tri}\ A \to A$

$\mathit{top}\ (\mathit{constr}\ a\ r) := a$

$\mathit{rest} : \forall A.\ \mathit{Tri}\ A \to \mathit{Trap}\ A$

$\mathit{rest}\ (\mathit{constr}\ a\ r) := r$

$\mathit{cut}\ (\mathit{constr}\ \langle e, a\rangle\ r) := \mathit{constr}\ a\ (\mathit{cut}\ r)$

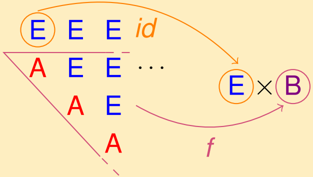
top: counit of the comonad (signature)

Outline

Reference Representation with a Coinductive Family

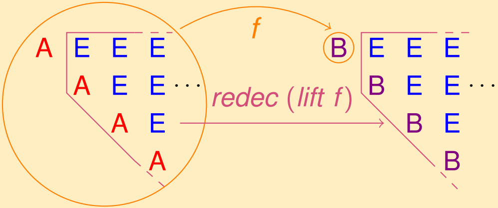
Redecoration

The *lift* operation



$$\begin{aligned} lift &: \forall A \forall B.\,(Tri\ A \to B) \to \\ &\quad Tri(E \times A) \to E \times B \\ lift\ f\ r &:= \langle fst(top\ r), f(cut\ r)\rangle \end{aligned}$$

The *redec* operation



$$\begin{aligned} redec &: \forall A \forall B.\,(Tri\ A \to B) \\ &\quad \to Tri\ A \to Tri\ B \\ redec\ f\ t &:= \\ &\quad constr\ (f\ t) \\ &\qquad\qquad \big(redec\ (lift\ f) \\ &\qquad\qquad\qquad (rest\ t)\big) \end{aligned}$$

Outline

Reference Representation with a Coinductive Family

Properties of redecoration – New tools on *Tri*

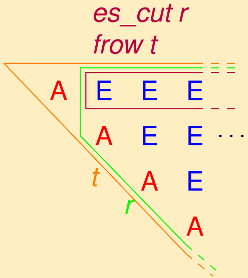
Equivalence relation on *Tri* and associated result

$$\frac{t_1\ ,\ t_2 : Tri\ A \qquad top\ t_1 = top\ t_2 \qquad rest\ t_1 \simeq rest\ t_2}{t_1 \simeq t_2}$$

$$\forall A \forall B \forall (f\ f' : Tri\ A \to B).\ (\forall t, f\ t = f'\ t) \Rightarrow \forall t, redec\ f\ t \simeq redec\ f'\ t$$

with full extensionality of *lift*.

New tools to manipulate *Tri*



$es\_cut : \forall A.\ Trap\ A \to Str\ E$

$es\_cut\ (constr\ \langle e, a\rangle\ r) := e :: (es\_cut\ r)$

$frow : \forall A.\ Tri\ A \to Str\ E$

$frow\ t := es\_cut\ (rest\ t)$

$addes : \forall A.\ Str\ E \to Tri\ A \to Trap\ A$

$addes\ (e :: es)\ (constr\ a\ r) :=$
$\quad constr\ \langle e, a\rangle\ (addes\ es\ r)$

$$\forall A \forall (r : Trap\ A).\ addes\ (es\_cut\ r)\ (cut\ r) \simeq r$$

Reference Representation with a Coinductive Family

Properties of redecoration – Comonad

Comonad laws

$$\forall A \forall B \forall f^{\mathit{Tri}\,A \to B} \forall t^{\mathit{Tri}\,A}.\ \mathit{top}(\mathit{redec}\ f\ t) = f\ t \tag{1}$$

$$\forall A \forall t^{\mathit{Tri}\,A}.\ \mathit{redec}\ \mathit{top}\ t \simeq t \tag{2}$$

$$\forall A \forall B \forall f^{\mathit{Tri}\,A \to B} \forall g^{\mathit{Tri}\,B \to C} \forall t^{\mathit{Tri}\,A}.$$
$$\mathit{redec}\,(g \circ \mathit{redec}\ f)\ t \simeq \mathit{redec}\ g\,(\mathit{redec}\ f\ t) \tag{3}$$

Application to *Tri*

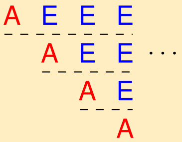
- The type transformation *Tri*, the projection function *top* (*counit*) and *redec* (*cobind*) form a constructive weak comonad w. r. t. $\simeq$. Weak: compatibility of g in (3) required.
- Proved with a stronger form of (2):
$$\forall A \forall (f : \mathit{Tri}\,A \to A).\,(\forall (t : \mathit{Tri}\,A), f\ t = \mathit{top}\ t) \Rightarrow \forall (t : \mathit{Tri}\,A).\ \mathit{redec}\ f\ t \simeq t \tag{2'}$$

Outline

Another Conception of Triangles

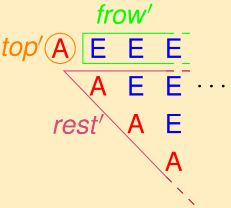
Definitions (1/2)

New visualization



$$Tri'\ A := Str(A \times Str\ E)$$

The three parts of the definition



$top'\ : \forall A.\ Tri'\ A \rightarrow A$
$top'\ (\langle a, es \rangle :: t) := a$

$frow'\ : \forall A.\ Tri'\ A \rightarrow Str\ E$
$frow'\ (\langle a, es \rangle :: t) := es$

$rest'\ : \forall A.\ Tri'\ A \rightarrow Tri'\ A$
$rest'\ (\langle a, es \rangle :: t) := t$

Another Conception of Triangles

Definitions (2/2)

Equivalence relation

$$\frac{t_1\,,\ t_2 : \mathit{Tri}'\ A \qquad \mathit{top}'\ t_1 = \mathit{top}'\ t_2 \qquad \mathit{frow}'\ t_1 \equiv \mathit{frow}'\ t_2 \qquad \mathit{rest}'\ t_1 \cong \mathit{rest}'\ t_2}{t_1 \cong t_2}$$

with $\equiv$ relation on *Str* (cannot use it directly)

Conversion functions to/from *Tri*

$$\begin{tikzcd} \mathit{Tri} \arrow[rr, bend left, "\mathit{toStrRep}"] & & \mathit{Tri}' \arrow[ll, bend left, "\mathit{fromStrRep}"] \end{tikzcd}$$

$\forall A \forall (t : \mathit{Tri}'\ A).\ \mathit{toStrRep}\,(\mathit{fromStrRep}\ t) \cong t$
$\forall A \forall (t : \mathit{Tri}\ A).\ \mathit{fromStrRep}\,(\mathit{toStrRep}\ t) \simeq t$

Outline

1. Reference Representation with a Coinductive Family

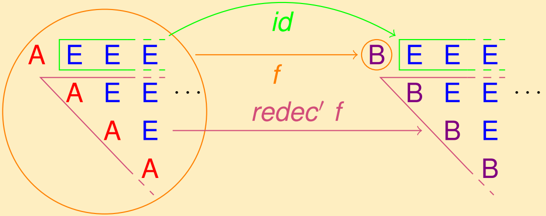
2. Another Conception of Triangles
   - Definitions
   - Redecoration

3. Conclusion

Another Conception of Triangles

Redecoration

Definition – Much simpler than *redec*



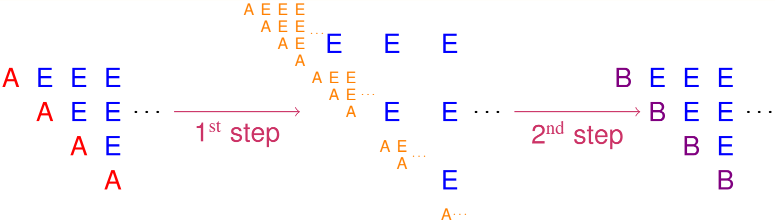
$$redec' : \forall A \forall B. (Tri'\ A \to B) \to Tri'\ A \to Tri'\ B$$
$$redec'\ f\ t := \langle f\ t, frow'\ t\rangle :: redec'\ f\ (rest'\ t)$$

Results

$$\begin{array}{ll} \forall f\ t, & (\forall t\ t', t \cong t' \Rightarrow f\ t = f\ t') \\ \quad\Rightarrow & redec'\ f\ t \cong toStrRep\,(redec\,(f \circ toStrRep)\,(fromStrRep\ t)) \\ \forall f\ t, & (\forall t\ t', t \simeq t' \Rightarrow f\ t = f\ t') \\ \quad\Rightarrow & redec\ f\ t \simeq fromStrRep\,(redec'\,(f \circ fromStrRep)\,(toStrRep\ t)) \end{array}$$

Another Conception of Triangles

Simplifying again



$tails : \forall A.\, Str\ A \to Str(Str\ A)$

$tails\ s := s :: tails\,(tl\ s)$

$lift' : \forall A \forall B.\,(Tri'\ A \to B) \to Tri'\ A \to B \times Str\ E$

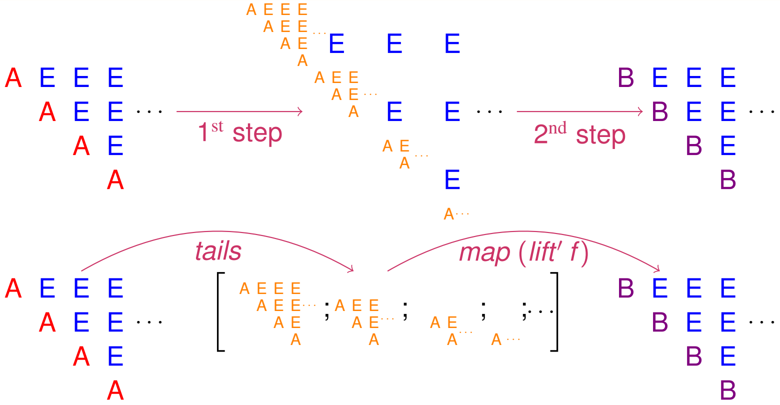
$lift'\ f := \lambda x.\langle f\ x, frow'\ x\rangle$

$redec'_{alt} : \forall A \forall B.\,(Tri'\ A \to B) \to Tri'\ A \to Tri'\ B$

$redec'_{alt}\ f\ t := map\,(lift'\ f)\,(tails\ t)$

$redec'\ f\ t \equiv redec'_{alt}\ f\ t$

Another Conception of Triangles
Simplifying again



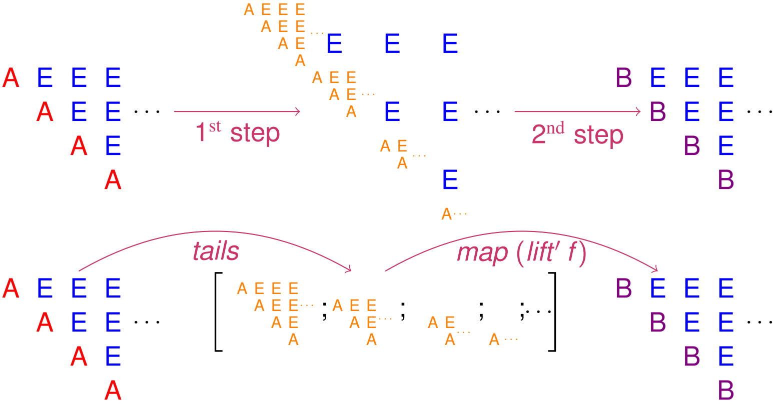
$$\mathit{tails} : \forall A.\, \mathit{Str}\ A \to \mathit{Str}(\mathit{Str}\ A) \qquad \mathit{lift}' : \forall A \forall B.\, (\mathit{Tri}'\ A \to B) \to \mathit{Tri}'\ A \to B \times \mathit{Str}\ E$$

$$\mathit{tails}\, s := s :: \mathit{tails}\, (\mathit{tl}\ s) \qquad \mathit{lift}'\ f := \lambda x. \langle f\ x, \mathit{frow}'\ x \rangle$$

$$\mathit{redec}'_{alt} : \forall A \forall B.\, (\mathit{Tri}'\ A \to B) \to \mathit{Tri}'\ A \to \mathit{Tri}'\ B$$

$$\mathit{redec}'_{alt}\ f\ t := \mathit{map}\, (\mathit{lift}'\ f)\, (\mathit{tails}\ t) \qquad \mathit{redec}'\ f\ t \equiv \mathit{redec}'_{alt}\ f\ t$$

Another Conception of Triangles
Simplifying again



$$tails : \forall A.\, Str\ A \to Str(Str\ A) \qquad lift' : \forall A \forall B.\, (Tri'\ A \to B) \to Tri'\ A \to B \times Str\ E$$
$$tails\ s := s :: tails\,(tl\ s) \qquad lift'\ f := \lambda x. \langle f\ x, frow'\ x\rangle$$

$$redec'_{alt} : \forall A \forall B.\, (Tri'\ A \to B) \to Tri'\ A \to Tri'\ B$$
$$redec'_{alt}\ f\ t := map\,(lift'\ f)\,(tails\ t)$$

$$redec'\ f\ t \equiv redec'_{alt}\ f\ t$$

Another Conception of Triangles

Generalizing

Generalizing redecoration (redecoration on *Str*)

$redec'_{gen} : \forall A \forall B. (Str\ A \to B) \to Str\ A \to Str\ B$
$redec'_{gen}\ f\ s := map\ f\ (tails\ s)$

$\forall A \forall B \forall (f : Tri'\ A \to B)(t : Tri'\ A).\ redec'_{alt}\ f\ t = redec'_{gen}\ (lift'\ f)\ t$

Comonads

- The type transformation *Str*, the projection function *hd* and $redec'_{gen}$ form a constructive comonad with respect to $\equiv$
- The type transformation Tri', the projection function top' and $redec'_{alt}$ form a constructive comonad with respect to $\equiv$ (more precisely, the equivalence relation is $\equiv_{A \times Str\ E}$ for every A) – stronger result than before

Conclusion

- Redecoration for infinite triangles:
  - dualized directly from representation of finite triangles:
    - columnwise visualization
    - tricky algorithm for *redec* (polymorphic recursion)
    - constructive *weak* comonad
  - representation with streams:
    - row-wise visualization
    - purely coinductive
    - simple versions of redecoration
    - constructive comonad
- Easier to work on totally infinite structure than with partially finite ones. Even easier than finite triangles.
- `www.irit.fr/~Celia.Picard/Coq/Redecoration/`

Conclusion

- Redecoration for infinite triangles:
  - dualized directly from representation of finite triangles:
    - columnwise visualization
    - tricky algorithm for *redec* (polymorphic recursion)
    - constructive *weak* comonad
  - representation with streams:
    - row-wise visualization
    - purely coinductive
    - simple versions of redecoration
    - constructive comonad
- Easier to work on totally infinite structure than with partially finite ones. Even easier than finite triangles.
- `www.irit.fr/~Celia.Picard/Coq/Redecoration/`

Thanks for your attention.