Verifying Tail Modulo Cons using Relational Separation Logic

Clément Allain
Gabriel Scherer
François Pottier

INRIA Paris

May 31, 2024

Verifying Tail Modulo Cons using Relational Separation Logic

Program transformation implemented in the OCaml compiler.

Verifying Tail Modulo Cons using Relational Separation Logic

Formalize the transformation and its soundness.

Verifying Tail Modulo Cons using Relational Separation Logic

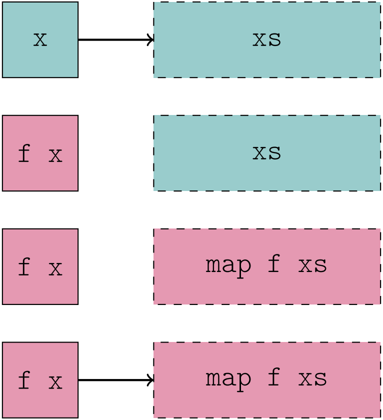
Prove soundness using an adequate IRIS binary logical relation à la SIMULIRIS.

The map problem: natural implementation

```
let rec map f xs =
  match xs with
  | [] →
      []
  | x :: xs →
      let y = f x in
      y :: map f xs
```

```
# List.init 250_000 (fun _ → ())
  |> map Fun.id
  |> ignore
  ;;
Stack overflow during evaluation (looping recursion?).
```

The `map` problem: natural implementation

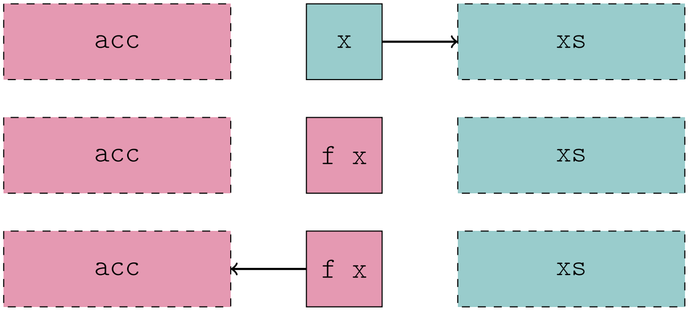


The map problem: APS implementation

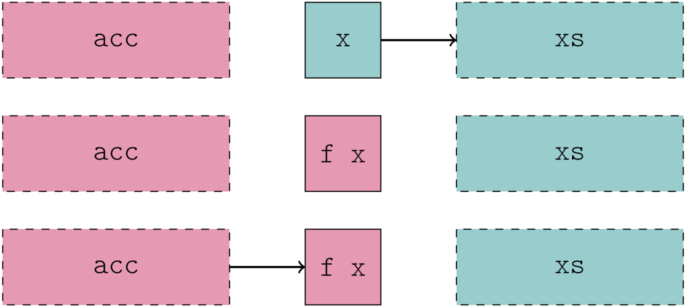
```
let rec map ys f xs =
  match xs with
  | [] →
      List.rev ys
  | x :: xs →
      let y = f x in
      map (y :: ys) f xs
let map xs =
  map [] f xs
```

```
# List.init 250_000 (fun _ → ())
  |> map Fun.id
  |> ignore
  ;;
- : unit = ()
```

The `map` problem: APS implementation



The `map` problem: DPS implementation



The map problem: DPS implementation

```
let rec map_dps dst f xs =
  match xs with
  | [] →
      set_field dst 1 []
  | x :: xs →
      let y = f x in
      let dst' = y :: [] in
      set_field dst 1 dst' ;
      map_dps dst' f xs
```

```
let map f xs =
  match xs with
  | [] →
      []
  | x :: xs →
      let y = f x in
      let dst = y :: [] in
      map_dps dst f xs ;
      dst
```

```
# List.init 250_000 (fun _ → ())
  |> map Fun.id
  |> ignore
  ;;
- : unit = ()
```

The map problem: TMC

```
let[@tail_mod_cons] rec map f xs =
  match xs with
  | [] →
      []
  | x :: xs →
      let y = f x in
      y :: map f xs
```

```
# List.init 250_000 (fun _ → ())
  |> map Fun.id
  |> ignore
  ;;
- : unit = ()
```

DataLang: syntax

$$
\begin{array}{llcll}
\text{Index} & \ni & i & ::= & \mathtt{0} \mid \mathtt{1} \mid \mathtt{2} \\
\text{Tag} & \ni & t & & \\
\mathbb{B} & \ni & b & & \\
\mathbb{L} & \ni & \ell & & \\
\mathbb{F} & \ni & f & & \\
\mathbb{X} & \ni & x, y & & \\
\text{Val} & \ni & v, w & ::= & \mathtt{()} \mid i \mid t \mid b \mid \ell \mid \mathtt{@}f \\
\text{Expr} & \ni & e & ::= & v \mid x \mid \mathtt{let}\ x \mathrel{\mathtt{=}} e_1\ \mathtt{in}\ e_2 \mid e_1\ \overline{e_2} \\
 & & & \mid & e_1 \mathrel{\mathtt{=}} e_2 \mid \mathtt{if}\ e_0\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \\
 & & & \mid & \mathtt{\{}\, t \mathtt{,}\ e_1 \mathtt{,}\ e_2 \,\mathtt{\}} \\
 & & & \mid & e_1 \mathtt{.(} e_2 \mathtt{)} \mid e_1 \mathtt{.(} e_2 \mathtt{)} \leftarrow e_3 \\
\text{Def} & \ni & d & ::= & \mathtt{rec}\ \overline{x} \mathrel{\mathtt{=}} e \\
\text{Prog} & \ni & p & := & \mathbb{F} \xrightarrow{\text{fin}} \text{Def} \\
\text{State} & \ni & \sigma & := & \mathbb{L} \xrightarrow{\text{fin}} \text{Val} \\
\text{Config} & \ni & \rho & := & \text{Expr} \times \text{State}
\end{array}
$$

DataLang: semantics

StepCall

$$\frac{p[f] = (\texttt{rec}\ \overline{x}\ \texttt{=}\ e)}{(\texttt{@}f\ \overline{v}, \sigma) \xrightarrow[\text{head}]{p} (e[\overline{x}\backslash\overline{v}], \sigma)}$$

StepBlock

$$\frac{\forall\, i \in \text{Index}, \ell + i \notin \text{dom}(\sigma)}{(\texttt{\{}\, t\, \texttt{,}\, v_1\, \texttt{,}\, v_2\, \texttt{\}}, \sigma) \xrightarrow[\text{head}]{p} (\ell, \sigma[\ell \mapsto t, v_1, v_2])}$$

StepLoad

$$\frac{\sigma[\ell] = v}{(\ell\, \texttt{.}\, \texttt{(}i\texttt{)}, \sigma) \xrightarrow[\text{head}]{p} (v, \sigma)}$$

StepStore

$$\frac{\ell + i \in \text{dom}(\sigma)}{(\ell\, \texttt{.}\, \texttt{(}i\texttt{)} \leftarrow v, \sigma) \xrightarrow[\text{head}]{p} (\texttt{()}, \sigma[\ell + i \mapsto v])}$$

DataLang: `map`

```
map ↦ rec f xs =
  match xs with
  | [] →
      []
  | x :: xs →
      let y = f x in
      y :: @map f xs
```

TMC transformation

$$e_s \overset{\xi}{\underset{\text{dir}}{\rightsquigarrow}} e_t \qquad d_s \overset{\xi}{\underset{\text{dir}}{\rightsquigarrow}} d_t$$

$$(e_{dst}, e_{idx}, e_s) \overset{\xi}{\underset{\text{dps}}{\rightsquigarrow}} e_t \qquad d_s \overset{\xi}{\underset{\text{dps}}{\rightsquigarrow}} d_t$$

$$p_s \rightsquigarrow p_t$$

TMC transformation: map

```
map ↦ rec f xs =
  match xs with
  | [] →
      []
  | x :: xs →
      let y = f x in
      let dst = y :: ■ in
      @map_dps dst 2 f xs ;
      dst
```

```
map_dps ↦ rec dst idx f xs =
  match xs with
  | [] →
      dst.(idx) ← []
  | x :: xs →
      let y = f x in
      let dst' = y :: ■ in
      dst.(idx) ← dst' ;
      @map_dps dst' 2 f xs
```

Transformation soundness

program p_s transforms into program p_t

$$p_s \rightsquigarrow p_t$$
$$\Longrightarrow$$
$$p_s \sqsupseteq p_t$$

program p_t refines program p_t
(termination-preserving refinement)

Specification in separation logic

$$\frac{\frac{\{v_s \approx v_t\}}{\texttt{@map}\ v_s \ \gtrsim\ \texttt{@map}\ v_t}}{\{w_s, w_t.\, w_s \approx w_t\}}$$

$$\frac{\frac{\{v_s \approx v_t * (\ell + i) \mapsto_t \blacksquare\}}{\texttt{@map}\ v_s \ \gtrsim\ \texttt{@map\_dps}\ \ell\ i\ v_t}}{\{(), w_t.\, \exists\, w_t.\, w_s \approx w_t * (\ell + i) \mapsto_t w_t\}}$$

Simuliris + protocols

RelDir (Simuliris)

$$\frac{f \in \mathrm{dom}(p_s) \qquad v_s \approx v_t \qquad \forall\, w_s, w_t.\, w_s \approx w_t \twoheadrightarrow \Phi(w_s, w_t)}{\texttt{@}f\ v_s \gtrsim \texttt{@}f\ v_t\ [\Phi]}$$

RelDPS

$$\frac{\xi[f] = f_{dps} \qquad \overline{v_s} \approx \overline{v_t} \qquad \ell \mapsto_t \overline{v} \qquad \forall\, w_s, w_t.\, w_s \approx w_t \twoheadrightarrow \ell \mapsto_t \overline{v}[i \mapsto w_t] \twoheadrightarrow \Phi(w_s, (\,))}{\texttt{@}f\ \overline{v_s} \gtrsim \texttt{@}f_{dps}\ \ell\ i\ \overline{v_t}\ [\Phi]}$$

RelProtocol

$$\frac{\mathrm{X}(\Psi, e_s, e_t) \qquad \forall\, e'_s, e'_t.\, \Psi(e'_s, e'_t) \twoheadrightarrow e'_s \gtrsim e'_t\ \langle \mathrm{X} \rangle\ \{\Phi\}}{e_s \gtrsim e_t\ \langle \mathrm{X} \rangle\ \{\Phi\}}$$

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

```
match xs_s with
| [] →
    []
| x :: xs' →
    let y = f_s x in
    y :: @map f_s xs'
```

$\gtrsim$

```
match xs_t with
| [] →
    []
| x :: xs' →
    let y = f_t x in
    let dst = y :: ■ in
    @map_dps dst 2 f_t xs' ;
    dst
```

Proof sketch

RelMatch

$$\frac{e_{s0} \gtrsim e_{t0}\ \{\approx\} \qquad e_{s1} \gtrsim e_{t1}\ [\Phi] \qquad \forall\, x_s, x_t, xs_s, xs_t.\, x_s \approx x_t \twoheadrightarrow\!\!* \; xs_s \approx xs_t \twoheadrightarrow\!\!* \; e_{s2} \gtrsim e_{t2}\ [\Phi]}{\begin{array}{l} \texttt{match}\ e_{s0}\ \texttt{with} \\ \mid \texttt{[]} \ \to\ e_{s1} \\ \mid x_s \ \texttt{::}\ xs_s \ \to\ e_{s2} \end{array} \gtrsim \begin{array}{l} \texttt{match}\ e_{t0}\ \texttt{with} \\ \mid \texttt{[]} \ \to\ e_{t1} \\ \mid x_t \ \texttt{::}\ xs_t \ \to\ e_{t2} \end{array}\ [\Phi]}$$

y :: @map f_s xs'   @map_dps dst 2 f_t xs' ; dst

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

`[]` $\gtrsim$ `[]`

Proof sketch

$$
\frac{f_s \approx f_t \qquad xs_s \approx xs_t \qquad x_s \approx x_t \qquad xs'_s \approx xs'_t}{\begin{array}{l}\textbf{let}\ \texttt{y} = f_s\ x_s\ \textbf{in} \\ \texttt{y :: @map}\ f_s\ xs'_s\end{array} \quad \gtrsim \quad \begin{array}{l}\textbf{let}\ \texttt{y} = f_t\ x_t\ \textbf{in} \\ \textbf{let}\ \texttt{dst = y ::}\ \blacksquare\ \textbf{in} \\ \texttt{@map\_dps dst 2}\ f_t\ xs'_t\ \texttt{;} \\ \texttt{dst}\end{array}}
$$

Proof sketch

$f_s \approx f_t \qquad xs_s \approx xs_t$

$x_s \approx x_t \qquad xs'_s \approx xs'_t$

RelvCallSimilar

$$\frac{f_s \approx f_t \qquad x_s \approx x_t}{f_s\ x_s \gtrsim f_t\ x_t\ \{\approx\}}$$

```
let y = f_s x_s in
y :: @map f_s xs'_s
```

$\gtrsim$

```
let y = f_t x_t in
let dst = y :: ■ in
@map_dps dst 2 f_t xs'_t ;
dst
```

Proof sketch

$$\dfrac{f_s \approx f_t \qquad xs_s \approx xs_t \qquad x_s \approx x_t \qquad xs'_s \approx xs'_t \qquad y_s \approx y_t}{\begin{array}{l}\texttt{let y = } y_s \texttt{ in} \\ \texttt{y :: @map } f_s \ xs'_s\end{array} \quad \gtrsim \quad \begin{array}{l}\texttt{let y = } y_t \texttt{ in} \\ \texttt{let dst = y :: } \blacksquare \texttt{ in} \\ \texttt{@map\_dps dst 2 } f_t \ xs'_t \texttt{ ;} \\ \texttt{dst}\end{array}}$$

Proof sketch

$$f_s \approx f_t \qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad xs'_s \approx xs'_t$$

RelPure

$$\frac{e_s \xrightarrow[\text{pure}]{p_s} e'_s \qquad e_t \xrightarrow[\text{pure}]{p_t} e'_t \qquad e'_s \gtrsim e'_t\ [\Phi]}{e_s \gtrsim e_t\ [\Phi]}$$

```
let y = y_s in                let y = y_t in
y :: @map f_s xs'_s     ≳     let dst = y :: ■ in
                              @map_dps dst 2 f_t xs'_t ;
                              dst
```

Proof sketch

$$\frac{\begin{array}{cc} f_s \approx f_t & xs_s \approx xs_t \\ x_s \approx x_t & xs'_s \approx xs'_t \\ \multicolumn{2}{c}{y_s \approx y_t} \end{array}}{y_s \;\texttt{::}\; \texttt{@map}\; f_s\; xs'_s \quad \gtrsim \quad \begin{array}{l} \textbf{let}\ \texttt{dst} = y_t \;\texttt{::}\; \blacksquare\ \textbf{in} \\ \texttt{@map\_dps dst 2}\; f_t\; xs'_t\ \texttt{;} \\ \texttt{dst} \end{array}}$$

Proof sketch

$$f_s \approx f_t \qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad xs'_s \approx xs'_t$$

RelTgtCons

$$\frac{\forall \ell.\, \ell \mapsto_t (\mathrm{CONS}, v_1, v_2) \mathrel{-\!\!*} e_s \gtrsim \ell\ [\Phi]}{e_s \gtrsim v_1 :: v_2\ [\Phi]}$$

$$y_s \texttt{ :: @map } f_s\ xs'_s \quad \gtrsim \quad \begin{array}{l} \texttt{let dst = } y_t \texttt{ :: } \blacksquare \texttt{ in} \\ \texttt{@map\_dps dst 2 } f_t\ xs'_t \texttt{ ;} \\ \texttt{dst} \end{array}$$

Proof sketch

$$\dfrac{f_s \approx f_t \qquad xs_s \approx xs_t \qquad x_s \approx x_t \qquad xs'_s \approx xs'_t \qquad y_s \approx y_t \qquad \ell_t \mapsto_t (\mathrm{CONS}, y_t, \blacksquare)}{y_s\ \texttt{::}\ \texttt{@map}\ f_s\ xs'_s \quad \gtrsim \quad \begin{array}{l}\textbf{\texttt{let}}\ \texttt{dst = }\ell_t\ \textbf{\texttt{in}}\\ \texttt{@map\_dps dst 2}\ f_t\ xs'_t\ \texttt{;}\\ \texttt{dst}\end{array}}$$

Proof sketch

$$f_s \approx f_t \qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad xs'_s \approx xs'_t$$

$$\textsc{RelTgtPure}\quad \frac{e_t \xrightarrow[\text{pure}]{p_t} e'_t \qquad e_s \gtrsim e'_t\ [\Phi]}{e_s \gtrsim e_t\ [\Phi]}$$

y_s `::` `@map` f_s xs'_s $\gtrsim$ `let dst =` ℓ_t `in` `@map_dps dst 2` f_t xs'_t `;` `dst`

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad\qquad xs'_s \approx xs'_t$$

$$y_s \approx y_t$$

$$\ell_t \mapsto_t (\text{CONS}, y_t, \blacksquare)$$

---

y_s `::` `@map` f_s xs'_s $\quad\gtrsim\quad$ `@map_dps` ℓ_t `2` f_t xs'_t `;` ℓ_t

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

RelDPS2

$$\frac{\xi[f] = f_{dps} \quad \overline{v_s} \approx \overline{v_t} \quad \ell \mapsto_t (t, v_1, v_2) \quad \forall\, w_s, w_t.\ w_s \approx w_t \twoheadrightarrow \ell \mapsto_t (t, v_1, w_t) \twoheadrightarrow \Phi(w_s, ())}{\texttt{@}f\ \overline{v_s} \gtrsim \texttt{@}f_{dps}\ \ell\ 2\ \overline{v_t}\ [\Phi]}$$

$$y_s \; :: \; \texttt{@map}\ f_s\ xs'_s \quad \gtrsim \quad \texttt{@map\_dps}\ \ell_t\ 2\ f_t\ xs'_t\ ; \ \ell_t$$

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad\qquad xs'_s \approx xs'_t$$

$$y_s \approx y_t$$

$$ys_s \approx ys_t$$

$$\ell_t \mapsto_t (\mathrm{CONS}, y_t, ys_t)$$

$$y_s \;::\; ys_s \qquad \gtrsim \qquad () \;;\; \ell_t$$

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad\qquad xs'_s \approx xs'_t$$

$$y_s \approx y_t$$

$$ys_s \approx ys_t$$

$$\ell_t \mapsto_t (\mathrm{CONS}, y_t, ys_t)$$

$$y_s \;::\; ys_s \qquad\qquad \gtrsim \quad \ell_t$$

Proof sketch

$$f_s \approx f_t \qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad xs'_s \approx xs'_t$$

RelSrcCons

$$\frac{\forall \ell.\, \ell \mapsto_s (\mathrm{CONS}, v_1, v_2) \mathrel{-\!\!*} \ell \gtrsim e_t \ [\Phi]}{v_1 \ :: \ v_2 \gtrsim e_t \ [\Phi]}$$

$$y_s \ :: \ ys_s \qquad \gtrsim \qquad \ell_t$$

Proof sketch

$$
\frac{\begin{array}{c} f_s \approx f_t \qquad\qquad\qquad xs_s \approx xs_t \\ x_s \approx x_t \qquad\qquad\qquad xs'_s \approx xs'_t \\ y_s \approx y_t \\ ys_s \approx ys_t \\ \ell_t \mapsto_t (\mathrm{CONS}, y_t, ys_t) \\ \ell_s \mapsto_s (\mathrm{CONS}, y_s, ys_s) \end{array}}{\ell_s \qquad\qquad\qquad\qquad \gtrsim \;\; \ell_t}
$$

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

$$\frac{\begin{array}{l}\textsc{RelBijInsert}\\ \quad \ell_s \mapsto_s \overline{v_s} \\ \quad \ell_t \mapsto_t \overline{v_t} \\ \quad \overline{v_s} \approx \overline{v_t} \\ \ell_s \approx \ell_t \twoheadrightarrow\!\!\ast\; e_s \gtrsim e_t \;[\Phi]\end{array}}{e_s \gtrsim e_t \;[\Phi]}$$

$$\ell_s \qquad\qquad \gtrsim \quad \ell_t$$

Proof sketch

$$f_s \approx f_t \qquad\qquad xs_s \approx xs_t$$

$$x_s \approx x_t \qquad\qquad xs'_s \approx xs'_t$$

$$y_s \approx y_t$$

$$ys_s \approx ys_t$$

$$\ell_s \approx \ell_t$$

ℓ_s $\gtrsim$ ℓ_t

Thank you for your attention!