Functional In-place Update with Layered Datatype Sharing

Michal Konečný
LFCS, University of Edinburgh

part of a wider collaboration with
David Aspinall, Martin Hofmann, Robert Atkey

Overview

- *LFPL* (Linear Functional Programming Language) [Hofmann 2000]
- functional language $\Rightarrow$ neat reasoning about programs
- resource type $\Diamond$ $\Rightarrow$ simple and efficient evaluation using heap

Overview

- *LFPL* (Linear Functional Programming Language) [Hofmann 2000]
- functional language $\Rightarrow$ neat reasoning about programs
- resource type $\Diamond$ $\Rightarrow$ simple and efficient evaluation using heap
- *linear typing* guarantees correctness of this evaluation
  forbids sharing on the heap

Overview

- *LFPL* (Linear Functional Programming Language) [Hofmann 2000]
- functional language $\Rightarrow$ neat reasoning about programs
- resource type $\Diamond$ $\Rightarrow$ simple and efficient evaluation using heap
- *linear typing* guarantees correctness of this evaluation
  forbids sharing on the heap
- developing less restrictive typings for LFPL
  - *usage aspects* (Aspinall & Hofmann 2002)
  - *explicit sharing in the context* (Atkey)
  - *layered datatype sharing* (K 2002)
- Scope: first-order, full recursion, arbitrary inductive datatypes

Append in LFPL

In ML :

$$
\begin{aligned}
\mathit{append}(x, y) = \ & \text{match } x \text{ with} \\
& \quad \text{Nil} \to y \\
& \quad \mid \text{Cons}(h, t) \quad \to \text{Cons}(h, \mathit{append}(t, y))
\end{aligned}
$$

$$h : A, t : \mathsf{L}(A) \quad \vdash \text{Cons}(h, t) \quad : \mathsf{L}(A)$$

Append in LFPL

In LFPL:

$$
\begin{aligned}
\mathit{append}(x, y) = \ & \text{match } x \text{ with} \\
& \quad \text{Nil} \to y \\
& \quad \mid \text{Cons}(h, t)@d \to \text{Cons}(h, \mathit{append}(t, y))@d
\end{aligned}
$$

$$h : A, t : \mathsf{L}(A), d : \Diamond \vdash \text{Cons}(h, t)@d : \mathsf{L}(A)$$

elements of $\Diamond$: units of heap space/heap locations

Append in LFPL

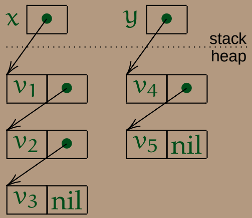
In LFPL:

$$\begin{aligned} \mathit{append}(x, y) = \ & \text{match } x \text{ with} \\ & \text{Nil} \to y \\ & \mid \text{Cons}(h, t)@d \to \text{Cons}(h, \mathit{append}(t, y))@d \end{aligned}$$

$$h : A, t : \mathsf{L}(A), d : \Diamond \vdash \text{Cons}(h, t)@d : \mathsf{L}(A)$$

elements of $\Diamond$: units of heap space/heap locations

Heap representation:



Append in LFPL

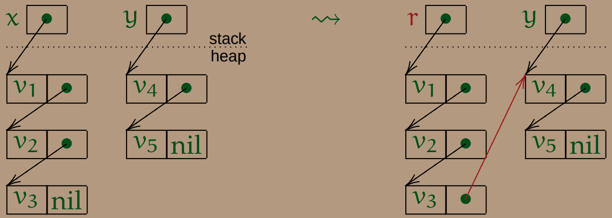
In LFPL:

$$\begin{aligned} \mathit{append}(x,y) = \ & \text{match } x \text{ with} \\ & \quad \text{Nil} \to y \\ & \quad \mid \text{Cons}(h,t)@d \to \text{Cons}(h, \mathit{append}(t,y))@d \end{aligned}$$

$$h : A, t : \mathsf{L}(A), d : \Diamond \vdash \text{Cons}(h,t)@d : \mathsf{L}(A)$$

elements of $\Diamond$: units of heap space/heap locations

Heap representation:



Non-linear typings

In LFPL:
- Every use of a variable is considered destructive
- No heap-sharing between arguments

Non-linear typings

In LFPL: – Every use of a variable is considered destructive

– No heap-sharing between arguments

LFPL with Usage aspects (UAPL, Aspinall & Hofmann 2002):

$x :^1 \mathsf{L}(A), y :^2 \mathsf{L}(A) \vdash \mathit{append}(x, y) : \mathsf{L}(A)$ $\qquad x :^3 \mathsf{L}(A) \vdash \mathit{length}(x) : \mathsf{Nat}$

1 = destructive, 2 = read-only, 3 = read-only & not sharing with the result

Non-linear typings

In LFPL: – Every use of a variable is considered destructive

– No heap-sharing between arguments

LFPL with Usage aspects (UAPL, Aspinall & Hofmann 2002):

$x :^1 \mathsf{L}(A), y :^2 \mathsf{L}(A) \vdash \mathit{append}(x, y) : \mathsf{L}(A)$ $\quad$ $x :^3 \mathsf{L}(A) \vdash \mathit{length}(x) : \mathsf{Nat}$

1 = destructive, 2 = read-only, 3 = read-only & not sharing with the result

expression	eval?	LFPL
$\mathit{append}(x, x)$	N	N
$\mathsf{Cons}(h, \mathsf{Cons}(h, t)@d1)@d2$	Y	N
$\mathsf{Cons}(\mathit{length}(x), x)@d$	Y	N
$\mathit{append}(\mathsf{Cons}(h, t1)@d1, \mathsf{Cons}(h, t2)@d2)$	Y	N

Non-linear typings

In LFPL:
- Every use of a variable is considered destructive
- No heap-sharing between arguments

LFPL with Usage aspects (UAPL, Aspinall & Hofmann 2002):

$x :^1 \mathsf{L}(A), y :^2 \mathsf{L}(A) \vdash \mathit{append}(x, y) : \mathsf{L}(A)$ $\quad$ $x :^3 \mathsf{L}(A) \vdash \mathit{length}(x) : \mathsf{Nat}$

1 = destructive, 2 = read-only, 3 = read-only & not sharing with the result

expression	eval?	LFPL	UAPL
$\mathit{append}(x, x)$	N	N	N
$\mathsf{Cons}(h, \mathsf{Cons}(h, t)@d1)@d2$	Y	N	Y
$\mathsf{Cons}(\mathit{length}(x), x)@d$	Y	N	Y
$\mathit{append}(\mathsf{Cons}(h, t1)@d1, \mathsf{Cons}(h, t2)@d2)$	Y	N	N

Non-linear typings

In LFPL: – Every use of a variable is considered destructive
– No heap-sharing between arguments

LFPL with Usage aspects (UAPL, Aspinall & Hofmann 2002):

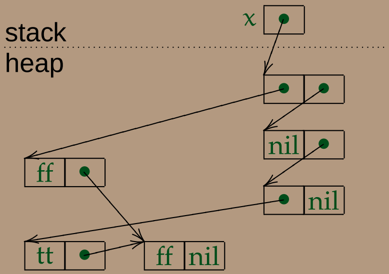
$x :^1 \mathsf{L}(A), y :^2 \mathsf{L}(A) \vdash \mathit{append}(x, y) : \mathsf{L}(A)$ $\quad$ $x :^3 \mathsf{L}(A) \vdash \mathit{length}(x) : \mathsf{Nat}$

1 = destructive, 2 = read-only, 3 = read-only & not sharing with the result

expression	eval?	LFPL	UAPL	DEEL
$\mathit{append}(x, x)$	N	N	N	N
$\mathsf{Cons}(h, \mathsf{Cons}(h, t)@d1)@d2$	Y	N	Y	Y
$\mathsf{Cons}(\mathit{length}(x), x)@d$	Y	N	Y	Y
$\mathit{append}(\mathsf{Cons}(h, t1)@d1, \mathsf{Cons}(h, t2)@d2)$	Y	N	N	Y

Datatype portions (layers)

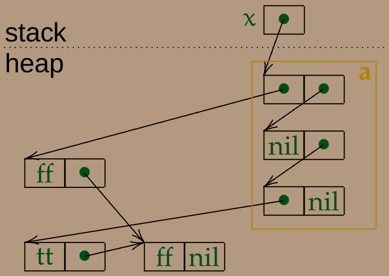
$$\mathsf{L}(A) = \mu X.\mathsf{Unit} + {}^{\Diamond}(A \times X)$$



$$[\![x]\!] = [[ff, ff], [], [tt, ff]]$$

Datatype portions (layers)

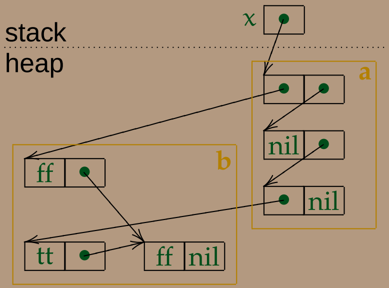
$$\mathsf{L}(A) = \mu X.\mathsf{Unit} + {}^{\Diamond}(A \times X)$$



$$[\![x]\!] = [[\mathit{ff}, \mathit{ff}], [], [\mathit{tt}, \mathit{ff}]]$$

Datatype portions (layers)

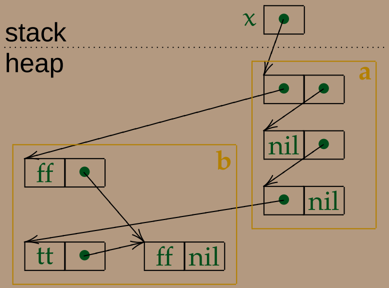
$$\mathsf{L}(A) = \mu X.\mathsf{Unit} + {}^{\Diamond}(A \times X)$$



$$[\![x]\!] = [[ff, ff], [], [tt, ff]]$$

Datatype portions (layers)

$$\mathsf{L}(A) = \mu X.\mathsf{Unit} + {}^{\Diamond}(A \times X)$$

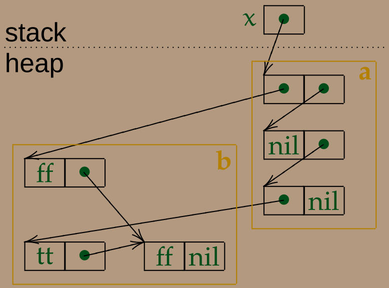


$$[\![x]\!] = [[ff, ff], [], [tt, ff]]$$

$$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool}))$$

Datatype portions (layers)

$$\mathsf{L}(A) = \mu \mathsf{X}.\mathsf{Unit} + {}^{\Diamond}(A \times \mathsf{X})$$



$$[\![x]\!] = [[\mathit{ff}, \mathit{ff}], [], [\mathit{tt}, \mathit{ff}]]$$

$$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool}))$$

$$\mathsf{L}^{[a]}(A) = \mu \mathsf{X}^{(a)}.\mathsf{Unit} + {}^{\Diamond[a]}(A \times \mathsf{X})$$

Basic Separation Assertions

notation	type pattern	name
$\mathbf{a} \otimes \mathbf{b}$	$\ldots^{\Diamond[\mathbf{a}]} \ldots^{\Diamond[\mathbf{b}]} \ldots$	$\mathbf{a}$ separated from $\mathbf{b}$

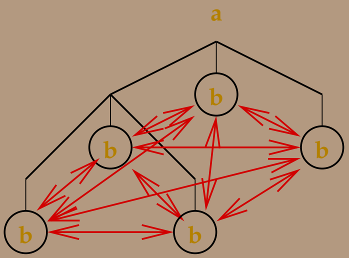
Basic Separation Assertions

notation	type pattern	name
$a \otimes b$	$\ldots \Diamond^{[a]} \ldots \Diamond^{[b]} \ldots$	a separated from b
$\otimes b / a$	$\ldots \mu X^{(a)}.(\ldots \Diamond^{[b]} \ldots) \ldots$	b separated along a
$\otimes b \curlywedge a$	$\ldots \mu X^{(a)}.(\ldots \Diamond^{[b]} \ldots) \ldots$	b separated across a

Basic Separation Assertions

notation	type pattern	name
$a \otimes b$	$\ldots^{\Diamond[a]} \ldots^{\Diamond[b]} \ldots$	a separated from b
$\otimes b/a$	$\ldots \mu X^{(a)}.(\ldots^{\Diamond[b]} \ldots) \ldots$	b separated along a
$\otimes b \curlywedge a$	$\ldots \mu X^{(a)}.(\ldots^{\Diamond[b]} \ldots) \ldots$	b separated across a

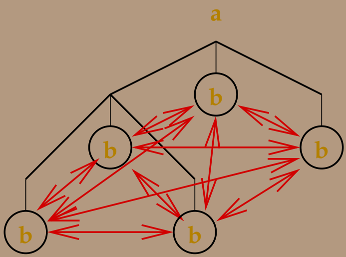
$\otimes b/a$



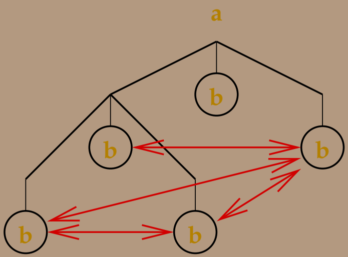
Basic Separation Assertions

notation	type pattern	name
$a \otimes b$	$\ldots^{\Diamond[a]} \ldots^{\Diamond[b]} \ldots$	a separated from b
$\otimes b / a$	$\ldots \mu X^{(a)}.(\ldots^{\Diamond[b]} \ldots) \ldots$	b separated along a
$\otimes b \curlywedge a$	$\ldots \mu X^{(a)}.(\ldots^{\Diamond[b]} \ldots) \ldots$	b separated across a

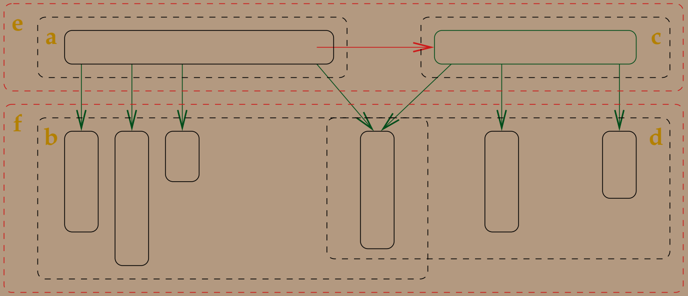
$\otimes b / a$



$\otimes b \curlywedge a$



Typing of append



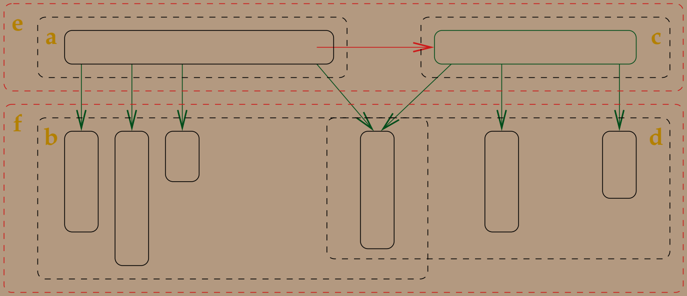
$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$ argument portions

$\vdash$

$append(x, y) : \mathsf{L}^{[e\quad]}(\mathsf{L}^{[f\quad]}(\mathsf{Bool}));$ result portions

;

Typing of append



$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$ argument portions

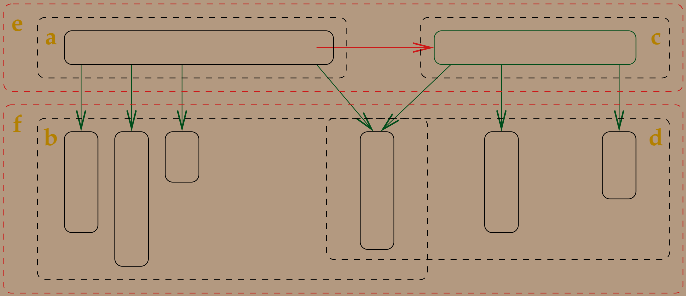
$\vdash$

$\mathit{append}(x, y) : \mathsf{L}^{[e\quad]}(\mathsf{L}^{[f\quad]}(\mathsf{Bool}));$ result portions

$\{a\}$ destroyed portions

;

Typing of append



$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$ argument portions

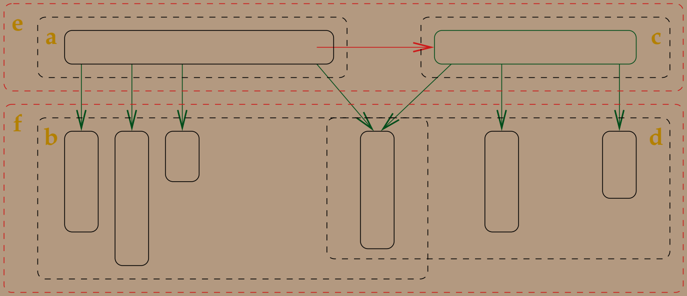
$\vdash$

$\mathit{append}(x, y) : \mathsf{L}^{[e \subseteq \{a,c\}]}(\mathsf{L}^{[f \subseteq \{b,d\}]}(\mathsf{Bool}));$ result portions, containment

$\{a\}$ destroyed portions

;

Typing of append



$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$ argument portions

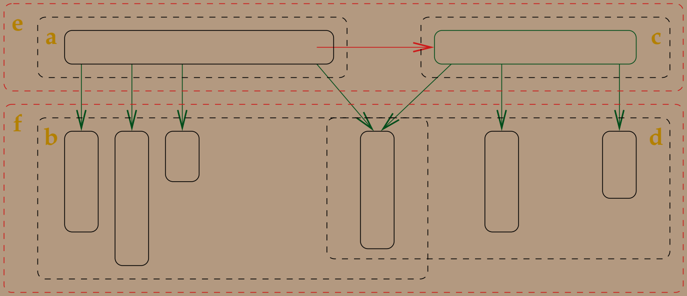
$\{a \otimes b, a \otimes c, a \otimes d\} \vdash$ separation pre-condition

$\mathit{append}(x, y) : \mathsf{L}^{[e \subseteq \{a,c\}]}(\mathsf{L}^{[f \subseteq \{b,d\}]}(\mathsf{Bool}));$ result portions, containment

$\{a\}$ destroyed portions

;

Typing of append



$x : L^{[a]}(L^{[b]}(\mathsf{Bool})), y : L^{[c]}(L^{[d]}(\mathsf{Bool}));$	argument portions
$\{a \otimes b, a \otimes c, a \otimes d\} \vdash$	separation pre-condition
$\mathit{append}(x, y) : L^{[e \subseteq \{a,c\}]}(L^{[f \subseteq \{b,d\}]}(\mathsf{Bool}));$	result portions, containment
$\{a\}$	destroyed portions
$\otimes f/e \Longleftarrow \{b \otimes d, \otimes b/a, \otimes d/c\}$;	separation rely-guarantees

A typing rule

[CONS]

$$B_h = A\left[\frac{\delta}{\rho}\right] \qquad B_t = \mathsf{L}^{[\zeta]}\left(A\left[\frac{\delta'}{\rho'}\right]\right)$$

$h : B_h, t : B_t, d : \Diamond^{[\zeta_d]};$

$\vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: \quad ;$ *containment guarantees*

$;$ *destroyed portions*

A typing rule

[CONS]

$$B_h = A[\tfrac{\delta}{\rho}] \qquad B_t = \mathsf{L}^{[\zeta]}(A[\tfrac{\delta'}{\rho'}])$$

$h : B_h, t : B_t, d : \lozenge^{[\zeta_d]};$

$\vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: \quad ;$ *containment guarantees*

$\{\zeta_d\};$ *destroyed portions*

A typing rule

[CONS]

$$B_h = A[\frac{\delta}{\rho}] \qquad B_t = \mathsf{L}^{[\zeta]}(A[\frac{\delta'}{\rho'}])$$

$h : B_h, t : B_t, d : \Diamond^{[\zeta_d]};$

$\{\zeta_d\} \otimes \big(N_D(B_h) \cup N_D(B_t)\big) \vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: \quad ;$ *containment guarantees*

$\{\zeta_d\}\,;$ *destroyed portions*

A typing rule

[CONS]

$$B_h = A\left[\tfrac{\delta}{\rho}\right] \qquad B_t = \mathsf{L}^{[\zeta]}\left(A\left[\tfrac{\delta'}{\rho'}\right]\right)$$

$$E = \mathsf{L}^{[\zeta' \subseteq \{\zeta_d\}]}(E_h) \cup E_t$$

$h : B_h, t : B_t, d : \Diamond^{[\zeta_d]};$

$\{\zeta_d\} \otimes \big(N_D(B_h) \cup N_D(B_t)\big) \vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: E;$ *containment guarantees*

$\{\zeta_d\};$ *destroyed portions*

A typing rule

[CONS]

$$\frac{B_h = A\left[\frac{\delta}{\rho}\right] \qquad B_t = L^{[\zeta]}\left(A\left[\frac{\delta'}{\rho'}\right]\right) \qquad E = L^{[\zeta' \subseteq \{\zeta_d\}]}(E_h) \cup E_t \qquad E_h \in E_Y(B_h) \qquad E_t \in E_Y(B_t)}{h : B_h, t : B_t, d : \Diamond^{[\zeta_d]};}$$

$\{\zeta_d\} \otimes \big(N_D(B_h) \cup N_D(B_t)\big) \vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: \; E\,;$ *containment guarantees*

$\{\zeta_d\}\,;$ *destroyed portions*

A typing rule

[CONS]

$$\frac{B_h = A\left[\tfrac{\delta}{\rho}\right] \qquad B_t = \mathsf{L}^{[\zeta]}\left(A\left[\tfrac{\delta'}{\rho'}\right]\right) \qquad\qquad E = \mathsf{L}^{[\zeta' \subseteq \{\zeta_d\}]}(E_h) \cup E_t \qquad E_h \in E_Y(B_h) \qquad E_t \in E_Y(B_t)}{h : B_h, t : B_t, d : \Diamond^{[\zeta_d]};}$$

$\{\zeta_d\} \otimes \big(N_D(B_h) \cup N_D(B_t)\big) \ \vdash$ *separation pre-condition*

$\mathsf{Cons}(h, t)@d$

$: \ E \ ;$ *containment guarantees*

$\{\zeta_d\} \ ;$ *destroyed portions*

G *separation rely-guarantees*

where $G = G_M(E) \cup G_Y(E_h, B_h) \cup G_Y(E_t, B_t)$
$\cup \left[\otimes \zeta_\xi / \zeta' \Longleftarrow \{\delta(\xi) \otimes \delta'(\xi)\}\right]_{\xi \in \mathrm{AddrD}(A)}$

Typing a recursive program

- annotating expressions - need annotated function symbols

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic:

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\varnothing \vdash$ *separation pre-condition*

match x with
  Nil $\rightarrow y$
  | Cons$(h, t)@d \rightarrow$ let $z =$ Cons$(h, y)@d$
    in $append(t, z)$

$: \mathsf{L}^{[e \subseteq \varnothing]}(\mathsf{L}^{[f \subseteq \varnothing]}(\mathsf{Bool}));$ *containment guarantees*

$\varnothing;$ *destroyed portions*

$\otimes \mathbf{f}/\mathbf{e} \Longleftarrow \varnothing$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression

$t : L^{[a]}(L^{[b]}(\mathsf{Bool})),\ z : L^{[c]}(L^{[d]}(\mathsf{Bool}));$

$\varnothing \vdash$ *separation pre-condition*

$\text{match } x \text{ with}$

$\quad \text{Nil} \to y$

$\quad | \text{ Cons}(h, t)@d \to \text{ let } z = \text{ Cons}(h, y)@d$

$\qquad\qquad\qquad\qquad \text{in } \mathit{append}(t, z)$

$: L^{[e \subseteq \varnothing]}(L^{[f \subseteq \varnothing]}(\mathsf{Bool}));$ *containment guarantees*

$\varnothing;$ *destroyed portions*

$\otimes \mathbf{f}/\mathbf{e} \Longleftarrow \varnothing$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression

$h : L^{[a]}(\text{Bool}),\ y : L^{[b]}(L^{[c]}(\text{Bool})),\ d : \Diamond^{[d]};$

$\{a \otimes d, b \otimes d, c \otimes d\} \vdash$ *separation pre-condition*

match x with

Nil $\to y$

$\mid$ Cons$(h, t)@d \to$ let $z =$ Cons$(h, y)@d$

in $append(t, z)$

$: L^{[e \subseteq \{b,d\}]}(L^{[f \subseteq \{a,c\}]}(\text{Bool}));$ *containment guarantees*

$\{d\};$ *destroyed portions*

$\otimes f/e \Longleftarrow \{a \otimes c, \otimes c/b\}$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression

$h : L^{[a]}(\mathrm{Bool}),\ y : L^{[b]}(L^{[c]}(\mathrm{Bool})),\ d : \Diamond^{[d]},\ t : L^{[e]}(L^{[f]}(\mathrm{Bool}))$

$\{a \otimes d, b \otimes d, c \otimes d, d \otimes e, d \otimes f\} \vdash$ *separation pre-condition*

match x with
Nil $\to y$
$|$ Cons$(h, t)@d \to$ let $z =$ Cons$(h, y)@d$
in $append(t, z)$

$: L^{[g \subseteq \emptyset]}(L^{[h \subseteq \emptyset]}(\mathrm{Bool}));$ *containment guarantees*

$\{d\};$ *destroyed portions*

$\otimes h/g \Longleftarrow \emptyset$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\{\mathbf{a} \otimes \mathbf{c}, \mathbf{a} \otimes \mathbf{d}\} \vdash$ *separation pre-condition*

match x with
  Nil $\rightarrow y$
  | Cons$(h, t)@d \rightarrow$ let $z =$ Cons$(h, y)@d$
  in $append(t, z)$

$: \mathsf{L}^{[e \subseteq \{c\}]}(\mathsf{L}^{[f \subseteq \{d\}]}(\mathsf{Bool}));$ *containment guarantees*

$\{\mathbf{a}\};$ *destroyed portions*

$\otimes \mathbf{f}/\mathbf{e} \Longleftarrow \{\otimes \mathbf{d}/\mathbf{c}\}$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression and repeat

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\{\mathbf{a}\otimes\mathbf{b}, \mathbf{a}\otimes\mathbf{c}, \mathbf{a}\otimes\mathbf{d}\} \vdash$ *separation pre-condition*

match x with
  Nil $\rightarrow y$
  | Cons(h, t)@$d \rightarrow$ let $z =$ Cons(h, y)@d
      in $\mathit{append}(t, z)$

$: \mathsf{L}^{[e\subseteq\{a,c\}]}(\mathsf{L}^{[f\subseteq\{b,d\}]}(\mathsf{Bool}));$ *containment guarantees*

$\{\mathbf{a}\};$ *destroyed portions*

$\otimes\mathbf{f}/\mathbf{e} \Longleftarrow \{\mathbf{b}\otimes\mathbf{d}, \otimes\mathbf{d}/\mathbf{c}\}$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression and repeat until fixpoint is found

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\{\mathbf{a}\otimes\mathbf{b}, \mathbf{a}\otimes\mathbf{c}, \mathbf{a}\otimes\mathbf{d}\} \vdash$ *separation pre-condition*

match x with

  Nil $\longrightarrow y$

  | Cons(h, t)@$d \longrightarrow$ let $z =$ Cons(h, y)@d

  in $\mathit{append}(t, z)$

$: \mathsf{L}^{[e\subseteq\{a,c\}]}(\mathsf{L}^{[f\subseteq\{b,d\}]}(\mathsf{Bool}));$ *containment guarantees*

$\{\mathbf{a}\};$ *destroyed portions*

$\otimes\mathbf{f}/\mathbf{e} \Longleftarrow \{\mathbf{b}\otimes\mathbf{d}, \otimes\mathbf{b}/\mathbf{a}, \otimes\mathbf{d}/\mathbf{c}\}$ *separation rely-guarantees*

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression and repeat until fixpoint is found

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\{a \otimes b, a \otimes c, a \otimes d\} \vdash$ *separation pre-condition*

match x with

Nil $\rightarrow y$

$|$ Cons$(h, t)@d \rightarrow$ let $z =$ Cons$(h, y)@d$

in $append(t, z)$

$: \mathsf{L}^{[e \subseteq \{a,c\}]}(\mathsf{L}^{[f \subseteq \{b,d\}]}(\mathsf{Bool}));$ *containment guarantees*

$\{a\};$ *destroyed portions*

$\otimes f/e \Longleftarrow \{b \otimes d, \otimes b/a, \otimes d/c\}$ *separation rely-guarantees*

- Typing rules need to be – monotone

Typing a recursive program

- annotating expressions - need annotated function symbols
- initial annotation - optimistic
- then annotate expression and repeat until fixpoint is found

$x : \mathsf{L}^{[a]}(\mathsf{L}^{[b]}(\mathsf{Bool})), y : \mathsf{L}^{[c]}(\mathsf{L}^{[d]}(\mathsf{Bool}));$

$\{\mathbf{a}\otimes\mathbf{b}, \mathbf{a}\otimes\mathbf{c}, \mathbf{a}\otimes\mathbf{d}\} \vdash$ *separation pre-condition*

match x with

Nil $\rightarrow y$

$|$ Cons(h, t)@$d \rightarrow$ let $z =$ Cons(h, y)@d

in $append(t, z)$

$: \mathsf{L}^{[\mathbf{e}\subseteq\{\mathbf{a},\mathbf{c}\}]}(\mathsf{L}^{[\mathbf{f}\subseteq\{\mathbf{b},\mathbf{d}\}]}(\mathsf{Bool}));$ *containment guarantees*

$\{\mathbf{a}\};$ *destroyed portions*

$\otimes\mathbf{f}/\mathbf{e} \Longleftarrow \{\mathbf{b}\otimes\mathbf{d}, \otimes\mathbf{b}/\mathbf{a}, \otimes\mathbf{d}/\mathbf{c}\}$ *separation rely-guarantees*

- Typing rules need to be – monotone
  – stable under strengthening of premises

Conclusion

- powerful and efficient static analysis of functional in-place udate
- implemented in Haskell

Conclusion

- powerful and efficient static analysis of functional in-place udate
- implemented in Haskell
- Future: – Infer formal proofs of correctness (ala Necula&Lee)

Conclusion

- powerful and efficient static analysis of functional in-place udate
- implemented in Haskell
- Future:
  - Infer formal proofs of correctness (ala Necula&Lee)
  - Infer *resource usage approximations*? (Hofmann&Jost)

Conclusion

- powerful and efficient static analysis of functional in-place udate
- implemented in Haskell
- Future:
  - Infer formal proofs of correctness (ala Necula&Lee)
  - Infer *resource usage approximations*? (Hofmann&Jost)
  - Scope: allocation, higher order, arrays