
L^3

A Linear Language with Locations

Greg Morrisett

Harvard University

greg@eecs.harvard.edu

Amal Ahmed

Cornell University

aahmed@cs.cornell.edu

Matthew Fluet

Cornell University

fluet@cs.cornell.edu

Introduction

- Explore foundational typing support for *strong updates*

Introduction

- Explore foundational typing support for *strong updates*

```
1. let
2.   val r = ref ()
3. in
4.   r := true;
5.   if (!r) then
6.     r := 42
7.   else
8.     r := 15;
9.   !r + 12
10. end
```

Introduction

- Explore foundational typing support for *strong updates*

```
1. fun f (r1:int ref,r2:int ref):int =  
2.   (r1 := true;  
3.    !r2 + 42)
```

Introduction

- Explore foundational typing support for *strong updates*
 - Control over aliasing
 - Reflect side-effects in the interface of functions

Introduction

- Explore foundational typing support for *strong updates*
- A core calculus based on standard linear logic
- Yields an extremely clean semantic interpretation

Linearity and Strong Updates

- Linear objects
 - cannot be duplicated
 - there are no aliases to the object

L³: Syntax

LocConsts ℓ \in *LocConsts*

LocVars ρ \in *LocVars*

Locs η ::= $\ell \mid \rho$

Types

τ ::= $1 \mid \tau_1 \otimes \tau_2 \mid \tau_1 \multimap \tau_2 \mid !\tau \mid$

Ptr η | Cap $\eta \tau$ | $\forall \rho. \tau$ | $\exists \rho. \tau$

L³: Syntax

Exprs

$$\begin{aligned} e ::= & \langle \rangle \mid \text{let } \langle \rangle = e_1 \text{ in } e_2 \mid \\ & \langle e_1, e_2 \rangle \mid \text{let } \langle x_1, x_2 \rangle = e_1 \text{ in } e_2 \mid \\ & x \mid \lambda x. e \mid e_1 e_2 \mid \\ & !v \mid \text{let } !x = e_1 \text{ in } e_2 \mid \text{dup } e \mid \text{drop } e \mid \\ & \text{ptr } \ell \mid \text{cap } \ell \mid \\ & \text{create } e \mid \text{destroy } e \mid \text{swap } e_1 e_2 e_3 \mid \\ & \Lambda \rho. e \mid e[\eta] \mid {}^\Gamma \eta, e^\rhd \mid \text{let } {}^\Gamma \rho, x^\rhd = e_1 \text{ in } e_2 \end{aligned}$$

L³: Operational Semantics

(let-bang) $(\sigma, \text{let } !x = !v \text{ in } e) \longmapsto (\sigma, e[v/x])$

(dup) $(\sigma, \text{dup } !v) \longmapsto (\sigma, \langle !v, !v \rangle)$

(drop) $(\sigma, \text{drop } !v) \longmapsto (\sigma, \langle \rangle)$

L³: Operational Semantics

Stores $\sigma ::= \{\ell_1 \mapsto v_1, \dots, \ell_n \mapsto v_n\}$

- (cre) $(\sigma, \text{create } v)$
 $\longmapsto (\sigma \uplus \{\ell \mapsto v\}, \lceil \ell, \langle \text{cap } \ell, !(\text{ptr } \ell) \rangle \rceil)$
- (des) $(\sigma \uplus \{\ell \mapsto v\}, \text{destroy } \lceil \ell, \langle \text{cap } \ell, !(\text{ptr } \ell) \rangle \rceil)$
 $\longmapsto (\sigma, \lceil \ell, v \rceil)$
- (swap) $(\sigma \uplus \{\ell \mapsto v_1\}, \text{swap } (\text{cap } \ell) (\text{ptr } \ell) v_2)$
 $\longmapsto (\sigma \uplus \{\ell \mapsto v_2\}, \langle \text{cap } \ell, v_1 \rangle)$

L³: Static Semantics

$$\boxed{\Delta; \Gamma \vdash e : \tau}$$

$$(\text{Bang}) \quad \frac{\Delta; \Gamma \vdash v : \tau \quad |\Gamma| = \bullet}{\Delta; \Gamma \vdash !v : !\tau}$$

$$(\text{Let-Bang}) \quad \frac{\Delta; \Gamma_1 \vdash e_1 : !\tau_1 \quad \Delta; \Gamma_2, x:\tau_1 \vdash e_2 : \tau_2}{\Delta; \Gamma_1 \boxplus \Gamma_2 \vdash \text{let } !x = e_1 \text{ in } e_2 : \tau_2}$$

$$(\text{Dup}) \quad \frac{\Delta; \Gamma \vdash e : !\tau}{\Delta; \Gamma \vdash \text{dup } e : !\tau \otimes !\tau}$$

$$(\text{Drop}) \quad \frac{\Delta; \Gamma \vdash e : !\tau}{\Delta; \Gamma \vdash \text{drop } e : \mathbf{1}}$$

L³: Static Semantics

$$\boxed{\Delta; \Gamma \vdash e : \tau}$$

$$(\text{Create}) \quad \frac{\Delta; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash \text{create } e : \exists \rho. (\text{Cap } \rho \tau \otimes !(\text{Ptr } \rho))}$$

$$(\text{Destroy}) \quad \frac{\Delta; \Gamma \vdash e : \exists \rho. (\text{Cap } \rho \tau \otimes !(\text{Ptr } \rho))}{\Delta; \Gamma \vdash \text{destroy } e : \exists \rho. \tau}$$

$$\Delta; \Gamma_1 \vdash e_1 : \text{Cap } \rho \tau_1 \quad \Delta; \Gamma_2 \vdash e_2 : \text{Ptr } \rho$$

$$(\text{Swap}) \quad \frac{\Delta; \Gamma_3 \vdash e_3 : \tau_2}{\Delta; \Gamma_1 \boxplus \Gamma_2 \boxplus \Gamma_3 \vdash \text{swap } e_1 e_2 e_3 : \text{Cap } \rho \tau_2 \otimes \tau_1}$$

L³: Examples

$\text{LRef } \tau \equiv \exists \rho. (\text{Cap } \rho \tau \otimes \text{!Ptr } \rho),$

$\text{lrswap} \equiv \lambda r:\text{LRef } \tau. \lambda x:\tau'.$

let $\lceil \rho, \text{cp} \rceil = r$ in

let $\langle c_0, p_0 \rangle = \text{cp}$ in

let $\langle p_1, p_2 \rangle = \text{dup } p_0$ in

let $\text{!}p'_2 = p_2$ in

let $\langle c_1, y \rangle = \text{swap } c_0 p'_2 x$ in

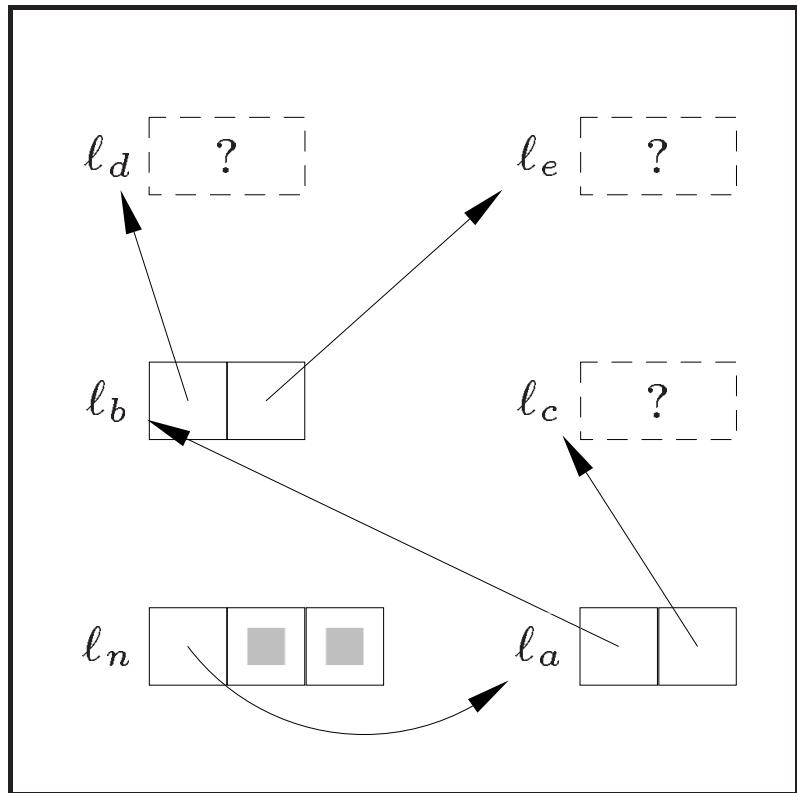
$\langle \lceil \rho, \langle c_1, p_1 \rangle \rceil, y \rangle$

L³: Examples

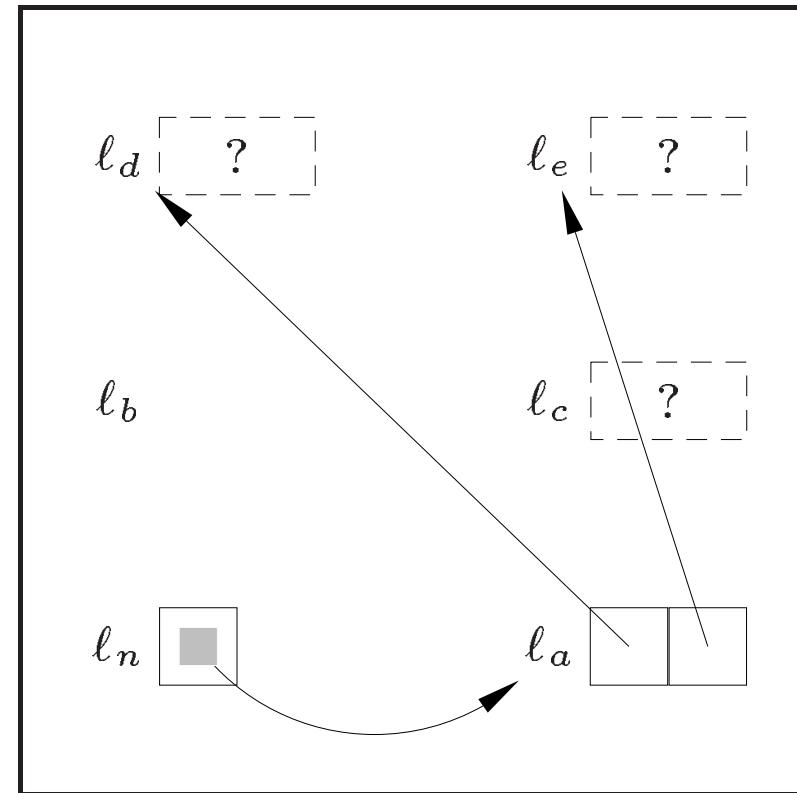
```
let  $\lceil \rho_n, \langle c_n^0, p_n^! \rangle \rceil = \text{create} \langle \rangle \text{ in}$ 
let  $\text{nuke}^! = !\Lambda \rho_a, \rho_b, \rho_c, \rho_d, \rho_e. \lambda c_n^0.$ 
    let  $\langle c_n^1, \langle p_a^!, \langle c_a^0, c_b^0 \rangle \rangle \rangle = \text{swap } c_n^0 \text{ } p_n \langle \rangle \text{ in}$ 
    let  $\langle c_a^1, \langle p_b^!, p_c^! \rangle \rangle = \text{swap } c_a^0 \text{ } p_a \langle \rangle \text{ in}$ 
    let  $\langle p_d^!, p_e^! \rangle = \text{destroy} \lceil \rho_b, \langle c_b^0, p_b \rangle \rceil \text{ in}$ 
    let  $\langle c_a^2, \langle \rangle \rangle = \text{swap } c_a^1 \text{ } p_a \langle p_d, p_e \rangle \text{ in}$ 
    let  $\langle c_n^2, \langle \rangle \rangle = \text{swap } c_n^1 \text{ } p_n \text{ } c_a^2 \text{ in}$ 
     $c_n^2 \text{ in}$ 

 $c_n^0 :: \text{Cap } \rho_n (!(\text{Ptr } \rho_a) \otimes (\text{Cap } \rho_a (!(\text{Ptr } \rho_b) \otimes !(\text{Ptr } \rho_c)) \otimes$ 
 $\text{Cap } \rho_b (!(\text{Ptr } \rho_d) \otimes !(\text{Ptr } \rho_e))))$ 
```

L³: Examples

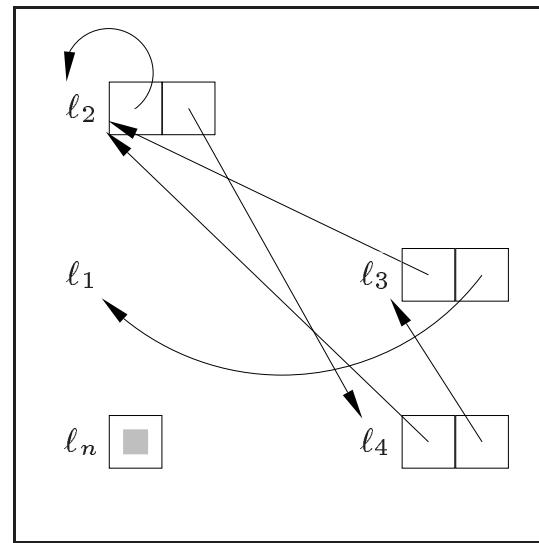
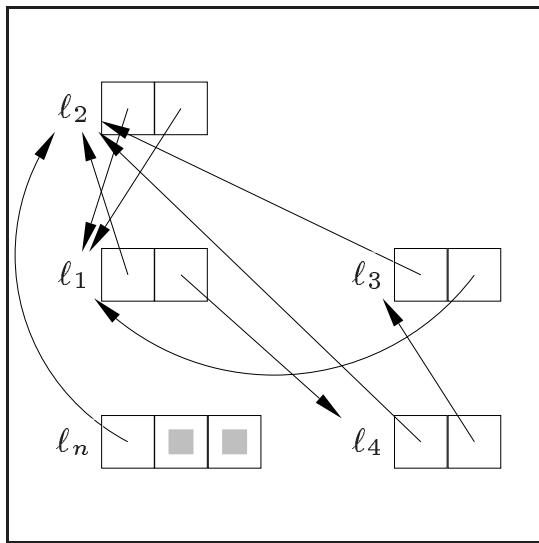
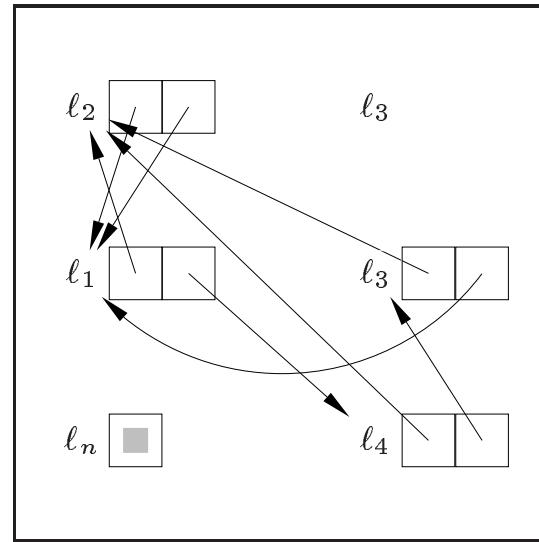
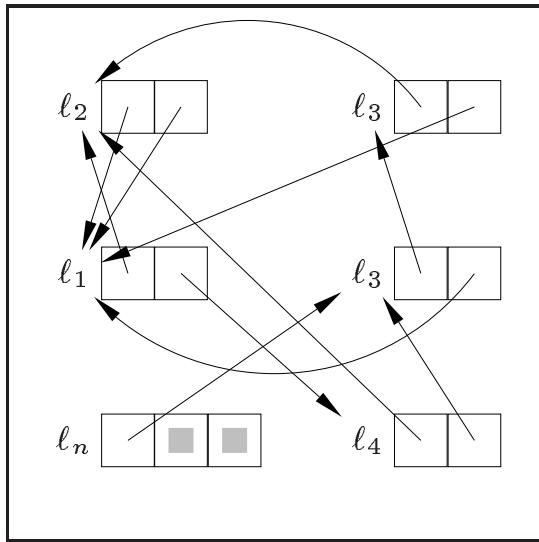


Store into nuke



Store out of nuke

L^3 : Examples



L³: Semantic Interpretation

$$\llbracket \Delta; \Gamma \vdash e : \tau \rrbracket = \forall \delta, \sigma, \gamma. \delta \in \mathcal{D}[\![\Delta]\!] \wedge (\sigma, \gamma) \in \mathcal{G}[\![\Gamma]\!] \delta \Rightarrow \\ (\sigma, \gamma(\delta(e))) \in \mathcal{C}[\![\delta(\tau)]\!]$$

$$\mathcal{C}[\![\tau]\!] = \{(\sigma_s, e_s) \mid \forall \sigma_r. \sigma_s \uplus \sigma_r \text{ defined } \Rightarrow \\ \exists n, \sigma_f, v_f. (\sigma_s \uplus \sigma_r, e_s) \longmapsto^n (\sigma_f \uplus \sigma_r, v_f) \wedge \\ (\sigma_f, v_f) \in \mathcal{V}[\![\tau]\!]\}$$

L³: Semantic Interpretation

$$\mathcal{V}[\![1]\!] = \{(\{\}, \langle \rangle)\}$$

$$\begin{aligned}\mathcal{V}[\![\tau_1 \otimes \tau_2]\!] = & \{(\sigma_1 \uplus \sigma_2, \langle v_1, v_2 \rangle) \mid (\sigma_1, v_1) \in \mathcal{V}[\![\tau_1]\!] \wedge \\ & (\sigma_2, v_2) \in \mathcal{V}[\![\tau_2]\!]\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\tau_1 \multimap \tau_2]\!] = & \{(\sigma_2, \lambda x. e) \mid \forall \sigma_1, v_1. (\sigma_1, v_1) \in \mathcal{V}[\![\tau_1]\!] \Rightarrow \\ & (\sigma_1 \uplus \sigma_2, e[v_1/x]) \in \mathcal{C}[\![\tau_2]\!]\}\end{aligned}$$

$$\mathcal{V}[\![!\tau]\!] = \{(\{\}, !v) \mid (\{\}, v) \in \mathcal{V}[\![\tau]\!]\}$$

$$\mathcal{V}[\![\text{Ptr } \ell]\!] = \{(\{\}, \text{ptr } \ell)\}$$

$$\mathcal{V}[\![\text{Cap } \ell \tau]\!] = \{(\sigma \uplus \{\ell \mapsto v\}, \text{cap } \ell) \mid (\sigma, v) \in \mathcal{V}[\![\tau]\!]\}$$

Extended L³

- L³ only supports linear capabilities
- Supporting unrestricted ML-style references:
 - freeze – fix the type of a location, yielding an unrestricted capability to access the location *at that type*
 - thaw – temporarily transform an unrestricted capability into a linear capability, enabling *strong updates*; during this time, the location cannot be accessed along another unrestricted capability
 - refreeze – consume the thawed linear capability, restoring the location to its frozen type and enabling access along other unrestricted capabilities

Examples

$$\text{Ref } !\tau \equiv !\exists \rho. (!\text{Frzn } \rho !\tau \otimes !\text{Ptr } \rho).$$

read $\equiv \lambda r^!: \text{Ref } !\tau. \lambda t^0: \text{Thwd} \bullet.$

let $\lceil \rho, \langle f_a^!, l^! \rangle \rceil = r$ in

let $\langle c^1, t^1 \rangle = \text{thaw void}_\rho t^0 f_a$ in

let $\langle c^2, x^! \rangle = \text{swap } c^1 l \langle \rangle$ in

let $\langle c^3, \langle \rangle \rangle = \text{swap } c^2 l x$ in

let $\langle f_b^!, t^2 \rangle = \text{refreeze } t^1 c^3$ in

$\langle x, t^2 \rangle$

Examples

$\text{Ref } !\tau \equiv !\exists \rho. (!\text{Frzn } \rho !\tau \otimes !\text{Ptr } \rho).$

$\text{write} \equiv \lambda r^!: \text{Ref } !\tau. \lambda z^!: !\tau. \lambda t^0: \text{Thwd} \bullet.$

$\text{let } \lceil \rho, \langle f_a^!, l^! \rangle \rceil = r \text{ in}$

$\text{let } \langle c^1, t^1 \rangle = \text{thaw void}_\rho t^0 f_a \text{ in}$

$\text{let } \langle c^2, x^! \rangle = \text{swap } c^1 l z \text{ in}$

$\text{let } \langle f_b^!, t^2 \rangle = \text{refreeze } t^1 c^2 \text{ in}$

t^2

L³: Static Semantics

$$\boxed{\Delta; \Gamma \vdash e : \tau}$$

$$\Delta; \Gamma_1 \vdash e_3 : \text{Notin } \rho \theta \quad \Delta; \Gamma_2 \vdash e_2 : \text{Thwd } \theta$$

$$(\text{Freeze}) \quad \frac{\Delta; \Gamma_3 \vdash e_3 : \text{Cap } \rho !\tau}{\Delta; \Gamma_1 \boxplus \Gamma_2 \boxplus \Gamma_3 \vdash \text{freeze } e_1 e_2 e_3 : !(\text{Frzn } \rho !\tau) \otimes \text{Thwd } \theta}$$

$$\Delta; \Gamma_1 \vdash e_1 : \text{Notin } \rho \theta \quad \Delta; \Gamma_2 \vdash e_2 : \text{Thwd } \theta$$

$$(\text{Thaw}) \quad \frac{\Delta; \Gamma_3 \vdash e_3 : !(\text{Frzn } \rho !\tau)}{\Delta; \Gamma_1 \boxplus \Gamma_2 \boxplus \Gamma_3 \vdash \text{thaw } e_1 e_2 e_3 : \text{Cap } \rho !\tau \otimes \text{Thwd } (\theta, \rho : !\tau)}$$

$$(\text{Refreeze}) \quad \frac{\Delta; \Gamma_1 \vdash e_1 : \text{Thwd } (\theta, \rho : !\tau) \quad \Delta; \Gamma_2 \vdash e_2 : \text{Cap } \rho !\tau}{\Delta; \Gamma_1 \boxplus \Gamma_2 \vdash \text{refreeze } e_1 e_2 : !(\text{Frzn } \rho !\tau) \otimes \text{Thwd } \theta}$$
