

# 15-312 Assignment 1

Andrew Carnegie  
(andrew)

October 13, 2017

Type	$\tau ::=$		
	<b>nat</b>	<b>nat</b>	naturals
	<b>unit</b>	<b>unit</b>	unit
	<b>bool</b>	<b>bool</b>	boolean
	<b>prod</b> ( $\tau_1; \tau_2$ )	$\tau_1 \times \tau_2$	product
	<b>arr</b> ( $\tau_1; \tau_2$ )	$\tau_1 \rightarrow \tau_2$	function
	<b>list</b> ( $\tau$ )	$\tau$ <b>list</b>	list
Exp	$e ::=$		
	$x$	$x$	variable
	<b>nat</b> [ $n$ ]	$\bar{n}$	number
	<b>unit</b>	()	unit
	<b>T</b>	<b>T</b>	true
	<b>F</b>	<b>F</b>	false
	<b>if</b> ( $x; e_1; e_2$ )	<b>if</b> $x$ <b>then</b> $e_1$ <b>else</b> $e_2$	if
	<b>lam</b> ( $x : \tau.e$ )	$\lambda x : \tau.e$	abstraction
	<b>ap</b> ( $f; x$ )	$f(x)$	application
	<b>tpl</b> ( $x_1; x_2$ )	$\langle x_1, x_2 \rangle$	pair
	<b>case</b> ( $x_1, x_2.e_1$ )	<b>case</b> $p \{ (x_1; x_2) \hookrightarrow e_1 \}$	match pair
	<b>nil</b>	$\square$	nil
	<b>cons</b> ( $x_1; x_2$ )	$x_1 :: x_2$	cons
	<b>case</b> { $l$ }( $e_1; x, xs.e_2$ )	<b>case</b> $l \{ \mathbf{nil} \hookrightarrow e_1 \mid \mathbf{cons}(x; xs) \hookrightarrow e_2 \}$	match list
	<b>let</b> ( $e_1; x : \tau.e_2$ )	<b>let</b> $x = e_1$ <b>in</b> $e_2$	let
Val	$v ::=$		
	<b>val</b> ( $n$ )	$n$	numeric value
	<b>val</b> ( <b>T</b> )	<b>T</b>	true value
	<b>val</b> ( <b>F</b> )	<b>F</b>	false value
	<b>val</b> ( <b>Null</b> )	<b>Null</b>	null value
	<b>val</b> ( <b>cl</b> ( $V; x.e$ ))	( $V, x.e$ )	function value
	<b>val</b> ( $l$ )	$l$	loc value
	<b>val</b> ( <b>pair</b> ( $v_1; v_2$ ))	$\langle v_1, v_2 \rangle$	pair value
State	$s ::=$		
	<b>alive</b>	<b>alive</b>	live value
	<b>dead</b>	<b>dead</b>	dead value
Loc	$l ::=$		
	<b>loc</b> ( $l$ )	$l$	location
Var	$l ::=$		
	<b>var</b> ( $x$ )	$x$	variable

# 1 Garbage collection semantics

Model dynamics using judgement of the form:

$$\boxed{V, H, R, F \vdash e \Downarrow v, H', F'}$$

Where  $V : \text{Var} \rightarrow \text{Val} \times \text{State}$ ,  $H : \text{Loc} \rightarrow \text{Val}$ , and  $R : \{\text{Loc}\}$ . This can be read as: under stack  $V$ , heap  $H$ , roots  $R$ , freelist  $F$ , the expression  $e$  evaluates to  $v$ , and engenders a new heap  $H'$  and freelist  $F'$ .

Note that the stack maps each variable to a value  $v$  *and* a state  $s$ . If  $s$  is alive, then  $v$  can still be used, while **dead** indicates that  $v$  is already used and cannot be used again. We write  $\overline{V} = \{x \in V \mid V(x) = (\_, \text{alive})\}$  for the variables in  $V$  that are alive.

Roots represents the set of locations required to compute the continuation *excluding* the current expression. We can think of roots as the heap allocations necessary to compute the context with a hole that will be filled by the current expression.

Below defines the size of reachable values and space for roots:

$$locs_{V,H}(e) = \bigcup_{x \in FV(e)} \{l \in H \mid \exists l' \in root(x). H \models p : l' \rightsquigarrow l\}$$

$$\begin{aligned} size(\langle v_1, v_2 \rangle) &= size(v_1) + size(v_2) \\ size(\_) &= 1 \end{aligned}$$

$$\begin{aligned} copy(H, L, \langle v_1, v_2 \rangle) &= \\ &\text{let } L_1 \subseteq L \text{ with } |L_1| = size(v_1) \text{ in} \\ &\text{let } H_1, \_ = copy(H, L_1, v_1) \text{ in} \\ &copy(H_1, L \setminus L_1, v_2) \\ copy(H, l, v) &= H[l \mapsto v], l \end{aligned}$$

$$\begin{array}{c}
\frac{x \in \text{dom}(V)}{V, H, R, F \vdash x \Downarrow V(x), H, F}^{(S_1)} \quad \frac{}{V, H, R, F \vdash \bar{n} \Downarrow \text{val}(n), H, F}^{(S_2)} \\
\\
\frac{}{V, H, R, F \vdash \mathbf{T} \Downarrow \text{val}(\mathbf{T}), H, F}^{(S_3)} \quad \frac{}{V, H, R, F \vdash \mathbf{F} \Downarrow \text{val}(\mathbf{F}), H, F}^{(S_4)} \\
\\
\frac{}{V, H, R, F \vdash () \Downarrow \text{val}(\mathbf{Null}), H, F}^{(S_5)} \\
\\
\frac{V(x) = \mathbf{T} \quad g = \{l \in H \mid l \notin F \cup R \cup \text{locs}_{V,H}(e_1)\} \quad V, H, R, F \cup g \vdash e_1 \Downarrow v, H', F'}{V, H, R, F \vdash \text{if}(x; e_1; e_2) \Downarrow v, H', F'}^{(S_6)} \\
\\
\frac{V(x) = \mathbf{F} \quad g = \{l \in H \mid l \notin F \cup R \cup \text{locs}_{V,H}(e_2)\} \quad V, H, R, F \cup g \vdash e_2 \Downarrow v, H', F'}{V, H, R, F \vdash \text{if}(x; e_1; e_2) \Downarrow v, H', F'}^{(S_7)} \\
\\
\frac{l \in F \quad F' = F \setminus \{l\} \quad H' = H[l \mapsto (V, x.e)]}{V, H, R, F \vdash \text{lam}(x : \tau.e) \Downarrow l, H', F'}^{(S_8)} \\
\\
\frac{V(f) = (V_1, x.e) \quad V(x) = v_1 \quad V_1[x \mapsto v_1], H, R \vdash e \Downarrow v, H', F'}{V, H, R, F \vdash f(x) \Downarrow v, H', F'}^{(S_9)} \\
\\
\frac{V(x_1) = v_1 \quad V(x_2) = v_2}{V, H, R, F \vdash \langle x_1, x_2 \rangle \Downarrow \langle v_1, v_2 \rangle, H, F}^{(S_{10})} \\
\\
\frac{g = \{l \in H \mid l \notin F \cup R \cup \text{locs}_{V,H}(e)\} \quad V[x_1 \mapsto v_1, x_2 \mapsto v_2], H, R, F \cup g \vdash e \Downarrow v, H', F'}{V, H, R, F \vdash \text{case } x \{(x_1; x_2) \hookrightarrow e\} \Downarrow v, H', F'}^{(S_{11})} \\
\\
\frac{}{V, H, R, F \vdash \text{nil} \Downarrow \text{val}(\mathbf{Null}), H, F}^{(S_{12})} \\
\\
\frac{v = \langle V(x_1), V(x_2) \rangle \quad L \subseteq F \quad |L| = \text{size}_H(v) \quad F' = F \setminus L \quad H', l = \text{copy}(H, L, v)}{V, H, R, F \vdash \text{cons}(x_1; x_2) \Downarrow l, H', F'}^{(S_{13})} \\
\\
\frac{V(x) = \mathbf{Null} \quad g = \{l \in H \mid l \notin F \cup R \cup \text{locs}_{V',H}(e_1)\} \quad V, H, R, F \cup g \vdash e_1 \Downarrow v, H', F'}{V, H, R, F \vdash \text{case } x \{\text{nil} \hookrightarrow e_1 \mid \text{cons}(x_h; x_t) \hookrightarrow e_2\} \Downarrow v, H', F'}^{(S_{14})} \\
\\
\frac{H(l) = \langle v_h, v_t \rangle \quad V' = V\{x \mapsto (l, \text{dead})\} \quad V'' = V'[x_h \mapsto (v_h, \text{alive}), x_t \mapsto (v_t, \text{alive})] \quad g = \{l \in H \mid l \notin F \cup R \cup \text{locs}_{V'',H}(e_2)\} \quad V'', H, R, F \cup g \vdash e_2 \Downarrow v, H', F'}{V, H, R, F \vdash \text{case } x \{\text{nil} \hookrightarrow e_1 \mid \text{cons}(x_h; x_t) \hookrightarrow e_2\} \Downarrow v, H', F'}^{(S_{15})} \\
\\
\frac{R' = R \cup \text{locs}_{V,H}(\text{lam}(x : \tau.e_2)) \quad V, H, R', F \vdash e_1 \Downarrow v_1, H_1, F_1 \quad V' = V[x \mapsto v_1] \quad R'' = R \cup \text{locs}_{V',H_1}(e_2) \quad g = \{l \in H_1 \mid l \notin R'' \cup F_1\} \quad V', H_1, R, F_1 \cup g \vdash e_2 \Downarrow v_2, H_2, F_2}{V, H, R, F \vdash \text{let}(e_1; x : \tau.e_2) \Downarrow v_2, H_2, F_2}^{(S_{16})}
\end{array}$$

## 2 Type rules

The type system takes into account of garbaged collected cells by returning potential locally in a match construct. Since we are interested in the number of heap cells, all constants are assumed to be nonnegative.

$$\begin{array}{c}
\frac{n \in \mathbb{Z}}{\Sigma; \emptyset \mid \frac{q}{q} n : \mathbf{nat}} (\text{L:ConstI}) \quad \frac{}{\Sigma; \emptyset \mid \frac{q}{q} () : \mathbf{unit}} (\text{L:ConstU}) \quad \frac{}{\Sigma; \emptyset \mid \frac{q}{q} \mathbf{T} : \mathbf{bool}} (\text{L:ConstT}) \\
\\
\frac{}{\Sigma; \emptyset \mid \frac{q}{q} \mathbf{F} : \mathbf{bool}} (\text{L:ConstF}) \quad \frac{}{\Sigma; x : B \mid \frac{q}{q} x : B} (\text{L:Var}) \\
\\
\frac{\Sigma; \Gamma \mid \frac{q}{q'} e_t : B \quad \Sigma; \Gamma \mid \frac{q}{q'} e_f : B}{\Sigma; \Gamma, x : \mathbf{bool} \mid \frac{q}{q'} \mathbf{if } x \mathbf{ then } e_t \mathbf{ else } e_f : B} (\text{L:Cond}) \\
\\
\frac{}{\Sigma; x_1 : A_1, x_2 : A_2 \mid \frac{q}{q} \langle x_1, x_2 \rangle : A_1 \times A_2} (\text{L:Pair}) \\
\\
\frac{\Sigma; \Gamma, x_1 : A_1, x_2 : A_2 \mid \frac{q}{q'} e : B}{\Sigma; \Gamma, x : (A_1, A_2) \mid \frac{q}{q'} \mathbf{case } x \{ (x_1; x_2) \hookrightarrow e \} : B} (\text{L:MatP}) \quad \frac{}{\Sigma; \emptyset \mid \frac{q}{q} \mathbf{nil} : L^p(A)} (\text{L:Nil}) \\
\\
\frac{}{\Sigma; \Gamma, x_h : A, x_t : L^p(A) \mid \frac{q+p+1}{q} \mathbf{cons}(x_h; x_t) : L^p(A)} (\text{L:Cons}) \\
\\
\frac{\Sigma; \Gamma \mid \frac{q}{q'} e_1 : B \quad \Sigma; \Gamma, x_h : A, x_t : L^p(A) \mid \frac{q+p+1}{q'} e_2 : B}{\Sigma; \Gamma, x : L^p(A) \mid \frac{q}{q'} \mathbf{case } x \{ \mathbf{nil} \hookrightarrow e_1 \mid \mathbf{cons}(x_h; x_t) \hookrightarrow e_2 \} : B} (\text{L:MatL}) \\
\\
\frac{\Sigma; \Gamma_1 \mid \frac{q}{p} e_1 : A \quad \Sigma; \Gamma_2, x : A \mid \frac{p}{q'} e_2 : B}{\Sigma; \Gamma_1, \Gamma_2 \mid \frac{q}{q'} \mathbf{let}(e_1; x : \tau.e_2) : B} (\text{L:Let})
\end{array}$$

## 3 Paths and aliasing

In order prove soundness of the type system, we need some auxiliary judgements to defining properties of a heap. Below we define  $root : Val \rightarrow \{\{Loc\}\}$  that maps stack values its the root *multiset*, the multiset of locations that's already on the stack.

$$\begin{aligned}
root(\langle v_1, v_2 \rangle) &= root(v_1) \uplus root(v_2) \\
root(l) &= \{l\} \\
root(-) &= \emptyset
\end{aligned}$$

For a multiset  $S$ , we write  $\mu : S \rightarrow \mathbb{N}^+$  for the multiplicity function of  $S$ , which maps each element to the count of its occurence. If  $\forall s \in S. \mu(s) = 1$ , then  $S$  is a property set, and we denote it by  $\mathbf{set}(S)$ .

Next, we define the judgements  $\boxed{H \models p : l \rightsquigarrow l'}$  for path formation and  $\boxed{H \models p = p' : l \rightsquigarrow l'}$  for path equality. A path can be thought of as a sequence of locations that is traversable by following pointers in the heap.

$$\boxed{H \models p : l \rightsquigarrow l'}$$

$$\frac{l \in H}{H \models id_l : l \rightsquigarrow l}(\text{Id}) \quad \frac{H(l) = v \quad l' \in \text{root}(v) \quad l' \in H}{H \models (l, l') : l \rightsquigarrow l'}(\text{Edge})$$

$$\frac{H \models p : l \rightsquigarrow l' \quad H \models q : l' \rightsquigarrow l''}{H \models q \circ p : l \rightsquigarrow l''}(\text{Comp})$$

$$\boxed{H \models p = p' : l \rightsquigarrow l'}$$

$$\frac{H \models p : l \rightsquigarrow l'}{H \models p \circ id_l \equiv p : l \rightsquigarrow l'}(\text{LeftID}) \quad \frac{H \models p : l \rightsquigarrow l'}{H \models id_{l'} \circ p \equiv p : l \rightsquigarrow l'}(\text{RightID})$$

$$\frac{H(l) = v \quad l' \in \text{root}(v) \quad l' \in H \quad H \models p \equiv q : l' \rightsquigarrow l''}{H \models p \circ (l, l') \equiv q \circ (l, l') : l \rightsquigarrow l''}(\text{Eq})$$

Note that it is *not* the case that  $id_l \equiv (l, l) : l \rightsquigarrow l$ , since the former is an actual identity, while the latter is an infinite loop in the heap:  $H(l) = l$ .

Next, we define the predicates  $\text{linear}_H$  and  $\text{no\_alias}$ :

$\text{linear}_H(R_1, R_2)$ :  $\forall l \in H, \forall l_1 \in R_1, l_2 \in R_2$ , let  $H \models p : l_1 \rightsquigarrow l$  and  $H \models q : l_2 \rightsquigarrow l$ , then  $l_1 = l_2$  and  $H \models p \equiv q : l_1 \rightsquigarrow l$ .

$\text{no\_alias}(V, H)$ :  $\forall l \in H, \forall x, y \in \bar{V}, x \neq y. \text{Let } r_x = \text{root}(\bar{V}(x)), r_y = \text{root}(\bar{V}(y))$ . Then:

- (1)  $\text{set}(\text{root}(H(l))), \text{set}(r_x), \text{set}(r_y)$
- (2)  $r_x \cap r_y = \emptyset$
- (3)  $\text{linear}_H(r_x, r_x), \text{linear}_H(r_y, r_y)$ , and  $\text{linear}_H(r_x, r_y)$

## 4 Soundness for heap allocation

We simplify the soundness proof of the type system for the general metric to one with monotonic resource. (No function types for now)

**Task 1.1** (Soundness). *let  $H \models V : \Gamma$ . If  $\Sigma; \Gamma \stackrel{q}{\vdash} e : B$  and  $V, H, R, F \vdash e \Downarrow v, H', F'$ , then*

1. *If  $\text{no\_alias}(V, H)$ ,  $R \cap \text{locs}_{V, H}(e) = \emptyset$ , and  $F \cap \text{locs}_{V, H}(e) = \emptyset$ , then  $|F| - |F'| \leq \Phi_{V, H}(\Gamma) + q - (\Phi_{H'}(v : B) + q')$  and  $\text{no\_alias}(V, H')$ .*

*Proof.* Induction on the evaluation judgement.

**Case 1: E:Var**

$$\begin{aligned}
V, H, R, F &\vdash x \Downarrow V(x), H, F && \text{(admissibility)} \\
\Sigma; x : B &\Big|_{\frac{q}{q'}} x : B && \text{(admissibility)} \\
|F| - |F'| &&& (1) \\
&= |F| - |F| && \text{(ad.)} \\
&= 0 && (2) \\
\Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') &&& (3) \\
&= \Phi_{V,H}(x : B) + q - (\Phi_H(V(x) : B) + q) && \text{(ad.)} \\
&= \Phi_H(V(x) : B) + q - (\Phi_H(V(x) : B) + q) && \text{(def. of } \Phi_{V,H}) \\
&= 0 && (4) \\
|F| - |F'| &\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') && ((3),(5))
\end{aligned}$$

**Case 2: E:Const\*** Due to similarity, we show only for E:ConstI

$$\begin{aligned}
|F| - |F'| &= |F| - |F| && \text{(ad.)} \\
&= 0 \\
\Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') &= \Phi_{V,H}(\emptyset) + q - (\Phi_H(v : \text{int}) + q) && \text{(ad.)} \\
&= 0 && \text{(def of } \Phi_{V,H}) \\
|F| - |F'| &\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q')
\end{aligned}$$

**Case 4: E:App**

**Case 5: E:CondT**

$$\begin{aligned}
\Gamma &= \Gamma', x : \text{bool} && \text{(ad.)} \\
H &\models V : \Gamma' && \text{(def of W.F.E)} \\
\Sigma; \Gamma' &\Big|_{\frac{q}{q'}} e_t : B && \text{(ad.)} \\
V, H, R, F \cup g &\vdash e_t \Downarrow v, H', F' && \text{(ad.)} \\
|F \cup g| - |F'| &\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') && \text{(IH)} \\
|F| - |F'| &\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q')
\end{aligned}$$

**Case 6: E:CondF** Similar to E:CondT

**Case 7: E:Let**

$$\begin{aligned}
V, H, R', F &\vdash e_1 \Downarrow v_1, H_1, F_1 && \text{(ad.)} \\
\Sigma; \Gamma_1 &\Big|_{\frac{q}{p}} e_1 : A && \text{(ad.)} \\
H &\models V : \Gamma_1 && (\Gamma_1 \subseteq \Gamma) \\
|F| - |F_1| &\leq \Phi_{V,H}(\Gamma_1) + q - (\Phi_{H_1}(v_1 : A) + p) && \text{(IH)}
\end{aligned}$$

$$V', H_1, R, F_1 \cup g \vdash e_2 \Downarrow v_2, H_2, F_2 \quad (\text{ad.})$$

$$\Sigma; \Gamma_2, x : A \Big|_{q'}^p e_2 : B \quad (\text{ad.})$$

$$H_1 \models v_1 : A \text{ and} \quad (\text{Theorem 3.3.4})$$

$$H_1 \models V : \Gamma_2 \quad (???)$$

$$H_1 \models V' : \Gamma_2, x : A \quad (\text{def of } \models)$$

$$|F_1 \cup g| - |F_2| \leq \Phi_{V', H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q') \quad (\text{IH})$$

$$|F_1| - |F_2| \leq \Phi_{V', H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q')$$

summing the inequalities:

$$|F| - |F_1| + |F_1| - |F_2| \leq \Phi_{V, H}(\Gamma_1) + q - (\Phi_{H_1}(v_1 : A) + p) + \Phi_{V', H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q')$$

$$|F| - |F_2| \leq \Phi_{V, H}(\Gamma_1) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V', H_1}(\Gamma_2, x : A) - (\Phi_{H_2}(v_2 : B) + q')$$

$$= \Phi_{V, H}(\Gamma_1) + \Phi_{V', H_1}(\Gamma_2) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V', H_1}(x : A) - (\Phi_{H_2}(v_2 : B) + q') \quad (\text{def of } \Phi_{V, H})$$

$$nn \quad = \Phi_{V, H}(\Gamma_1) + \Phi_{V, H}(\Gamma_2) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V', H_1}(x : A) - (\Phi_{H_2}(v_2 : B) + q') \quad (\text{Lemma 4.3.3})$$

$$= \Phi_{V, H}(\Gamma) + q - \Phi_{H_1}(v_1 : A) + \Phi_{H_1}(v_1 : A) - (\Phi_{H_2}(v_2 : B) + q') \quad (\text{def of } \Phi_{V, H})$$

$$= \Phi_{V, H}(\Gamma) + q - (\Phi_{H_2}(v_2 : B) + q')$$

**Case 8: E:Pair** Similar to E:Const\*

**Case 9: E:MatP** Similar to E:MatCons

**Case 10: E:Nil** Similar to E:Const\*

**Case 11: E:Cons**

$$\begin{aligned} & |F| - |F'| \\ &= |F| - |F \setminus \{l\}| \quad (\text{ad.}) \\ &= 1 \end{aligned}$$

$$\begin{aligned} & \Phi_{V, H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') \\ &= \Phi_{V, H}(x_h : A, x_t : L^p(A)) + q + p + 1 - (\Phi_{H'}(v : L^p(A)) + q) \quad (\text{ad.}) \end{aligned}$$

$$= \Phi_{V, H}(x_h : A, x_t : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A))$$

$$= \Phi_H(V(x_h) : A) + \Phi_H(V(x_t) : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A)) \quad (\text{def of } \Phi_{V, H})$$

$$= \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A)) \quad (\text{ad.})$$

$$= \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - (p + \Phi_{H'}(v_h : A) + \Phi_{H'}(v_t : L^p(A))) \quad (\text{Lemma 4.1.1})$$

$$= \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - (p + \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A))) \quad (\text{Lemma 4.3.3})$$

$$= 1$$

Hence,

$$|F| - |F'| \leq \Phi_{V, H}(\Gamma) + q - (\Phi_{H'}(v : B) + q')$$



**Case 12: E:MatNil** Similar to E:Cond\*

**Case 13: E:MatCons**

$$\begin{aligned}
V(x) &= (l, \mathbf{alive}) && (\text{ad.}) \\
H(l) &= \langle v_h, v_t \rangle && (\text{ad.}) \\
\Gamma &= \Gamma', x : L^p(A) && (\text{ad.}) \\
\Sigma; \Gamma', x_h : A, x_t : L^p(A) &\Big|_{\frac{q+p+1}{q'}} e_2 : B && (\text{ad.}) \\
V'', H, R, F \cup g &\vdash e_2 \Downarrow v_2, H_2, F' && (\text{ad.}) \\
H &\models V(x) : L^p(A) && (\text{def of W.D.E}) \\
H'' &\models v_h : A, H'' \models v_t : L^p(A) && (\text{ad.}) \\
H &\models v_h : A, H \models v_t : L^p(A) && (???) \\
H &\models V'' : \Gamma', x_h : A, x_t : L^p(A) && (\text{def of W.D.E}) \\
\text{Suppose } \mathbf{no\_alias}(V, H), R \cap \text{locs}_{V,H}(e) = \emptyset, \text{ and } F \cap \text{locs}_{V,H}(e) = \emptyset \\
\text{NTS } |F| - |F'| \leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') \text{ and } \mathbf{no\_alias}(V, H') \\
\text{WTS } \mathbf{no\_alias}(V'', H) \\
\text{let } l \in H \text{ arbitrary, } y, z \in \overline{V}'' \text{ arbitrary, } r_y = \text{root}(\overline{V}''(y)), r_z = \text{root}(\overline{V}''(z)) \\
\quad \mathbf{case: } y \notin \{x_h, x_t\}, z \notin \{x_h, x_t\} \\
\quad y, z \in \overline{V} &&& (\text{def of } V'') \\
\quad (1) - (3) \text{ holds} &&& (\text{Sp.}) \\
\quad \mathbf{case: } y = x_h, z \notin \{x_h, x_t\} \\
\quad \text{set}(\text{root}(\langle v_h, v_t \rangle)) &&& (\text{Sp.}) \\
\quad \text{set}(\text{root}(v_h)) &&& (\text{def of set}) \\
\quad \text{set}(r_y) &&& (\text{def of } V'') \\
\quad z \in \overline{V} &&& (\text{def of } V'') \\
\quad \text{set}(r_z) &&& (\text{Sp.}) \\
\quad \text{hence we have (1)} \\
\quad \text{Suppose } l' \in r_y \cap r_z \\
\quad l' \in H &&& (H \models V'' : \Gamma', x_h : A, x_t : L^p(A)) \\
\quad H \models \text{id}_{l'} : l' \rightsquigarrow l' &&& (\text{Id}) \\
\quad H \models (l, l') : l \rightsquigarrow l' &&& (\text{Edge}) \\
\quad H \models \text{id}_{l'} \equiv (l, l') : l' \rightsquigarrow l' &&& (\text{linear}_H(r_x, r_z)) \\
\quad \text{contradiction, hence } r_y \cap r_z = \emptyset, &&& (\text{hence we have (2)}) \\
\quad \text{let } l' \in H \text{ arbitrary, } l_1, l_2 \in r_y &&& (\text{arbitrary}) \\
\quad \text{suppose } H \models p : l_1 \rightsquigarrow l', H \models q : l_2 \rightsquigarrow l' \\
\quad H \models (l, l_1) : l \rightsquigarrow l_1 \text{ and } H \models (l, l_2) : l \rightsquigarrow l_2 &&& (\text{Edge}) \\
\quad H \models p \circ (l, l_1) : l \rightsquigarrow l' \text{ and } H \models q \circ (l, l_2) : l \rightsquigarrow l' &&& (\text{Comp}) \\
\quad H \models p \circ (l, l_1) \equiv q \circ (l, l_2) : l \rightsquigarrow l' &&& (\text{linear}_H(r_x, r_x))
\end{aligned}$$

$H \models p \equiv q : l_1 \rightsquigarrow l'$  (inversion on Eq)  
 hence we have  $\text{linear}_H(r_y, r_y)$   
 $\text{linear}_H(r_z, r_z)$  (Sp.)  
 let  $l' \in H$  arbitrary,  $l_1 \in r_y, l_2 \in r_z$  (arbitrary)  
 suppose  $H \models p : l_1 \rightsquigarrow l', H \models q : l_2 \rightsquigarrow l'$   
 $H \models (l, l_1) : l \rightsquigarrow l_1$  (Edge)  
 $H \models p \circ (l, l_1) : l \rightsquigarrow l'$  (Comp)  
 $l = l_2$  ( $\text{linear}_H(r_x, r_z)$ )  
 contradiction since  $r_x \cap r_z = \emptyset$   
 hence we have  $\text{linear}_H(r_y, r_z)$   
 hence we have (3)  
**case:**  $y = x_t, z \notin \{x_h, x_t\}$   
**case:**  $y \neq \{x_h, x_t\}, z = x_h$   
**case:**  $y \neq \{x_h, x_t\}, z = x_t$   
 all symmetric to previous case  
**case:**  $y = x_h, z = x_t$   
 we get (1) the same way as the previous case  
 $\text{set}(\text{root}(\langle v_h, v_t \rangle))$  ((1))  
 $\text{set}(\text{root}(v_h) \uplus \text{root}(v_t))$  (def of  $\text{root}$ )  
 $\text{root}(v_h) \cap \text{root}(v_t) = \emptyset$  (def of  $\text{set}$ )  
 $r_y \cap r_z = \emptyset$  (def of  $r_y, r_z$ )  
 we get (3) the same way as the previous case  
 hence we have  $\text{no\_alias}(V'', H)$   
 let  $l' \in \text{locs}_{V'', H}(e_2)$  arbitrary  
 $\exists! x' \in \overline{V}'' . \exists! l'' \in \text{root}(\overline{V}''(x')). H \models p : l'' \rightsquigarrow l'$  (def of  $\text{locs}_{V, H}$ )  
**case:**  $x' \notin \{x_h, x_t\}$   
 $x \in \overline{V}$  (def of  $V''$ )  
 $l' \in \text{locs}_{V, H}(e)$  (def of  $\text{locs}_{V, H}$ )  
**case:**  $x' = x_h$   
 $H \models (l, l'') : l \rightsquigarrow l''$  (Edge)  
 $H \models p \circ (l, l'') : l \rightsquigarrow l'$  (Comp)  
 $l' \in \text{locs}_{V, H}(e)$  (def of  $\text{locs}_{V, H}$ )  
 thus we have  $\text{locs}_{V''H}(e_2) \subseteq \text{locs}_{V, H}(e)$   
 $F \cap \text{locs}_{V'', H}(e_2) = \emptyset$  (Sp.)  
 $g \cap \text{locs}_{V'', H}(e_2) = \emptyset$  (def. of  $g$ )  
 $(F \cup g) \cap \text{locs}_{V'', H}(e_2) = \emptyset$   
 $|F \cup g| - |F'| \leq \Phi_{V, H}(\Gamma', x_h : A, x_t : L^p(A)) + q + p + 1 - (\Phi_{H'}(v : B) + q')$  (IH)

$$\begin{aligned}
&= \Phi_{V,H}(\Gamma') + \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + q + 1 - (\Phi_{H'}(v : B) + q') \\
&\quad \text{(def of } \Phi_{V,H}) \\
&= \Phi_{V,H}(\Gamma') + \Phi_H(\langle v_h, v_t \rangle^L : L^p(A)) + q + 1 - (\Phi_{H'}(v : B) + q') \quad \text{(Lemma 4.1.1)} \\
&= \Phi_{V,H}(\Gamma', z : L^p(A)) + q + 1 - (\Phi_{H'}(v : B) + q') \quad \text{(def of } \Phi_{V,H}) \\
&= \Phi_{V,H}(\Gamma) + q + 1 - (\Phi_{H'}(v : B) + q') \quad \text{(Lemma 4.1.1)}
\end{aligned}$$

suppose  $l \in \text{locs}_{V',H}(e_2)$

$$\exists x' \in FV(e_2) \cap \bar{V}'', l' \in \text{root}(\bar{V}''(x')). x \neq x', H \models p : l' \rightsquigarrow l \quad \text{(def. of } \text{locs}_{V,H})$$

**case:**  $x' \notin \{x_h, x_t\}$

contradiction `byno_alias`( $V, H$ )

**case:**  $x' = x_h$

$$H \models p \circ (l, l') : l \rightsquigarrow l$$

$$H \models \text{id}_l : l \rightsquigarrow l$$

contradiction since `linear` <sub>$H$</sub> ( $r_x, r_x$ )

hence we have  $l \notin \text{locs}_{V'',H}(e_2)$

$$l \in g \quad \text{(def of } g)$$

$$|g| \geq 1$$

$$|F \cup g| - |F'|$$

$$= |F| + |g| - |F'| \quad (F, g \text{ disjoint})$$

Hence,

$$|F| + |g| - |F'| \leq \Phi_{V,H}(\Gamma) + q + 1 - (\Phi_{H'}(v : B) + q')$$

$$|F| - |F'| \leq \Phi_{V,H}(\Gamma) + q + 1 - |g| - (\Phi_{H'}(v : B) + q')$$

$$\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') \quad (|g| \geq 1)$$

□