15-312 Assignment 1

Andrew Carnegie
(andrew)

October 12, 2017

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

$$
\begin{aligned}
&reach_H((V, x.e)) = \bigcup_{y \in FV(e) \setminus x} reach_H(V(y)) \\
&reach_H(l) = \{l\} \cup reach_H(H(l)) \\
&reach_H(\langle v_1, v_2 \rangle) = reach_H(v_1) \cup reach_H(v_2) \\
&reach_H(\_) = \emptyset
\end{aligned}
$$

$$locs_{V,H}(e) = \bigcup_{x \in FV(e)} reach_H(V(x))$$

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

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

$$\frac{x \in dom(V)}{V, H, R, F \vdash x \Downarrow V(x), H, F}\,(\mathrm{S}_1) \qquad \frac{}{V, H, R, F \vdash \overline{n} \Downarrow \texttt{val}(n), H, F}\,(\mathrm{S}_2)$$

$$\frac{}{V, H, R, F \vdash \texttt{T} \Downarrow \texttt{val}(\texttt{T}), H, F}\,(\mathrm{S}_3) \qquad \frac{}{V, H, R, F \vdash \texttt{F} \Downarrow \texttt{val}(\texttt{F}), H, F}\,(\mathrm{S}_4)$$

$$\frac{}{V, H, R, F \vdash () \Downarrow \texttt{val}(\texttt{Null}), H, F}\,(\mathrm{S}_5)$$

$$\frac{V(x) = \texttt{T} \qquad g = \{l \in H | l \notin F \cup R \cup locs_{V,H}(e_1)\} \qquad V, H, R, F \cup g \vdash e_1 \Downarrow v, H', F'}{V, H, R, F \vdash \texttt{if}(x; e_1; e_2) \Downarrow v, H', F'}\,(\mathrm{S}_6)$$

$$\frac{V(x) = \texttt{F} \qquad g = \{l \in H | l \notin F \cup R \cup locs_{V,H}(e_2)\} \qquad V, H, R, F \cup g \vdash e_2 \Downarrow v, H', F'}{V, H, R, F \vdash \texttt{if}(x; e_1; e_2) \Downarrow v, H', F'}\,(\mathrm{S}_7)$$

$$\frac{l \in F \qquad F' = F \setminus \{l\} \qquad H' = H[l \mapsto (V, x.e)]}{V, H, R, F \vdash \texttt{lam}(x : \tau.e) \Downarrow l, H', F'}\,(\mathrm{S}_8)$$

$$\frac{V(f) = (V_1, x.e) \qquad V(x) = v_1 \qquad V_1[x \mapsto v_1], H, R \vdash e \Downarrow v, H', F'}{V, H, R, F \vdash f(x) \Downarrow v, H', F'}\,(\mathrm{S}_9)$$

$$\frac{V(x_1) = v_1 \qquad V(x_2) = v_2}{V, H, R, F \vdash \langle x_1, x_2 \rangle \Downarrow \langle v_1, v_2 \rangle, H, F}\,(\mathrm{S}_{10})$$

$$\frac{\begin{array}{c} V(x) = \langle v_1, v_2 \rangle \\ g = \{l \in H | l \notin F \cup R \cup locs_{V,H}(e)\} \qquad V[x_1 \mapsto v_1, x_2 \mapsto v_2], H, R, F \cup g \vdash e \Downarrow v, H', F' \end{array}}{V, H, R, F \vdash \texttt{case}\ x\ \{(x_1; x_2) \hookrightarrow e\} \Downarrow v, H', F'}\,(\mathrm{S}_{11})$$

$$\frac{}{V, H, R, F \vdash \texttt{nil} \Downarrow \texttt{val}(\texttt{Null}), H, F}\,(\mathrm{S}_{12})$$

$$\frac{v = \langle V(x_1), V(x_2) \rangle \qquad L \subseteq F \qquad |L| = size_H(v) \qquad F' = F \setminus L \qquad H', l = copy(H, L, v)}{V, H, R, F \vdash \texttt{cons}(x_1; x_2) \Downarrow l, H', F'}\,(\mathrm{S}_{13})$$

$$\frac{V(x) = \texttt{Null} \qquad g = \{l \in H | l \notin F \cup R \cup locs_{V',H}(e_1)\} \qquad V, H, R, F \cup g \vdash e_1 \Downarrow v, H', F'}{V, H, R, F \vdash \texttt{case}\, x\, \{\texttt{nil} \hookrightarrow e_1 \mid \texttt{cons}(x_h; x_t) \hookrightarrow e_2\} \Downarrow v, H', F'}\,(\mathrm{S}_{14})$$

$$\frac{\begin{array}{c} V(x) = (l, \texttt{alive}) \\ H(l) = \langle v_h, v_t \rangle \qquad g = \{l \in H | l \notin F \cup R \cup locs_{V',H}(e_2)\} \qquad V' = V\{x \mapsto (l, \texttt{dead})\} \\ V'' = V'[x_h \mapsto (v_h, \texttt{alive}), x_t \mapsto (v_t, \texttt{alive})] \qquad V'', H, R, F \cup g \vdash e_2 \Downarrow v, H', F' \end{array}}{V, H, R, F \vdash \texttt{case}\, x\, \{\texttt{nil} \hookrightarrow e_1 \mid \texttt{cons}(x_h; x_t) \hookrightarrow e_2\} \Downarrow v, H', F'}\,(\mathrm{S}_{15})$$

$$\frac{\begin{array}{c} R' = R \cup locs_{V,H}(\texttt{lam}(x : \tau.e_2)) \qquad V, H, R', F \vdash e_1 \Downarrow v_1, H_1, F_1 \qquad V' = V[x \mapsto v_1] \\ R'' = R \cup locs_{V',H_1}(e_2) \qquad g = \{l \in H_1 | l \notin R'' \cup F_1\} \qquad V', H_1, R, F_1 \cup g \vdash e_2 \Downarrow v_2, H_2, F_2 \end{array}}{V, H, R, F \vdash \texttt{let}(e_1; x : \tau.e_2) \Downarrow v_2, H_2, F_2}\,(\mathrm{S}_{16})$$

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.

$$\frac{n \in \mathbb{Z}}{\Sigma; \emptyset \left|\frac{q}{q}\right. n : \mathtt{nat}}\text{(L:ConstI)} \qquad \frac{}{\Sigma; \emptyset \left|\frac{q}{q}\right. () : \mathtt{unit}}\text{(L:ConstU)} \qquad \frac{}{\Sigma; \emptyset \left|\frac{q}{q}\right. \mathtt{T} : \mathtt{bool}}\text{(L:ConstT)}$$

$$\frac{}{\Sigma; \emptyset \left|\frac{q}{q}\right. \mathtt{F} : \mathtt{bool}}\text{(L:ConstF)} \qquad \frac{}{\Sigma; x : B \left|\frac{q}{q}\right. x : B}\text{(L:Var)}$$

$$\frac{\Sigma; \Gamma \left|\frac{q}{q'}\right. e_t : B \qquad \Sigma; \Gamma \left|\frac{q}{q'}\right. e_f : B}{\Sigma; \Gamma, x : \mathtt{bool} \left|\frac{q}{q'}\right. \mathtt{if}\, x\, \mathtt{then}\, e_t\, \mathtt{else}\, e_f : B}\text{(L:Cond)}$$

$$\frac{}{\Sigma; x_1 : A_1, x_2 : A_2 \left|\frac{q}{q}\right. \langle x_1, x_2 \rangle : A_1 \times A_2}\text{(L:Pair)}$$

$$\frac{\Sigma; \Gamma, x_1 : A_1, x_2 : A_2 \left|\frac{q}{q'}\right. e : B}{\Sigma; \Gamma, x : (A_1, A_2) \left|\frac{q}{q'}\right. \mathtt{case}\ x\ \{(x_1; x_2) \hookrightarrow e\} : B}\text{(L:MatP)} \qquad \frac{}{\Sigma; \emptyset \left|\frac{q}{q}\right. \mathtt{nil} : L^p(A)}\text{(L:Nil)}$$

$$\frac{}{\Sigma; \Gamma, x_h : A, x_t : L^p(A) \left|\frac{q+p+1}{q}\right. \mathtt{cons}(x_h; x_t) : L^p(A)}\text{(L:Cons)}$$

$$\frac{\Sigma; \Gamma \left|\frac{q}{q'}\right. e_1 : B \qquad \Sigma; \Gamma, x_h : A, x_t : L^p(A) \left|\frac{q+p+1}{q'}\right. e_2 : B}{\Sigma; \Gamma, x : L^p(A) \left|\frac{q}{q'}\right. \mathtt{case}\, x\, \{\mathtt{nil} \hookrightarrow e_1 \mid \mathtt{cons}(x_h; x_t) \hookrightarrow e_2\} : B}\text{(L:MatL)}$$

$$\frac{\Sigma; \Gamma_1 \left|\frac{q}{p}\right. e_1 : A \qquad \Sigma; \Gamma_2, x : A \left|\frac{p}{q'}\right. e_2 : B}{\Sigma; \Gamma_1, \Gamma_2 \left|\frac{q}{q'}\right. \mathtt{let}(e_1; x : \tau.e_2) : B}\text{(L:Let)}$$

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 $root : Val \to \{\{Loc\}\}$ 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 $\mu : S \to \mathbb{N}^+$ 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 $\mathsf{set}(S)$.

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

$\boxed{\mathrm{H} \vDash p : l \rightsquigarrow l'}$

$$\frac{l \in H}{H \vDash id_l : l \rightsquigarrow l}(\mathrm{Id}) \qquad \frac{H(l) = v \qquad l' \in root(v)}{H \vDash (l, l') : l \rightsquigarrow l'}(\mathrm{Edge})$$

$$\frac{H \vDash p : l \rightsquigarrow l' \qquad H \vDash q : l' \rightsquigarrow l''}{H \vDash q \circ p : l \rightsquigarrow l''}(\mathrm{Comp})$$

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

$$\frac{H \vDash p : l \rightsquigarrow l'}{H \vDash p \circ id_l \equiv p : l \rightsquigarrow l'}(\mathrm{LeftID}) \qquad \frac{H \vDash p : l \rightsquigarrow l'}{H \vDash id_{l'} \circ p \equiv p : l \rightsquigarrow l'}(\mathrm{RightID})$$

$$\frac{H(l) = v \qquad l' \in root(v) \qquad H \vDash p \equiv q : l' \rightsquigarrow l''}{H \vDash p \circ (l, l') \equiv q \circ (l, l') : l \rightsquigarrow l''}(\mathrm{Eq})$$

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 linear_H and $\mathsf{no\_alias}$:

$\mathsf{linear}_H(R_1, R_2)$**:** $\forall l_1 \in R_1, l_2 \in R_2$, if $H \vDash p : l_1 \rightsquigarrow l$ and $H \vDash q : l_2 \rightsquigarrow l$, then $H \vDash p \equiv q : l_1 \rightsquigarrow l$.

$\mathsf{no\_alias}(V, H)$**:** $\forall l \in H$, $\forall x, y \in \overline{V}$, let $r_x = root(\overline{V}(x))$, $r_y = root(\overline{V}(y))$. Then:

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

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* $H \vDash V : \Gamma$. *If* $\Sigma; \Gamma \left|\frac{q}{q'}\right. e : B$ *and* $V, H, R, F \vdash e \Downarrow v, H', F'$, *then*

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

Proof. Induction on the evaluation judgement.

Case 1: E:Var

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

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

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

Case 4: E:App

Case 5: E:CondT

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

Case 6: E:CondF Similar to E:CondT

Case 7: E:Let

$$V, H, R', F \vdash e_1 \Downarrow v_1, H_1, F_1 \qquad \text{(ad.)}$$
$$\Sigma; \Gamma_1 \left|\frac{q}{p}\right. e_1 : A \qquad \text{(ad.)}$$
$$H \vDash V : \Gamma_1 \qquad (\Gamma_1 \subseteq \Gamma)$$
$$|F| - |F_1| \leq \Phi_{V,H}(\Gamma_1) + q - (\Phi_{H_1}(v_1 : A) + p) \qquad \text{(IH)}$$
$$V', H_1, R, F_1 \cup g \vdash e_2 \Downarrow v_2, H_2, F_2 \qquad \text{(ad.)}$$
$$\Sigma; \Gamma_2, x : A \left|\frac{p}{q'}\right. e_2 : B \qquad \text{(ad.)}$$
$$H_1 \vDash v_1 : A \text{ and} \qquad \text{(Theorem 3.3.4)}$$
$$H_1 \vDash V : \Gamma_2 \qquad \text{(???)}$$
$$H_1 \vDash V' : \Gamma_2, x : A \qquad \text{(def of } \vDash)$$
$$|F_1 \cup g| - |F_2| \leq \Phi_{V',H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q') \qquad \text{(IH)}$$
$$|F_1| - |F_2| \leq \Phi_{V',H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q')$$

summing the inequalities:

$$|F| - |F_1| + |F_1| - |F_2| \leq \Phi_{V,H}(\Gamma_1) + q - (\Phi_{H_1}(v_1 : A) + p) + \Phi_{V',H_1}(\Gamma_2, x : A) + p - (\Phi_{H_2}(v_2 : B) + q')$$
$$|F| - |F_2| \leq \Phi_{V,H}(\Gamma_1) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V',H_1}(\Gamma_2, x : A) - (\Phi_{H_2}(v_2 : B) + q')$$
$$= \Phi_{V,H}(\Gamma_1) + \Phi_{V',H_1}(\Gamma_2) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V',H_1}(x : A) - (\Phi_{H_2}(v_2 : B) + q') \qquad \text{(def of } \Phi_{V,H})$$
$$nn \quad = \Phi_{V,H}(\Gamma_1) + \Phi_{V,H}(\Gamma_2) + q - \Phi_{H_1}(v_1 : A) + \Phi_{V',H_1}(x : A) - (\Phi_{H_2}(v_2 : B) + q') \qquad \text{(Lemma 4.3.3)}$$
$$= \Phi_{V,H}(\Gamma) + q - \Phi_{H_1}(v_1 : A) + \Phi_{H_1}(v_1 : A) - (\Phi_{H_2}(v_2 : B) + q') \qquad \text{(def of } \Phi_{V,H})$$
$$= \Phi_{V,H}(\Gamma) + q - (\Phi_{H_2}(v_2 : B) + q')$$

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

$$
\begin{aligned}
&|F| - |F'| \\
&\quad = |F| - |F \setminus \{l\}| && \text{(ad.)} \\
&\quad = 1 \\
&\Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q') \\
&\quad = \Phi_{V,H}(x_h : A, x_t : L^p(A)) + q + p + 1 - (\Phi_{H'}(v : L^p(A)) + q) && \text{(ad.)} \\
&\quad = \Phi_{V,H}(x_h : A, x_t : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A))) \\
&\quad = \Phi_H(V(x_h) : A) + \Phi_H(V(x_t) : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A))) && \text{(def of } \Phi_{V,H}\text{)} \\
&\quad = \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - \Phi_{H'}(v : L^p(A))) && \text{(ad.)} \\
&\quad = \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - (p + \Phi_{H'}(v_h : A) + \Phi_{H'}(v_t : L^p(A))) \\
& && \text{(Lemma 4.1.1)} \\
&\quad = \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + 1 - (p + \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A))) \\
& && \text{(Lemma 4.3.3)} \\
&\quad = 1
\end{aligned}
$$

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

—F $\cup g| - |F'| \leq \Phi_{V,H}(\Gamma', x_h : A, x_t : L^p(A)) + q + p + 1 - (\Phi_{H'}(v : B) + q')$
$= \Phi_{V,H}(\Gamma') + \Phi_H(v_h : A) + \Phi_H(v_t : L^p(A)) + p + q + 1 - (\Phi_{H'}(v : B) + q')$
$= \Phi_{V,H}(\Gamma') + \Phi_H(\langle v_h, v_t \rangle^L : L^p(A)) + q + 1 - (\Phi_{H'}(v : B) + q')$
$= \Phi_{V,H}(\Gamma', z : L^p(A)) + q + 1 - (\Phi_{H'}(v : B) + q')$
$= \Phi_{V,H}(\Gamma) + q + 1 - (\Phi_{H'}(v : B) + q')$
$(l, n+1) \in F \cup g$ and $\nexists!(l', k) \in F \cup g. l' = l \wedge k \neq n+1$
$(l, n+1) \notin F$
$(l, n+1) \in g$
$|g| \geq 1$
$|F \cup g| - |F'|$
$= |F| + |g| - |F'|$
Hence,
$|F| + |g| - |F'| \leq \Phi_{V,H}(\Gamma) + q + 1 - (\Phi_{H'}(v : B) + q')$
$|F| - |F'| \leq \Phi_{V,H}(\Gamma) + q + 1 - |g| - (\Phi_{H'}(v : B) + q')$
$\leq \Phi_{V,H}(\Gamma) + q - (\Phi_{H'}(v : B) + q')$

□