15-312 Assignment 1

Andrew Carnegie
(andrew)

September 4, 2017

1 Syntax

Type	τ ::=		
	`nat`	`nat`	naturals
	`unit`	`unit`	unit
	`bool`	`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
	`unit`	$()$	unit
	`T`	`T`	true
	`F`	`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
	$\texttt{tpl}(x_1;x_2)$	$\langle x_1, x_2\rangle$	tuple
	$\texttt{fst}(x)$	$x \cdot \texttt{l}$	first projection
	$\texttt{snd}(x)$	$x \cdot \texttt{r}$	second projection
	`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 ::=		
	$\texttt{val}[l](n)$	n^l	numeric value
	$\texttt{val}[l](\texttt{T})$	$\texttt{T}^l$	true value
	$\texttt{val}[l](\texttt{F})$	$\texttt{F}^l$	false value
	$\texttt{val}[l](\texttt{Null})$	$\texttt{Null}^l$	null value
	$\texttt{val}[l](\texttt{cl}(V;x.e))$	$(V, x.e)^l$	function value
	$\texttt{val}[l_2](l_1)$	$l_1^{l_2}$	loc value
	$\texttt{val}[l](\texttt{pair}(v_1;v_2))$	$\langle v_1, v_2\rangle^l$	pair value
Loc	l ::=		
	$\texttt{loc}(l)$	l	location

2 Garbage collection semantics

Model dynamics using judgement of the form:

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

Where $V : VID \to Val$, $H : Loc \to Val$, and $R : \{Loc\}$. This can be read as: under stack

V, heap H, roots R, freelist s, the expression e evaluates to v, and engenders a new heap H' and freelist s'.

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:

$$
\begin{aligned}
&reach_H(n^l) = \{l\} \\
&reach_H(\mathtt{T}^l) = \{l\} \\
&reach_H(\mathtt{F}^l) = \{l\} \\
&reach_H(\mathtt{Null}^l) = \{l\} \\
&reach_H((V, x.e)^l) = \{l\} \cup (\bigcup_{y \in FV(e) \setminus x} reach_H(V(y))) \\
&reach_H(l_1^{l_2}) = \{l_2\} \cup loc_H(H(l_1)) \\
&reach_H(\langle v_1, v_2 \rangle^l) = \{l\} \cup reach_H(v_1) \cup reach_H(v_2)
\end{aligned}
$$

$$loc_H(l) = \{l\} \cup reach_H(H(l))$$

$$space_H(R) = |\bigcup_{l \in R} loc_H(l)|$$

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

$$\frac{x \in dom(V)}{V, H, R \vdash x \Downarrow^{space_H(R \cup (reach_H(V(x))))} V(x), H}(\mathrm{S}_1) \qquad \frac{(l \text{ fresh}) \qquad H' = H[l \mapsto n^l]}{V, H, R \vdash \overline{n} \Downarrow^{space_{H'}(R \cup (\{l\})} n^l, H'}(\mathrm{S}_2)$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{T}^l]}{V, H, R \vdash \mathtt{T} \Downarrow^{space_{H'}(R \cup \{l\})} \mathtt{T}^l, H'}(\mathrm{S}_3) \qquad \frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{F}^l]}{V, H, R \vdash \mathtt{F} \Downarrow^{space_{H'}(R \cup \{l\})} \mathtt{F}^l, H'}(\mathrm{S}_4)$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{Null}^l]}{V, H, R \vdash () \Downarrow^{space_{H'}(R \cup \{l\})} \mathtt{Null}^l, H'}(\mathrm{S}_5) \qquad \frac{V(x) = \mathtt{T}^l \qquad V, H, R \vdash e_1 \Downarrow^{s_1} v_1, H_1}{V, H, R \vdash \mathtt{if}(x; e_1; e_2) \Downarrow^{s_1} v_1, H_1}(\mathrm{S}_6)$$

$$\frac{V(x) = \mathtt{F}^l \qquad V, H, R \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H, R \vdash \mathtt{if}(x; e_1; e_2) \Downarrow^{s_2} v_2, H_2}(\mathrm{S}_7)$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto (V, x.e)^l]}{V, H, R \vdash \mathtt{lam}(x : \tau.e) \Downarrow^{space_{H'}(R \cup \{l\})} (V, x.e)^l, H'}(\mathrm{S}_8)$$

$$\frac{V(f) = (V_1, x.e)^{l_1} \qquad V(x) = v_1 \qquad V_1[x \mapsto v_1], H, R \vdash e \Downarrow^s v, H'}{V, H, R \vdash f(x) \Downarrow^s v, H'}(\mathrm{S}_9)$$

$$\frac{V(x_1) = v_1 \qquad V(x_2) = v_2 \qquad (l \text{ fresh}) \qquad H' = H[l \mapsto \langle v_1, v_2 \rangle^l]}{V, H, R \vdash \langle x_1, x_2 \rangle \Downarrow^{space_{H'}(R \cup \{l\})} \langle v_1, v_2 \rangle^l, H'}(\mathrm{S}_{10})$$

$$\frac{V(x) = \langle v_1, v_2 \rangle^l}{V, H, R \vdash x \cdot \mathtt{l} \Downarrow^{space_{H'}(R \cup reach(v_1))} v_1, H}(\mathrm{S}_{11}) \qquad \frac{V(x) = \langle v_1, v_2 \rangle^l}{V, H, R \vdash x \cdot \mathtt{r} \Downarrow^{space_{H'}(R \cup reach(v_2))} v_2, H'}(\mathrm{S}_{12})$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{Null}^l]}{V, H, R \vdash \mathtt{nil} \Downarrow^{space_{H'}(R \cup \{l\})} \mathtt{Null}^l, H'}(\mathrm{S}_{13})$$

$$\frac{V(x_1) = v_1 \qquad V(x_2) = v_2 \qquad (l \text{ fresh}) \qquad H' = H[l \mapsto \langle v_1, v_2 \rangle^l]}{V, H, R \vdash \mathtt{cons}(x_1; x_2) \Downarrow^{space_{H'}(R \cup \{l\})} \langle v_1, v_2 \rangle^l, H'}(\mathrm{S}_{14})$$

$$\frac{V(z) = \mathtt{Null}^l \qquad V, H, R \vdash e_1 \Downarrow^{s_1} v_1, H_1}{V, H, R \vdash \mathtt{case}\, z \,\{\mathtt{nil} \hookrightarrow e_1 \mid \mathtt{cons}(x; xs) \hookrightarrow e_2\} \Downarrow^{s_1} v_1, H_1}(\mathrm{S}_{15})$$

$$\frac{V(z) = \langle v_h, v_t \rangle^l \qquad V[x \mapsto v_h, xs \mapsto v_t], H, R \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H, R \vdash \mathtt{case}\, z \,\{\mathtt{nil} \hookrightarrow e_1 \mid \mathtt{cons}(x; xs) \hookrightarrow e_2\} \Downarrow^{s_2} v_2, H_2}(\mathrm{S}_{16})$$

$$\frac{V, H, R \cup locs_{V,H}(\mathtt{lam}(x : \tau.e_2)) \vdash e_1 \Downarrow^{s_1} v_1, H' \qquad V[x \mapsto v_1], H', R \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H, R \vdash \mathtt{let}(e_1; x : \tau.e_2) \Downarrow^{\max(s_1, s_2)} v_2, H_2}(\mathrm{S}_{17})$$

3 Heap allocation semantics

The following rules will use judgements of the form:

$$\boxed{V, H \vdash e \Downarrow^{s} v, H'}$$

Where $V : VID \to Val$ and $H : Loc \to Val$. This can be read as: under stack V and heap H the expression e evaluates to v while allocating s heap cells, and engenders a new heap H'.

This is different from the above GC semantics, since s counts the number of heap allocations, not the maximum heap size. Hence this heap allocation semantics is an upperbound on the GC semantics.

We define the following metric for measuring the number of heap allocations:

$$\begin{aligned}
K^{int} &= 1 \\
K^{true} &= 1 \\
K^{false} &= 1 \\
K^{null} &= 1 \\
K^{tuple} &= 1 \\
K^{nil} &= 1 \\
K^{cons} &= 1 \\
K^{\text{-}} &= 0
\end{aligned}$$

Where $K^{\text{-}}$ are all other constants.

$$\frac{x \in dom(V)}{V, H \vdash x \Downarrow^{K^{var}} V(x), H}(\mathrm{S}_{18}) \qquad \frac{(l \text{ fresh}) \qquad H' = H[l \mapsto n^l]}{V, H \vdash \overline{n} \Downarrow^{K^{int}} n^l, H'}(\mathrm{S}_{19})$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{T}^l]}{V, H \vdash \mathtt{T} \Downarrow^{K^{true}} \mathtt{T}^l, H'}(\mathrm{S}_{20}) \qquad \frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{F}^l]}{V, H \vdash \mathtt{F} \Downarrow^{K^{false}} \mathtt{F}^l, H'}(\mathrm{S}_{21})$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{Null}^l]}{V, H \vdash () \Downarrow^{K^{null}} \mathtt{Null}^l, H'}(\mathrm{S}_{22}) \qquad \frac{V(x) = \mathtt{T}^l \qquad V, H \vdash e_1 \Downarrow^{s_1} v_1, H_1}{V, H \vdash \mathtt{if}(x; e_1; e_2) \Downarrow^{s_1} v_1, H_1}(\mathrm{S}_{23})$$

$$\frac{V(x) = \mathtt{F}^l \qquad V, H \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H \vdash \mathtt{if}(x; e_1; e_2) \Downarrow^{s_2} v_2, H_2}(\mathrm{S}_{24}) \qquad \frac{(l \text{ fresh}) \qquad H' = H[l \mapsto (V, x.e)^l]}{V, H \vdash \mathtt{lam}(x : \tau.e) \Downarrow^{K^{lam}} (V, x.e)^l, H'}(\mathrm{S}_{25})$$

$$\frac{V(f) = (V_1, x.e)^{l_1} \qquad V(x) = v_1 \qquad V[x \mapsto v_1], H \vdash e \Downarrow^{s} v, H'}{V, H \vdash f(x) \Downarrow^{s} v, H'}(\mathrm{S}_{26})$$

$$\frac{V(x_1) = v_1 \qquad V(x_2) = v_2 \qquad (l \text{ fresh}) \qquad H' = H[l \mapsto \langle v_1, v_2 \rangle^l]}{V, H \vdash \langle x_1, x_2 \rangle \Downarrow^{K^{tuple}} \langle v_1, v_2 \rangle^l, H'}(\mathrm{S}_{27})$$

$$\frac{V(x) = \langle v_1, v_2 \rangle^l}{V, H \vdash x \cdot \mathtt{l} \Downarrow^{K^{fst}} v_1, H}(\mathrm{S}_{28}) \qquad \frac{V(x) = \langle v_1, v_2 \rangle^l}{V, H \vdash x \cdot \mathtt{r} \Downarrow^{K^{snd}} v_2, H'}(\mathrm{S}_{29})$$

$$\frac{(l \text{ fresh}) \qquad H' = H[l \mapsto \mathtt{Null}^l]}{V, H \vdash \mathtt{nil} \Downarrow^{K^{nil}} \mathtt{Null}^l, H'}(\mathrm{S}_{30})$$

$$\frac{V(x_1) = v_1 \qquad V(x_2) = v_2 \qquad (l \text{ fresh}) \qquad H' = H[l \mapsto \langle v_1, v_2 \rangle^l]}{V, H \vdash \mathtt{cons}(x_1; x_2) \Downarrow^{K^{cons}} \langle v_1, v_2 \rangle^l, H'}(\mathrm{S}_{31})$$

$$\frac{V(z) = \mathtt{Null}^l \qquad V, H \vdash e_1 \Downarrow^{s_1} v_1, H_1}{V, H \vdash \mathtt{case}\, z\, \{\mathtt{nil} \hookrightarrow e_1 \mid \mathtt{cons}(x; xs) \hookrightarrow e_2\} \Downarrow^{s_1} v_1, H_1}(\mathrm{S}_{32})$$

$$\frac{V(z) = \langle v_h, v_t \rangle^l \qquad V[x \mapsto v_h, xs \mapsto v_t], H \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H \vdash \mathtt{case}\, z\, \{\mathtt{nil} \hookrightarrow e_1 \mid \mathtt{cons}(x; xs) \hookrightarrow e_2\} \Downarrow^{s_2} v_2, H_2}(\mathrm{S}_{33})$$

$$\frac{V, H \vdash e_1 \Downarrow^{s_1} v_1, H' \qquad V[x \mapsto v_1], H' \vdash e_2 \Downarrow^{s_2} v_2, H_2}{V, H \vdash \mathtt{let}(e_1; x : \tau.e_2) \Downarrow^{s_1 + s_2} v_2, H_2}(\mathrm{S}_{34})$$

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* $H \vDash V : \Gamma$ *and* $\Sigma; \Gamma \,\Big|\!\frac{q}{q'}\, e : B$

1. *If* $V, H \vdash e \rightsquigarrow v$