Lab 5 Written Solutions

CAS CS 320: *Concepts of Programming Languages*

Typing rules for options:

$$\frac{}{\Gamma \vdash \texttt{None} : \tau\ \texttt{option}}\ \text{(none)} \qquad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \texttt{Some}\ e : \tau\ \texttt{option}}\ \text{(some)} \qquad \frac{\Gamma \vdash e_1 : \tau_1\ \texttt{option} \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma, x : \tau_1 \vdash e_3 : \tau}{\Gamma \vdash \texttt{match}\ e_1\ \texttt{with | None ->}\ e_2\ \texttt{| Some}\ x\ \texttt{->}\ e_3 : \tau}\ \text{(opt-match)}$$

Semantic rules for options:

$$\frac{}{\texttt{None} \Downarrow \text{None}}\ \text{(none-eval)} \qquad \frac{e \Downarrow v}{\texttt{Some}\ e \Downarrow \text{Some}(v)}\ \text{(some-eval)}$$

$$\frac{e_1 \Downarrow \text{None} \qquad e_2 \Downarrow v}{\texttt{match}\ e\ \texttt{with}_1\ \texttt{| None ->}\ e_2\ \texttt{| Some}\ x\ \texttt{->}\ e_3 \Downarrow v}\ \text{(opt-eval-none)} \qquad \frac{e_1 \Downarrow \text{Some}(v_1) \qquad e = [v_1/x]e_3 \qquad e \Downarrow v}{\texttt{match}\ e_1\ \texttt{with | None ->}\ e_2\ \texttt{| Some}\ x\ \texttt{->}\ e_3 \Downarrow v}\ \text{(opt-eval-some)}$$

Typing Derivation:

$$\dfrac{\dfrac{\dfrac{}{\{\texttt{x} : \texttt{int option}\} \vdash \texttt{x} : \texttt{int option}}\ \text{(var)} \quad \dfrac{}{\{\texttt{x} : \texttt{int option}\} \vdash \texttt{0} : \texttt{int}}\ \text{(int)} \quad \dfrac{}{\{\texttt{x} : \texttt{int option}, \texttt{y} : \texttt{int}\} \vdash \texttt{y} : \texttt{int}}\ \text{(var)}}{\{\texttt{x} : \texttt{int option}\} \vdash \texttt{match x with | None -> 0 | Some y -> y} : \texttt{int}}\ \text{(opt-match)}}{\varnothing \vdash \texttt{fun x -> match x with | None -> 0 | Some y -> y} : \texttt{int option -> int}}\ \text{(fun)}$$

Semantic Derivation:

$$\dfrac{\dfrac{\dfrac{}{\texttt{2} \Downarrow 2}\ \text{(int-eval)}}{\texttt{Some 2} \Downarrow \text{Some}(2)}\ \text{(some-eval)} \quad \dfrac{}{\texttt{2} \Downarrow 2}\ \text{(int-eval)}}{\texttt{match Some 2 with | None -> 0 | Some y -> y} \Downarrow 2}\ \text{(opt-eval-some)}$$

Note the right premise comes from the fact that the side condition: $[2/\texttt{y}]\texttt{y} = \texttt{2}$.[1]

[1] There's a somewhat subtle point here. When we substitute a value into an expression, we also convert it into an expression. In this case when we substitute 2 for `y` in `y`, it becomes *the expression* `2`, not the value 2.