Ungraded Problem Set #3

1. Using the typing rules presented on the last page, give a type derivation for each of the following assuming the specified type context (Γ) to start. If a type cannot be found for a term, clearly indicate where the derivation fails (show the derivation to that point).

   - `{x=2, y=3, color=c}`
     where $\Gamma = \{\mathrm{c} : \mathrm{Color}\}$
   - `{x=2, y=3, color=c}.color`
     where $\Gamma = \{\mathrm{c} : \mathrm{Color}\}$
   - $\lambda\mathrm{obj} : \underline{\{\mathrm{move} : \{\mathrm{x} : \mathrm{Int}, \mathrm{y} : \mathrm{Int}\} \to \{\mathrm{x} : \mathrm{Int}, \mathrm{y} : \mathrm{Int}\}, \mathrm{point} : \{\mathrm{x} : \mathrm{Int}, \mathrm{y} : \mathrm{Int}\}\}}.\ \mathrm{obj.point.x}$
     where $\Gamma = \{\}$, the bound variable's type annotation is underlined for clarity

2. Though we have not yet covered subtyping, consider a simple class hierarchy wherein a method is overridden within each subclass. As such, invocation of the method (via a reference to the parent class) invokes the proper implementation in the subclass used to instantiate the object (i.e., if B extends A and overrides the f method, then invoking f on a B object, even via a reference of type A, will invoke the f in B).

   Discuss how this relates to the use of a variant type defined as follows (using the name A as a label) with a dispatch function as given below.

   $\mathrm{A} = \langle B : \{f : BData \to Int, data : BData\}, C : \{f : CData \to Int, data : CData\}\rangle$

   $\mathrm{f} = \lambda\mathrm{a} : A\ .\ \mathrm{case\ a\ of}$
   $\quad \langle \mathrm{B} = \mathrm{b}\rangle \Rightarrow \mathrm{b.f\ \ b.data}$
   $\quad \langle \mathrm{C} = \mathrm{c}\rangle \Rightarrow \mathrm{c.f\ \ c.data}$

3. Using the typing rules presented on the last page, give a type derivation for each of the following assuming the specified type context (Γ) to start. If a type cannot be found for a term, clearly indicate where the derivation fails (show the derivation to that point).

   - $\langle \mathrm{b} = \mathrm{true}\rangle$ as $\langle \mathrm{b} : \mathrm{Bool}, \mathrm{i} : \mathrm{Int}, \mathrm{u} : \mathrm{Unit}\rangle$
     where $\Gamma = \{\}$
   - $\langle \mathrm{x} = 7\rangle$ as $\langle \mathrm{b} : \mathrm{Bool}, \mathrm{i} : \mathrm{Int}, \mathrm{u} : \mathrm{Unit}\rangle$
     where $\Gamma = \{\}$
   - $\lambda\mathrm{a} : \langle \mathrm{b} : \mathrm{Bool}, \mathrm{i} : \mathrm{Int}, \mathrm{u} : \mathrm{Unit}\rangle\ .\ \mathrm{case\ a\ of}$
     $\quad \langle \mathrm{b} = \mathrm{y}\rangle \Rightarrow y$
     $\quad \langle \mathrm{i} = \mathrm{n}\rangle \Rightarrow n$
     $\quad \langle \mathrm{u} = \mathrm{x}\rangle \Rightarrow 0$

     where $\Gamma = \{\}$
   - $\lambda\mathrm{a} : \langle \mathrm{b} : \mathrm{Bool}, \mathrm{i} : \mathrm{Int}, \mathrm{u} : \mathrm{Unit}\rangle\ .\ \mathrm{case\ a\ of}$
     $\quad \langle \mathrm{b} = \mathrm{y}\rangle \Rightarrow 0$
     $\quad \langle \mathrm{i} = \mathrm{n}\rangle \Rightarrow n$

     where $\Gamma = \{\}$
   - $\mathrm{A} = \langle B : \{f : BData \to Int, data : BData\}, C : \{f : CData \to Int, data : CData\}\rangle$

     $\lambda\mathrm{a} : A\ .\ \mathrm{case\ a\ of}$
     $\quad \langle \mathrm{B} = \mathrm{b}\rangle \Rightarrow \mathrm{b.f\ \ b.data}$
     $\quad \langle \mathrm{C} = \mathrm{c}\rangle \Rightarrow \mathrm{c.f\ \ c.data}$

     where $\Gamma = \{\}$

The terms for the expression language are to be inferred from the rules below coupled with the discussion in lecture (and in the textbook).

nv is for numeric values.

$$\Gamma \vdash \text{true} : \text{Bool} \qquad \text{(T-True)}$$

$$\Gamma \vdash \text{false} : \text{Bool} \qquad \text{(T-False)}$$

$$\frac{\Gamma \vdash t_1 : \text{Bool} \quad \Gamma \vdash t_2 : \tau \quad \Gamma \vdash t_3 : \tau}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : \tau} \qquad \text{(T-If)}$$

$$\Gamma \vdash nv : \text{Int} \qquad \text{(T-IntConst)}$$

$$\frac{t_1 : \text{I}nt}{\Gamma \vdash \text{iszero } t_1 : \text{Bool}} \qquad \text{(T-IsZero)}$$

$$\frac{\Gamma \vdash t_1 : \text{Int} \quad \Gamma \vdash t_2 : \text{Int}}{\Gamma \vdash t_1 + t_2 : \text{Int}} \qquad \text{(T-Add)}$$

$$\frac{\Gamma \vdash t_1 : \text{Int} \quad \Gamma \vdash t_2 : \text{Int}}{\Gamma \vdash t_1 - t_2 : \text{Int}} \qquad \text{(T-Sub)}$$

$$\frac{\text{x} : \tau \in \Gamma}{\Gamma \vdash \text{x} : \tau} \qquad \text{(T-Var)}$$

$$\frac{\Gamma, \text{x} : \alpha \vdash t_1 : \beta}{\Gamma \vdash \lambda \text{x} : \alpha .\ t_1 : \alpha \to \beta} \qquad \text{(T-Abs)}$$

$$\frac{\Gamma \vdash t_1 : \alpha \to \beta \quad \Gamma \vdash t_2 : \alpha}{\Gamma \vdash t_1\ t_2 : \beta} \qquad \text{(T-App)}$$

$$\frac{\text{for each } i \quad \Gamma \vdash t_i : \tau_i}{\Gamma \vdash \{l_i = t_i\ ^{i \in 1..n}\} : \{l_i : \tau_i\ ^{i \in 1..n}\}} \qquad \text{(T-Rcd)}$$

$$\frac{\Gamma \vdash t_1 : \{l_i : \tau_i\ ^{i \in 1..n}\}}{\Gamma \vdash t_1.j : \tau_j} \qquad \text{(T-Proj)}$$

$$\frac{\Gamma \vdash t_j : \tau_j}{\Gamma \vdash \langle l_j = t_j \rangle \text{ as } \langle l_i : \tau_i\ ^{i \in 1..n} \rangle : \langle l_i : \tau_i\ ^{i \in 1..n} \rangle} \qquad \text{(T-Variant)}$$

$$\frac{\Gamma \vdash t_0 : \langle l_i : \tau_i\ ^{i \in 1..n} \rangle \quad \text{for each } i \quad \Gamma, x_i : \tau_i \vdash t_i : \tau}{\Gamma \vdash \text{case } t_0 \text{ of } \langle l_i = x_i \rangle \Rightarrow t_i\ ^{i \in 1..n} : \tau} \qquad \text{(T-Case)}$$

$$\Gamma \vdash \text{unit} : \text{Unit} \qquad \text{(T-Unit)}$$