Assignment 8

Name: Ashish Kosana
UML ID: 02148256
Collaborators: NONE

Make sure that the remaining pages of this assignment do not contain any identifying information.

1 Type Inference (30 points)

(a) Inference rules for pairs and projection

1. Pair Expression (e_1, e_2)**:** To infer the type of a pair expression, we must:

i. Infer the type of the first element e_1, resulting in type τ_1 with constraints C_1.
ii. Infer the type of the second element e_2, resulting in type τ_2 with constraints C_2.
iii. Combine the two into a pair type $(\tau_1 \times \tau_2)$ with combined constraints $C_1 \cup C_2$.

$$\frac{\Gamma \vdash e_1 : \tau_1 \ \nabla \ C_1 \quad \Gamma \vdash e_2 : \tau_2 \ \nabla \ C_2}{\Gamma \vdash (e_1, e_2) : (\tau_1 \times \tau_2) \ \nabla \ C_1 \cup C_2}$$

2. First Projection $\#1e$**:** For the projection $\#1e$, we must:

i. Ensure e is of pair type $(\tau_1 \times \tau_2)$, with constraints C.
ii. Infer the type of the first element as τ_1.

$$\frac{\Gamma \vdash e : (\tau_1 \times \tau_2) \ \nabla \ C}{\Gamma \vdash \#1e : \tau_1 \ \nabla \ C}$$

3. Second Projection $\#2e$**:** For the projection $\#2e$, we must:

i. Ensure e is of pair type $(\tau_1 \times \tau_2)$, with constraints C.
ii. Infer the type of the second element as τ_2.

$$\frac{\Gamma \vdash e : (\tau_1 \times \tau_2) \ \nabla \ C}{\Gamma \vdash \#2e : \tau_2 \ \nabla \ C}$$

(b) Inference rules for conditionals and integer inequality

1. Conditional Expression if e_1 **then** e_2 **else** e_3**:** To type-check a conditional expression:

i. Ensure the condition e_1 has type bool with constraints C_1.
ii. Ensure both branches e_2 and e_3 have the same type τ, with constraints C_2 and C_3, respectively.
iii. Combine all constraints into $C_1 \cup C_2 \cup C_3$.

$$\frac{\Gamma \vdash e_1 : \text{bool} \ \nabla \ C_1 \quad \Gamma \vdash e_2 : \tau \ \nabla \ C_2 \quad \Gamma \vdash e_3 : \tau \ \nabla \ C_3}{\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \ \nabla \ C_1 \cup C_2 \cup C_3}$$

2. Integer Equality Expression $e_1 = e_2$**:** To type-check $e_1 = e_2$:

i. Ensure e_1 and e_2 both have type int, with constraints C_1 and C_2, respectively.
ii. The result type is bool, with combined constraints $C_1 \cup C_2$.

$$\frac{\Gamma \vdash e_1 : \text{int} \ \nabla \ C_1 \quad \Gamma \vdash e_2 : \text{int} \ \nabla \ C_2}{\Gamma \vdash e_1 = e_2 : \text{bool} \ \nabla \ C_1 \cup C_2}$$

(c) Inference rules for let

Let Expression let $x = e_1$ in e_2: For a `let` expression:

i. Infer the type of e_1 as τ_1, with constraints C_1.
ii. Extend the typing context Γ by adding $x : \tau_1$.
iii. Infer the type of the body e_2 as τ_2, with constraints C_2.
iv. The overall type is τ_2, and the combined constraints are $C_1 \cup C_2$.

$$\frac{\Gamma \vdash e_1 : \tau_1 \ \nabla \ C_1 \quad \Gamma, x : \tau_1 \vdash e_2 : \tau_2 \ \nabla \ C_2}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : \tau_2 \ \nabla \ C_1 \cup C_2}$$