CSE526: Principles of Programming Languages (Spring 2003)

Scott Stoller

hw8 (version 22:00, 1may2003), due in class on 8 May 2003

Problem 1

For each of the closed expressions given in Exercise 15.1, determine all of the valid types for the expression in the simple type system described in Section 15.2 (without recursive types). You do not need to give proofs of these typing judgments.

You should describe families of types schematically. For example, if one of the given expressions were $(\lambda x.x)$, your answer would be: the valid types for $(\lambda x.x)$ are exactly those of the form $\theta \rightarrow \theta$, for all types θ.

Problem 2

Consider the typed functional programming language with subtyping from chapter 16, in which int $\leq$ real. Which of the following subtyping relations hold? Briefly justify your answers.

(a) (int → int) → (real → real) $\leq$ (int → real) → (real → real)
(b) (int → int) → (real → real) $\leq$ (real → int) → (real → real)
(c) (int → int) → (real → real) $\leq$ (int → real) → (int → real)

Problem 3

Consider the function eqlist defined in Section 11.5. Write a polymorphic version of eqlist in the explicitly-typed polymorphic language of chapter 17. Let's call this polymorphic version "peqlist". peqlist takes 4 arguments: (1) a type θ, (2) a function that provides an "equality test" for type θ (this function takes two values of type θ as arguments and returns true iff the two values are equal), (3) a list of elements of type θ and (4) another list of elements of type θ. peqlist, like eqlist, returns true iff the two lists are structurally equal. For example, the following expression evaluates to false. The dots indicate the parts that you should write.

letrec $\mathrm{peqlist}_{...} \equiv \cdots$ in
peqlist[int] $(\lambda\ \mathrm{i}_{\mathrm{int}}.\ \lambda\ \mathrm{j}_{\mathrm{int}}.\ \mathrm{i{=}j})\ (0{::}1{::}\mathrm{nil}_{\mathrm{int}})\ (0{::}\mathrm{nil}_{\mathrm{int}})$