Advanced Programming Languages (COP 4930/CIS 6930) [Spring 2015]

Assignment V

Due Date: Monday 3/23/15 at 5pm

Assignment Description

Do the following by yourself (please don't discuss solutions until after the due date).

Consider a language L with the following types.

$\tau ::= \bot \mid \top \mid \{l_1{:}\tau_1 \,..\, l_n{:}\tau_n\} \mid \tau_1 \texttt{->} \tau_2 \mid \mu t.\tau \mid t$

Assume that all types under consideration start out with no free (type) variables, alpha-conversion for type variables has already been defined, and all types under consideration start out containing "uniquified" type variables.

a) Define the subtyping relation for L using deterministic (algorithmic) rules.
b) Define a join relation (least upper bound) for types in L, again deterministically.

For a concrete example of joins, let τ_1 be $\bot\texttt{->}\bot$ and τ_2 be $\top\texttt{->}\top$. Neither τ_1 nor τ_2 is a subtype of the other. Nonetheless, we can compute the join of τ_1 and τ_2 as $\bot\texttt{->}\top$. If the "then" branch of an if-then-else expression has type τ_1, and the "else" branch has type τ_2, then $\bot\texttt{->}\top$ could be used as the type of the whole conditional expression because both τ_1 and τ_2 are subtypes of $\bot\texttt{->}\top$, and no other type more precisely characterizes the types of both branches.