Homework 8: Lectures 15 & 16

CS 440: Programming Languages and Translators, Spring 2020

Due Fri Mar 27, 11:59 pm

What to submit

There's no programming assignment, so just submit your written work. Remember the new requirements: Include the names of everyone in your group in the submission (if you work alone, please say so); group members who don't submit should add a short file with the names of everyone in your group (including themselves), and submit that to Blackboard (in the HW 8 folder).

Problems [50 pts]

Lecture 15: Typechecking & Substitution [12 pts]

1. [12 = 2*6 pts] Are the following pairs of types compatible? If so, what are the instantiations and the resulting common type? E.g., (α, α, β) and (α, β, β) are compatible with instantiation $\alpha = \beta$, result (β, β, β) [or instantiation $\beta = \alpha$, result (α, α, α)]

   a. $(\alpha, \texttt{int}, \delta, \alpha, \beta)$ and $(\texttt{int}, \delta, \texttt{char}, \beta, \delta)$.

   b. $(\beta, (\gamma, \eta), (\beta, \alpha))$ and $((\alpha, \gamma), \zeta, (\beta, \alpha))$.

Lecture 16: Unification [8 pts]

2. [8 = 2 * 4 pts] For the given set of unifiers, give the most general unifier(s), if any, plus the substitutions that take each mgu to the non-mgus. If none of the unifiers is most general, give a brief explanation why. As an example, [`X` ↦ `f(Z)`] is more general than [`X` ↦ `f(Y*Y)`] using [`Z` ↦ `Y*Y`].

   a. [`A` ↦ `B*C`, `D` ↦ `B*C` → `B*C`], [`A` ↦ `B*B`, `D` ↦ `B*B` → `B*B`], and [`A` ↦ `E`, `D` ↦ `E` → `E`].

   b. [`A` ↦ `t(bb, C, D)`], [`A` ↦ `t(bb, cc, D)`], and [`A` ↦ `t(B, C, dd)`].

Lecture 16: Unification Algorithm [30 pts]

3. [30 = 5+5+10+10 pts] Solve the following unification problems. Here's the format to use (it's simplified from Lecture 16).

Substitution	**Problem**
[]	{`Y` = `3`, `X` = `Y`}
[`Y` ↦ `3`]	{`X` = `3`}
[`X` ↦ `3`, `Y` ↦ `3`]	{}

   Notes: In the first line, [] or ∅ means empty substitution. The order of equations in a problem doesn't matter;

so it doesn't matter which equation you pick*. In the last line, the order of bindings in a substitution doesn't matter, so [`Y` $\mapsto$ `3`, `X` $\mapsto$ `3`] is fine too.

a. {`X` = `d(Y, Z)`, `X` = `d(Z, Y)`}

b. {`d(a, b)` = `d(A, B)`, `f(A)` = `f(B)`}

c. {`p(X, Y)` = `p(X, p(X, Z))`, `Y` = `p(A, y)`, `X` = `p(x, Z)`}

d. {`t(A, B, d(a, c))` = `t(p(a, E), B, C)`, `p(d(E, c), d(a, F))` = `p(d(b, F), C)`}

* The final unifier is unique, but only up to renaming.

