Programming Languages (COP 4020/6021) [Spring 2018]

Assignment V

Objectives

1. To understand several programming-language properties used to prove type safety, including Weakening, Substitution, Inversion, and Canonical Forms.
2. To understand type safety at a technical level by proving it for a small language.

Due Date: Monday, April 9, 2018 (at 5pm).

Assignment Description

Do the following by yourself.

Recall the following language L from previous assignments:

types $\tau ::= \text{int} \mid \tau_1 \times \tau_2$

expressions $e ::= i \mid e_1+e_2 \mid x \mid (e_1,e_2) \mid \text{case } e_1 \text{ of } p \Rightarrow e_2 \text{ else } \_ \Rightarrow e_3$

patterns $p ::= x \mid \_ \mid i \mid (p_1,p_2)$

values $v ::= i \mid (v_1,v_2)$

value-substitution sequences $\sigma ::= \varepsilon \mid \sigma[v/x]$

In this assignment, you'll use the static and dynamic semantics of L as discussed in class, in addition to the following judgment, which defines the context that a value-substitution sequence implies.

$\boxed{\Gamma \vdash \sigma : \Gamma'}$

$$\frac{\forall i \in \{1..n\} : (\Gamma \vdash v_i : \tau_i)}{\Gamma \vdash [v_1/x_1]..[v_n/x_n] : \{x_1:\tau_1, .., x_n:\tau_n\}} \ \sigma\text{-Ctxt}$$

First state and prove all the standard type-safety lemmas for L (i.e., Weakening, Substitution, Inversion, and Canonical Forms). Then state and prove that L is type safe. Your progress and preservation proofs may assume that the following properties hold.

Lemma 1:

$\forall \Gamma,v,\tau,p,\sigma,\Gamma' : (\Gamma \vdash v:\tau \ \wedge \ v \sim p \text{ with } \sigma \ \wedge \ p:(\tau,\Gamma')) \ \Rightarrow \Gamma \vdash \sigma:\Gamma'$

Lemma 2:

$\forall \Gamma,\sigma,\Gamma',e,\tau : (\Gamma \vdash \sigma:\Gamma' \ \wedge \ \Gamma \cup \Gamma' \vdash e:\tau) \ \Rightarrow \Gamma \vdash \sigma e:\tau$

Lemma 3:

$\forall v,p : (v \not\sim p \ \text{ xor } \ (\exists \sigma: v \sim p \text{ with } \sigma))$

For extra credit, you may prove Lemmas 1-3 (up to +10% extra credit per lemma).

Hints

This assignment may require several hours of writing, but you have all the tools needed to complete it.

You may assume that case expressions, like function expressions in diML, may be alpha-converted when convenient and never declare a variable that has been declared previously. Hence, for example, you never have to consider the possibility of a Γ or σ having multiple entries for the same variable.

Your proofs may also assume all the normal properties of set operators. For example, you can assume that $S \cup S = S$, $S \cup T = T \cup S$, etc.

Submission Notes

- Write the following pledge at the end of your submission: "I pledge my Honor that I have not cheated, and will not cheat, on this assignment." Sign your name after the pledge. Not including this pledge will lower your grade 50%.
- For full credit, turn in a hardcopy (handwritten or printed) version of your solutions.
- You may submit this assignment late (i.e., between 5pm on 4/9 and 5pm on 4/11) with a 15% penalty on the whole assignment.
- Late submissions may be emailed or submitted in hardcopy.
- All emailed submissions, even if sent before the deadline, will be graded as if they were submitted late, i.e., with a 15% penalty.
- If you think there's a chance you'll be absent or late for class on the date this assignment is due, you're welcome to submit solutions early by giving them to me or the TA before or after class, or during any of our office hours.