COMP 523: Language-based security
Assignment 3 (100 points total)

Prof. B. Pientka
McGill University

October 22, 2017—**Due: Thursday, 2 November at 2:35pm**

Exercise 1 (30 points) Weak Normalization.

In class, we discussed in detail weak normalization for a language containing functions, unit, and natural numbers. Your task is to extend the proof of normalization (including the necessary lemmas) to booleans.

$$\begin{array}{llll} \text{Term} & t, s & := & \ldots \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ t\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \\ \text{Types} & T, S & := & \ldots \mid \mathsf{bool} \end{array}$$

5 pts State the reducibility relation for booleans.

5 pts Revisit the backwards closed lemma for booleans. If $t \rightarrow t'$ and $t' \in \mathcal{R}_T$ then $t \in \mathcal{R}_T$.

20 pts Prove the main lemma. If $\Gamma \vdash t : T$ and $\sigma \in \mathcal{R}_\Gamma$ then $[\sigma]t \in \mathcal{R}_T$.

Exercise 2 (70 points): Type Safety using Logical Relations

In this course, we have proven type safety sytnactically by proving progress and preservation.

Lemma 1

1. *Type Preservation: If* $\vdash t : T$ *and* $t \rightarrow t'$ *then* $\vdash t' : T$
2. *Progress If* $\vdash t : T$ *then either* t *is a value or there exists a term* t' *s.t.* $t \rightarrow t'$.

Using preservation and progress, we can show type safety.

$$t\ \mathsf{is\_safe} \quad := \quad \text{for all } s.\ \text{if } t \longrightarrow^* s \text{ then } s \text{ is a value or there exists an } s' \text{ s.t. } s \rightarrow s'$$

Theorem 2 (Type Safety) *If* $\vdash t : T$ *then* t is_safe.

10 pts Give the proof of type safety.

Next, we consider an alternative proof of type safety using logical relations[1]. Instead of showing directly that all well-typed terms are safe, we describe the set of safe terms semantically using $\mathcal{S}_T$. We then show:

[1] Technically, we only use a logical predicate in proving normalization and type safety, as both $t \in \mathcal{R}_T$ and $t \in \mathcal{S}_T$ are unary relations.

1. If $t \in \mathcal{S}_T$ then t is_safe.
2. If $\vdash t : T$ then $t \in \mathcal{S}_T$.

As in the previous proof of normalization, we require a generalization to prove the second statement.

First, we semantically define the set of safe terms at type T, described by $\mathcal{S}_T$, and the set of values at type T described by $\mathcal{V}_T$. We say that a term t is safe at type T, i.e. $t \in \mathcal{S}_T$, if for all terms s where $t \longrightarrow^* s$ and s cannot be reduced further, i.e. there is no step we can take, s must be a value.

Semantic Interpretation of Values

$v \in \mathcal{V}_{\mathsf{bool}}$ iff $v = \mathsf{true}$ or $v = \mathsf{false}$

$\lambda x{:}S.t \in V_{S\to T}$ iff for all v if $v \in \mathcal{S}_S$ then $[s/x]t \in \mathcal{S}_T$

Semantic Interpretation of Safe Expressions

$t \in \mathcal{S}_T$ iff for all s.if $t \longrightarrow^* s$ and there is no s' s.t. $s \to s'$ then $s' \in \mathcal{V}_T$

10 pts Prove that if $t \in \mathcal{S}_T$ then t is_safe.

30 pts Prove that if $\vdash t : T$ then $t\mathcal{S}_T$. Note that you need to generalize this statement and you need to define a notion of safe substitutions.

10 pts Extend the semantic interpretation of values to $T * S$ (cross products).

10 pts Extend your proof that well-typed terms are in the semantic interpretation of safe expressions.

Further Reading: If you want to read more on logical relations, check out Amal Ahmed's lectures at the Oregon Programming Languages Summer School (OPLSS). Some of the questions here are based on it. Her lectures are recorded and available online. One of the students attending OPLSS also wrote a nice summary of on proofs by logical relation.

`http://cs.au.dk/~lask/main.pdf`