CS101

Special Topics in Computer Science

Language-Based Security

Lecture 4: SLam: λ-Calculus with Security

Aleksey Nogin

October 14, 2005

"Well-Typed Programs Can Do No Wrong"

Theorem. If $\vdash e \in t$ and $e \rightarrow^* e'$, then either e' is a value, or $e' \rightarrow e''$ for some e''

"Well-Typed Programs Can Do No Wrong"

Theorem. If $\vdash e \in t$ and $e \rightarrow^* e'$, then either e' is a value, or $e' \rightarrow e''$ for some e''

In presence of type preservation, it is sufficient to show that:

If $\vdash e' \in t$, then e' is a value, or $e' \rightarrow e''$ for some e''.

"Well-Typed Programs Can Do No Wrong"

Theorem. If $\vdash e \in t$ and $e \rightarrow^* e'$, then either e' is a value, or $e' \rightarrow e''$ for some e''

In presence of type preservation, it is sufficient to show that:

If $\vdash e' \in t$, then e' is a value, or $e' \rightarrow e''$ for some e''.

Proof. Induction on e'

Reminder: λ-Calculus "Small-Step" Evaluation

Binary operations:

$$\frac{}{n_1\ op\ n_2 \rightarrow n}\ (n = n_1\ op\ n_2)$$

$$\frac{e_1 \rightarrow e_1'}{e_1\ op\ e_2 \rightarrow e_1'\ op\ e_2} \qquad \frac{e_2 \rightarrow e_2'}{v\ op\ e_2 \rightarrow v\ op\ e_2'}$$

Applications:

$$\frac{}{\lambda x : t.e\ v \rightarrow e[v/x]} \qquad \frac{e_1 \rightarrow e_1'}{e_1\ e_2 \rightarrow e_1'\ e_2} \qquad \frac{e_2 \rightarrow e_2'}{v\ e_2 \rightarrow v\ e_2'}$$

Reminder: λ-Calculus "Small-Step" Evaluation

Binary operations:

$$\frac{}{n_1\ op\ n_2 \rightarrow n}\ (n = n_1\ op\ n_2)$$

$$\frac{e_1 \rightarrow e_1'}{e_1\ op\ e_2 \rightarrow e_1'\ op\ e_2} \qquad \frac{e_2 \rightarrow e_2'}{v\ op\ e_2 \rightarrow v\ op\ e_2'}$$

Applications:

$$\frac{}{\lambda x : t.e\ v \rightarrow e[v/x]} \qquad \frac{e_1 \rightarrow e_1'}{e_1\ e_2 \rightarrow e_1'\ e_2} \qquad \frac{e_2 \rightarrow e_2'}{v\ e_2 \rightarrow v\ e_2'}$$

This is too verbose!

"Small-Step" Evaluation with Environments

Evaluation Environments ("expression with a hole"):

$$\begin{array}{rcl} E & := & [\cdot] \\ & | & E\ op\ e \\ & | & v\ op\ E \\ & | & E\ e \\ & | & v\ E \end{array}$$

Rules:

$$E[n_1\ op\ n_2] \rightarrow E[n] (\text{when } n = n_1\ op\ n_2)$$
$$E[\lambda x : t.e\ v] \rightarrow E[e[v/x]]$$

Homework II

Second homework will be posted tonight, due next Friday in class.

SLam!

Paper:
Nevin Heintze and Jon G. Riecke. *The SLam calculus: programming with secrecy and integrity*. In Proceedings of the 25th ACM Symposium on Principles of Programming Languages (POPL), pages 365–377, San Diego, CA, January 19-21, 1998.