CS101

Special Topics in Computer Science

Language-Based Security

Lecture 1

Aleksey Nogin

October 5, 2005

Traditional Approach to Security

Programs are treated as “black boxes”.

Language-Based Security

- Code analysis — type-safe languages, *etc*;
- Rewrite code — software fault isolation (SFI), *etc*;
- Certifying compilation, proof-carrying code (PCC).

Quick Information

Web Page:	`http://nogin.org/cs101/`
Time:	WF 1:00–1:55
Place:	Jorgensen 287
Instructor:	Aleksey Nogin
Units:	9 (2+0+7), pass/fail or letter grade
Office Hours:	Wed 6 PM, Moore 04 and by appointment
TA:	Nathan Gray

OCaml Syntax — Basics

Operations on integers:	`1 + 2`
Operations on floats:	`1.0 +.  2.0`
Variable definitions:	`let x = 2 + 4`
Function definitions:	`let f x = x + 1`
Anonymous functions:	`fun x -> x + 2`

λ-Calculus: Syntax

Expressions:

$e := n$	(Numerical constants)
$\mid e_1\ op\ e_2$	(Binary operations; $op = +, -, *$)
$\mid \lambda x : t.e$	(Functions — "`fun x:t -> e`")
$\mid e_1\ e_2$	(Function applications)

Types:

$t :=$ `int`	(Integers)
$\mid t_1 \rightarrow t_2$	(Functions)

λ-Calculus: Evaluation

Notation:

$e \longrightarrow v$ Expression e *evaluates to* (computes to) *value* v

$$\frac{A_1 \quad \cdots \quad A_n}{C}$$

Whenever *assumptions* A_i are true, the *conclusion* C must be true as well. If n is 0, C must be true unconditionally.

Numbers and functions are values:

$$\frac{}{n \longrightarrow n} \qquad \frac{}{\lambda x : t.e \longrightarrow \lambda x : t.e}$$

Binary operations: $\dfrac{e_1 \longrightarrow n_1 \quad e_2 \longrightarrow n_2 \quad n = n_1\ op\ n_2}{e_1\ op\ e_2 \longrightarrow n}$

λ-Calculus: Evaluation

Numbers and functions are values:

$$\frac{}{n \longrightarrow n} \qquad \frac{}{\lambda x : t.e \longrightarrow \lambda x : t.e}$$

Binary operations: $\dfrac{e_1 \longrightarrow n_1 \quad e_2 \longrightarrow n_2 \quad n = n_1\ op\ n_2}{e_1\ op\ e_2 \longrightarrow n}$

Function applications:

$$\frac{e_1 \longrightarrow \lambda x : t.e_1' \quad e_2 \longrightarrow v \quad e_1'[v/x] \longrightarrow v'}{e_1\ e_2 \longrightarrow v'}$$

Here $e_1'[v/x]$ stands for the result of *substitution* of the value v for the variable x in expression e_1'.

λ-Calculus: Example

In OCaml we had:

```
# let twice f x = f (f x);;
val twice : ('a -> 'a) -> 'a -> 'a = <fun>
# let add1 x = x + 1;;
val add1 : int -> int = <fun>
# let add2 = twice add1;;
val add2 : int -> int = <fun>
# add2 3;;
- : int = 5
```

In λ-Calculus, `add2 3` is $\Big((\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1)\Big)\ 3$

λ-Calculus: Evaluation Example

$$\Big((\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1)\Big)\ 3 \longrightarrow$$

λ-Calculus: Evaluation Example

$$(\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1) \longrightarrow$$

$$\Big((\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1)\Big)\ 3 \longrightarrow$$

λ-Calculus: Evaluation Example

$$\lambda f.\lambda x.f\ (f\ x) \longrightarrow$$

$$(\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1) \longrightarrow$$

$$\Big((\lambda f.\lambda x.f\ (f\ x))\ (\lambda x.x+1)\Big)\ 3 \longrightarrow$$

λ-Calculus: Evaluation Example

$$\frac{\dfrac{}{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x)}}{\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\left(\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1)\right)\ 3 \longrightarrow}{}$$

λ-Calculus: Evaluation Example

$$\frac{\dfrac{}{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x)} \quad \dfrac{}{\lambda x.x+1 \longrightarrow}}{\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\left(\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1)\right)\ 3 \longrightarrow}{}$$

λ-Calculus: Evaluation Example

$$\frac{\dfrac{}{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x)} \quad \dfrac{}{\lambda x.x+1 \longrightarrow \lambda x.x+1}}{\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\left(\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1)\right)\ 3 \longrightarrow}{}$$

λ-Calculus: Evaluation Example

$$\frac{\dfrac{}{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x)} \quad \dfrac{}{\lambda x.x+1 \longrightarrow \lambda x.x+1} \quad \dfrac{}{\lambda x.(\lambda y.y+1)\ \left((\lambda z.z+1)\ x\right) \longrightarrow}}{\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\left(\left(\lambda f.\lambda x.f\ (f\ x)\right)\ (\lambda x.x+1)\right)\ 3 \longrightarrow}{}$$

λ-Calculus: Evaluation Example

$$\frac{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x) \qquad \lambda x.x+1 \longrightarrow \lambda x.x+1 \qquad \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big) \longrightarrow \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big)}{\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1) \longrightarrow \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big)}{\Big(\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1)\Big)\ 3 \longrightarrow}$$

λ-Calculus: Evaluation Example

$$\frac{\lambda f.\lambda x.f\ (f\ x) \longrightarrow \lambda f.\lambda x.f\ (f\ x) \qquad \lambda x.x+1 \longrightarrow \lambda x.x+1 \qquad \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big) \longrightarrow \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big)}{\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1) \longrightarrow}$$

$$\frac{\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1) \longrightarrow \lambda x.(\lambda y.y+1)\ \big((\lambda z.z+1)\ x\big)}{\Big(\big(\lambda f.\lambda x.f\ (f\ x)\big)\ (\lambda x.x+1)\Big)\ 3 \longrightarrow}$$