5118014 Programming Language Theory

Ch 8. First-order Functions

Shin Hong

Function

- a function is a sequence of operations that implements a high-level operation
  - a function receives input values and produces an output value
  - fundamental aspect of program abstraction

- first-order function cannot have functions as input or output
  - while first-class function can do both (see Chapter 9)

F1VAE

- add first-order functions to VAE
- a function must be defined at top-level, not inside an expression
  - closure is not possible with first-order function
  - now a program consists of a set of function definitions and an expression

Syntax

- `<program> ::= <func-list> <expr>`
- `<func-list> ::= <func> <func-list> | ϵ`
- `<func> ::= def x(x) = <expr>`
- `<expr> ::= … | x(<expr>)`

- currently, we restrict a function to have one parameter only to keep our discussion simple

Semantics (1/3)

- function environment
  - a map from identifiers to function definitions
  - $FEnv = Id \nrightarrow FunDef$
  - $\Lambda \in FEnv$
- adding function environment to semantics function
  - $\Rightarrow \subseteq Env \times FEnv \times E \times \mathbb{Z}$
  - $(\sigma, \Lambda, e, n) \in \Rightarrow$
  - $\sigma, \Lambda \vdash e \Rightarrow n$

Semantics (2/3)

$$\sigma, \Lambda \vdash n \Rightarrow n \quad [\textsc{Num}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma, \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash e_1 + e_2 \Rightarrow n_1 + n_2} \quad [\textsc{Add}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma, \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash e_1 - e_2 \Rightarrow n_1 - n_2} \quad [\textsc{Sub}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma[x \mapsto n_1], \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash \mathsf{val}\ x{=}e_1\ \mathsf{in}\ e_2 \Rightarrow n_2} \quad [\textsc{Val}]$$

$$\frac{x \in \mathit{Domain}(\sigma)}{\sigma, \Lambda \vdash x \Rightarrow \sigma(x)} \quad [\textsc{Id}]$$

Semantics (3/3)

$$\frac{\sigma, \Lambda \vdash e \Rightarrow n' \qquad x \in Domain(\Lambda) \qquad \Lambda(x) = \text{def } x(x')=e' \qquad [x' \mapsto n'], \Lambda \vdash e' \Rightarrow n}{\sigma, \Lambda \vdash x(e) \Rightarrow n} \text{ [Call]}$$

- $x(e)$ evaluates to n under σ and Λif
  - $x \in \Lambda$,
  - $\Lambda(x)$ is def $x(x') = e'$,
  - $\sigma, \Lambda \vdash e \Rightarrow n'$, and
  - $[x' \mapsto n'], \Lambda \vdash e' \Rightarrow n$ // the function is evaluated regardless of σ (static scoping)

Exercise

$$\sigma, \Lambda \vdash n \Rightarrow n \quad [\textsc{Num}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma, \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash e_1 + e_2 \Rightarrow n_1 + n_2} \quad [\textsc{Add}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma, \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash e_1 - e_2 \Rightarrow n_1 - n_2} \quad [\textsc{Sub}]$$

$$\frac{\sigma, \Lambda \vdash e_1 \Rightarrow n_1 \qquad \sigma[x \mapsto n_1], \Lambda \vdash e_2 \Rightarrow n_2}{\sigma, \Lambda \vdash \text{val } x{=}e_1 \text{ in } e_2 \Rightarrow n_2} \quad [\textsc{Val}]$$

$$\frac{x \in \mathit{Domain}(\sigma)}{\sigma, \Lambda \vdash x \Rightarrow \sigma(x)} \quad [\textsc{Id}]$$

$$\frac{\sigma, \Lambda \vdash e \Rightarrow n' \qquad x \in \mathit{Domain}(\Lambda) \qquad \Lambda(x) = \text{def } x(x')=e' \qquad [x' \mapsto n'], \Lambda \vdash e' \Rightarrow n}{\sigma, \Lambda \vdash x(e) \Rightarrow n} \quad [\textsc{Call}]$$

Exercise 8.1 With the following list of function definitions in F1VAE:

```
def twice(x)=x + x
def x(y)=y
def f(x)=x + 1
def g(g)=g
```

Show the results of evaluating the following expressions under the error, describe which error it is.

1. `twice(twice)`
2. `val x=5 in x(x)`
3. `g(3)`
4. `g(f)`
5. `g(g)`