First Exam
CS 3366 Programming Languages

KEY

Wednesday March 1, 2017

Instructor Muller
Boston College
Spring 2017

Before reading further, please arrange to have an empty seat on either side of you. Now that you are seated, please write your name **on the back** of this exam.

This is a closed-notes and closed-book exam. Computers, calculators, and books are prohibited.

- Partial credit will be given so be sure to show your work.
- Feel free to write helper functions if you need them.
- **Please write neatly.**

Problem	Points	Out Of
1		7
2		6
3		7
Total		20

1. (7 Points): Context-Free Grammars

JavaScript is king of the hill in web programming. JavaScript objects can be created using record notation. A slight variation of this notation, JavaScript Object Notation (JSON), has become a standard language for data exchange.

JSON forms are made up of key/value pairs where the keys are specified as strings and the values can be strings, numbers, booleans, JSON forms or lists of JSON values. For example

```
{ "name" : "Alice",
  "eyog" : 2017,
  "classes" : [ {"dept" : "CSCI", "number" : 3366, "passfail" : true},
                {"dept" : "ECON", "number" : 2234, "passfail" : false}],
  "clubs" : []
}
```

Assuming the existence of terminal symbols **String**, **Boolean** and **Number**, give a context-free grammar for JSON forms. Use the symbol **JSON** as your start symbol. Note that records and lists can both be empty.

Answer:

```
JSON     ::= {} | { Bindings }
Bindings ::= Binding | Binding , Bindings
Binding  ::= String : Value
Value    ::= String | Number | Boolean | JSON | List
List     ::= [] | [ Values ]
Values   ::= Value | Value , Values
```

2. (6 Points): Evaluation

This question is about the operational semantics of Venus. Figure 1 shows a *call-by-value* semantics for Venus. The axiom system defines an evaluation function as shown in the caption. Let $A_0 = \epsilon$ be the empty environment and let `E` be the following Venus program:

```
let x : int = 2 in let y : int = x in x + y
```

Is $(A_0, \mathtt{E}, 4)$ in eval_V? If not, why not. If so, prove it by showing a derivation.

Answer:

Yes it is. Let A_x abbreviate $A_0[x \mapsto 2]$ and A_y abbreviate $A_x[y \mapsto 2]$. We omit the type annotations.

$$\dfrac{\dfrac{\dfrac{}{A_0 \vdash 2 \Downarrow 2}}{A_0 \vdash x = 2 \Rightarrow A_x} \quad ; \quad \dfrac{\dfrac{\dfrac{A_x(x) = 2}{A_x \vdash x \Downarrow 2}}{A_x \vdash y = x \Rightarrow A_y} \quad ; \quad \dfrac{\dfrac{A_y(x) = 2}{A_y \vdash x \Downarrow 2} \quad ; \quad \dfrac{A_y(y) = 2}{A_y \vdash y \Downarrow 2} \quad ; \quad 2 + 2 = 4}{A_y \vdash x + y \Downarrow 4}}{A_x \vdash \mathtt{let}\ y = x\ \mathtt{in}\ x + y \Downarrow 4}}{A_0 \vdash \mathtt{let}\ x = 2\ \mathtt{in}\ \mathtt{let}\ y = x\ \mathtt{in}\ x + y \Downarrow 4}$$

Figure 2 shows a *call-by-name* semantics for Venus. The system defines an evaluation function eval_{N} as shown in the caption. Let $A_0 = \epsilon$ be the empty environment and let `E` be the program from before:

```
let x : int = 2 in let y : int = x in x + y
```

Is $(A_0, \texttt{E}, 4)$ in eval_{N}? If not, why not. If so, prove it by showing a derivation.

Answer:

Yes it is. Let C abbreviate Closure, let A_x abbreviate $A_0[x \mapsto C(A_0, 2)]$ and A_y abbreviate $A_x[y \mapsto C(A_x, x)]$. We omit the type annotations.

$$\dfrac{\dfrac{}{A_0 \vdash x = 2 \Rightarrow A_x} \quad ; \quad \dfrac{\dfrac{}{A_x \vdash y = x \Rightarrow A_y} \quad ; \quad \dfrac{\ldots \; ; \quad \dfrac{A_y(y) = C(A_x, x); \quad \dfrac{A_x(x) = C(A_0, 2); \quad \dfrac{}{A_0 \vdash 2 \Downarrow 2}}{A_x \vdash x \Downarrow 2}}{A_y \vdash y \Downarrow 2} \quad ; \; \ldots}{A_y \vdash x + y \Downarrow 4}}{A_x \vdash \texttt{let}\ y = x\ \texttt{in}\ x + y \Downarrow 4}}{A_0 \vdash \texttt{let}\ x = 2\ \texttt{in}\ \texttt{let}\ y = x\ \texttt{in}\ x + y \Downarrow 4}$$

3. (7 Points): Implementation

The defining characteristic of a variable is that it can be replaced. Let E_1 and E_2 be Venus-programs. The notation $E_1[\texttt{x} := E_2]$ denotes E_1 with free occurrences of x replaced by E_2.

$$
\begin{array}{rcl}
\texttt{V}[x := E] & \equiv & \texttt{V} \\
\texttt{y}[x := E] & \equiv & \texttt{y} \\
\texttt{x}[x := E] & \equiv & E \\
(E_1 \ \texttt{op}\ E_2)[x := E] & \equiv & (E_1[x := E] \ \texttt{op}\ E_2[x := E]) \\
\texttt{i2r}(E_1)[x := E_2] & \equiv & \texttt{i2r}(E_1[x := E_2]) \\
\texttt{r2i}(E_1)[x := E_2] & \equiv & \texttt{r2i}(E_1[x := E_2]) \\
(\texttt{let x} = E_1 \ \texttt{in}\ E_2)[x := E_3] & \equiv & (\texttt{let x} = E_1[x := E_3] \ \texttt{in}\ E_2) \\
(\texttt{let y} = E_1 \ \texttt{in}\ E_2)[x := E_3] & \equiv & (\texttt{let z} = E_1[x := E_3] \ \texttt{in}\ E_2[y := z][x := E_3]) \ \texttt{z} \text{ fresh}
\end{array}
$$

With the following representation of Venus-programs

```
type t =
  | Literal of {typ : Typ.t; bits : int}
  | Var of Symbol.t
  | App of {rator : t; rands : t list}
  | Let of {decl : decl; body : t}
and
  decl = VarDef of {bv : Symbol.t; defn : t}
```

write an OCaml function `val substitute : t -> Symbol.t -> t -> t`. You may assume the existence of

```
val Symbol.fresh : unit -> Symbol.t
val Symbol.equal : Symbol.t -> Symbol.t -> boolean
```

Answer:

```
let rec substitute e1 x e2 =
  match e1 with
  | Literal _ -> e1
  | Var y -> if (Symbol.equal x y) then e2 else e1
  | App {rator; rands} ->
    App {rator = rator; rands = List.map (fun e1 -> substitute e1 x e2) rands}
  | Let {decl = VarDef {bv; defn}; body} ->
    let defn' = substitute defn x e2
    in
    if (Symbol.equal bv x) then
      Let {decl = VarDef {bv = bv; defn = defn'}; body = body}
    else
      let z = Symbol.fresh() in
      let body' = substitute body bv (Var z) in
      let body'' = substitute body' x e2
      in
      Let {decl = VarDef {bv = z; defn = defn'}; body = body''}
```

This page almost empty.

$$\frac{}{A \vdash \mathtt{i} \Downarrow \mathtt{i}}\,(\mathtt{int}) \qquad \frac{}{A \vdash \mathtt{r} \Downarrow \mathtt{r}}\,(\mathtt{real})$$

$$\frac{A \vdash \mathtt{E} \Downarrow \mathtt{i};\ \mathtt{i2r}(\mathtt{i}) = \mathtt{r}}{A \vdash \mathtt{i2r}(\mathtt{E}) \Downarrow \mathtt{r}}\,(\mathtt{i2r}) \qquad \frac{A \vdash \mathtt{E} \Downarrow \mathtt{r};\ \mathtt{r2i}(\mathtt{r}) = \mathtt{i}}{A \vdash \mathtt{r2i}(\mathtt{E}) \Downarrow \mathtt{i}}\,(\mathtt{r2i})$$

$$\frac{A \vdash \mathtt{E}_1 \Downarrow \mathtt{V}_1;\ A \vdash \mathtt{E}_2 \Downarrow \mathtt{V}_2;\ \mathrm{Apply}_2(\mathtt{o}, \mathtt{V}_1, \mathtt{V}_2) = \mathtt{V}}{A \vdash \mathtt{E}_1\ \mathtt{o}\ \mathtt{E}_2 \Downarrow \mathtt{V}}\,(\mathtt{BinOp})$$

$$\frac{A(\mathtt{x}) = \mathtt{V}}{A \vdash \mathtt{x} \Downarrow \mathtt{V}}\,(\mathtt{Id})$$

$$\frac{A \vdash D \Rightarrow A';\ A' \vdash \mathtt{E} \Downarrow \mathtt{V}}{A \vdash \mathtt{let}\ D\ \mathtt{in}\ \mathtt{E} \Downarrow \mathtt{V}}\,(\mathtt{Let}) \qquad \frac{A \vdash \mathtt{E} \Downarrow \mathtt{V}}{A \vdash \mathtt{x} : \tau = \mathtt{E} \Rightarrow A[\mathtt{x} \mapsto \mathtt{V}]}\,(\mathtt{Decl})$$

Figure 1: $\mathrm{eval}_{\mathrm{V}} = \{(A, \mathtt{E}, \mathtt{V}) \mid A \vdash \mathtt{E} \Downarrow \mathtt{V}\}$.

$$\frac{}{A \vdash \texttt{i} \Downarrow \texttt{i}}\ (\texttt{int}) \qquad \frac{}{A \vdash \texttt{r} \Downarrow \texttt{r}}\ (\texttt{real})$$

$$\frac{A \vdash \texttt{E} \Downarrow \texttt{i};\ \texttt{i2r(i)} = \texttt{r}}{A \vdash \texttt{i2r(E)} \Downarrow \texttt{r}}\ (\texttt{i2r}) \qquad \frac{A \vdash \texttt{E} \Downarrow \texttt{r};\ \texttt{r2i(i)} = \texttt{i}}{A \vdash \texttt{r2i(E)} \Downarrow \texttt{i}}\ (\texttt{r2i})$$

$$\frac{A \vdash \texttt{E}_1 \Downarrow \texttt{V}_1;\ A \vdash \texttt{E}_2 \Downarrow \texttt{V}_2;\ \mathrm{Apply}_2(\texttt{o}, \texttt{V}_1, \texttt{V}_2) = \texttt{V}}{A \vdash \texttt{E}_1\ \texttt{o}\ \texttt{E}_2 \Downarrow \texttt{V}}\ (\texttt{BinOp})$$

$$\frac{A(\texttt{x}) = \texttt{Closure}(A', \texttt{E});\ A' \vdash \texttt{E} \Downarrow \texttt{V}}{A \vdash \texttt{x} \Downarrow \texttt{V}}\ (\texttt{Id})$$

$$\frac{A \vdash D \Rightarrow A';\ A' \vdash \texttt{E} \Downarrow \texttt{V}}{A \vdash \texttt{let}\ D\ \texttt{in}\ \texttt{E} \Downarrow \texttt{V}}\ (\texttt{Let})$$

$$\frac{}{A \vdash \texttt{x} : \tau = \texttt{E} \Rightarrow A[\texttt{x} \mapsto \texttt{Closure}(A, E)]}\ (\texttt{Decl})$$

Figure 2: $\mathrm{eval}_{\mathrm{N}} = \{(A, \texttt{E}, \texttt{V}) \mid A \vdash \texttt{E} \Downarrow \texttt{V}\}$.