

# Computer Language Processing

## Exercise Sheet 06

November 2, 2022

Welcome to the sixth exercise session of CS320!

### Exercise 1

Consider a simple programming language with integer arithmetic, boolean expressions and user-defined functions.

$T := \mathbf{Int} \mid \mathbf{Bool} \mid (T_1, \dots, T_n) \Rightarrow T$   
 $t := \mathbf{true} \mid \mathbf{false} \mid c$   
     $\mid t_1 == t_2 \mid t_1 + t_2 \mid t_1 \ \&\& \ t_2$   
     $\mid \mathbf{if} \ (t_1) \ t_2 \ \mathbf{else} \ t_3$   
     $\mid f(t_1, \dots, t_n) \mid x$

Where  $c$  represents integer literals,  $==$  represents equality (between **Int**, as well as between **Bool**),  $+$  represents the usual integer addition and  $\&\&$  represents conjunction.

The meta-variable  $f$  refers to names of user-defined function and  $x$  refers to names of variables. You may assume that you have a fixed environment  $e$  which contains information about user-defined functions (i.e. the function arguments, their types, the function body and the result type).

Write down the typing rules for this language. (We give you an example for **true**)

$$\frac{}{\Gamma \vdash \mathbf{true}: \mathbf{Bool}}$$

## Exercise 2

Consider the following typing rules.

<b>VARIABLE</b> $\frac{v : T \in \Gamma}{\Gamma \vdash v : T}$	<b>INT LITERAL</b> $\frac{i \text{ is an integer literal}}{\Gamma \vdash i : \text{Int}}$	<b>STRING LITERAL</b> $\frac{s \text{ is a string literal}}{\Gamma \vdash s : \text{String}}$	<b>UNIT</b> $\frac{}{\Gamma \vdash () : \text{Unit}}$
<b>BOOLEAN LITERAL</b> $\frac{b \in \{\text{true}, \text{false}\}}{\Gamma \vdash b : \text{Boolean}}$	<b>ARITH. BIN. OPERATORS</b> $\frac{\Gamma \vdash e_1 : \text{Int} \quad \Gamma \vdash e_2 : \text{Int} \quad op \in \{+, -, *, /, \%\}}{\Gamma \vdash e_1 \text{ op } e_2 : \text{Int}}$		
<b>ARITH. COMP. OPERATORS</b> $\frac{\Gamma \vdash e_1 : \text{Int} \quad \Gamma \vdash e_2 : \text{Int} \quad op \in \{<, <=\}}{\Gamma \vdash e_1 \text{ op } e_2 : \text{Boolean}}$		<b>ARITH. NEGATION</b> $\frac{\Gamma \vdash e : \text{Int}}{\Gamma \vdash -e : \text{Int}}$	
<b>BOOLEAN BIN. OPERATORS</b> $\frac{\Gamma \vdash e_1 : \text{Boolean} \quad \Gamma \vdash e_2 : \text{Boolean} \quad op \in \{\&\&,   \}}{\Gamma \vdash e_1 \text{ op } e_2 : \text{Boolean}}$		<b>BOOLEAN NEGATION</b> $\frac{\Gamma \vdash e : \text{Boolean}}{\Gamma \vdash !e : \text{Boolean}}$	
<b>STRING CONCATENATION</b> $\frac{\Gamma \vdash e_1 : \text{String} \quad \Gamma \vdash e_2 : \text{String}}{\Gamma \vdash e_1 ++ e_2 : \text{String}}$		<b>EQUALITY</b> $\frac{\Gamma \vdash e_1 : T \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1 == e_2 : \text{Boolean}}$	
<b>SEQUENCE</b> $\frac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash e_1 ; e_2 : T_2}$		<b>LOCAL VARIABLE DEFINITION</b> $\frac{\Gamma \vdash e_1 : T_1 \quad \Gamma, n : T_1 \vdash e_2 : T_2}{\Gamma \vdash \text{val } n : T_1 = e_1 ; e_2 : T_2}$	
<b>FUNCTION/CLASS CONSTRUCTOR INVOCATION</b> $\frac{\Gamma \vdash e_1 : T_1 \quad \dots \quad \Gamma \vdash e_n : T_n \quad \Gamma \vdash f : (T_1, \dots, T_n) \Rightarrow T}{\Gamma \vdash f(e_1, \dots, e_n) : T}$			
<b>IF-THEN-ELSE</b> $\frac{\Gamma \vdash e_1 : \text{Boolean} \quad \Gamma \vdash e_2 : T \quad \Gamma \vdash e_3 : T}{\Gamma \vdash \text{if } (e_1) \{e_2\} \text{ else } \{e_3\} : T}$		<b>ERROR</b> $\frac{\Gamma \vdash e : \text{String}}{\Gamma \vdash \text{error}(e) : T}$	

Assuming an empty initial environment, type check the following expressions. Write down the derivation trees.

1. `3 + 5`
2. `val x: Int = 4; val y: Int = x + x; x * y`

Now, assume that the initial environment is:

`{(x, Boolean), (power, (Int, Int) => Int)}`

type check the following expressions. Write down their derivation trees.

3. `val x: Int = if (x) 1 else 0; x * 3`
4. `val x: Int = 7; if (x < 100) power(x, 10) else error("Too big!")`

### Exercise 3

Consider the following typing rules for a simple language with integers, pairs and functions:

$$\begin{array}{c}
 \frac{n \text{ is an integer literal}}{\Gamma \vdash n : \text{Int}} \qquad \frac{\Gamma \vdash e_1 : \text{Int} \quad \Gamma \vdash e_2 : \text{Int}}{\Gamma \vdash e_1 + e_2 : \text{Int}} \qquad \frac{\Gamma \vdash e_1 : \text{Int} \quad \Gamma \vdash e_2 : \text{Int}}{\Gamma \vdash e_1 \cdot e_2 : \text{Int}} \\
 \\
 \frac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash (e_1, e_2) : (T_1, T_2)} \qquad \frac{\Gamma \vdash e : (T_1, T_2)}{\Gamma \vdash \text{fst}(e) : T_1} \qquad \frac{\Gamma \vdash e : (T_1, T_2)}{\Gamma \vdash \text{snd}(e) : T_2} \\
 \\
 \frac{\Gamma \oplus \{(x, T_1)\} \vdash e : T_2}{\Gamma \vdash x \Rightarrow e : T_1 \Rightarrow T_2} \qquad \frac{\Gamma \vdash e_1 : T_1 \Rightarrow T_2 \quad \Gamma \vdash e_2 : T_1}{\Gamma \vdash e_1(e_2) : T_2} \qquad \frac{(x, T) \in \Gamma}{\Gamma \vdash x : T}
 \end{array}$$

Consider the following type derivation, with type variables  $T_1, \dots, T_5$ , where  $\Gamma_0 = \emptyset$  and  $\Gamma = \{(x, T_2)\}$ :

$$\begin{array}{c}
 \frac{(x, T_2) \in \Gamma}{\Gamma \vdash x : T_2} \qquad \frac{(x, T_2) \in \Gamma}{\Gamma \vdash x : T_2} \\
 \frac{\Gamma \vdash \text{fst}(x) : T_4 \quad \Gamma \vdash \text{snd}(x) : T_5}{\Gamma \vdash \text{fst}(x)(\text{snd}(x)) : T_3} \\
 \hline
 \Gamma_0 \vdash x \Rightarrow \text{fst}(x)(\text{snd}(x)) : T_1
 \end{array}$$

Circle all the correct answers:

- A. There are no assignments of  $T_1, \dots, T_5$  such that the resulting derivation is valid.
- B. In all valid derivations,  $T_3$  is equal to  $T_5$ .
- C. There does **not** exist valid derivations where  $T_1$  is Int.
- D. In all valid derivations,  $T_2$  is equal to  $(T_4, T_5)$ .
- E. In all valid derivations,  $T_3$  is equal to  $T_2 \Rightarrow T_1$ .

## Exercise 4

Infer the type of the following expressions:

1.  $x \Rightarrow x + 5$
2.  $x \Rightarrow y \Rightarrow x + y$
3.  $x \Rightarrow (y \Rightarrow x + 5)$
4.  $x \Rightarrow \text{if } (x > 0) \ x \ \text{else } -x$
5.  $x \Rightarrow \text{if } (x) \ 1 \ \text{else } x$
6.  $x \Rightarrow \text{if } (x > 0) \ (y \Rightarrow 0) \ \text{else } (y = y)$
7.  $x \Rightarrow y \Rightarrow x(y) \ \&\& \ y(0)$
8.  $x \Rightarrow y \Rightarrow i \Rightarrow x(y) + y(i) + i$