Computer Language Processing

Exercise Sheet 06

November 3, 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 := Int | Bool | (T1, ..., Tn) => T
t := true | false | c
     | t1 == t2 | t1 + t2 | t1 && t2
     | if (t1) t2 else t3
     | f(t1, ... , tn) | x
```

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

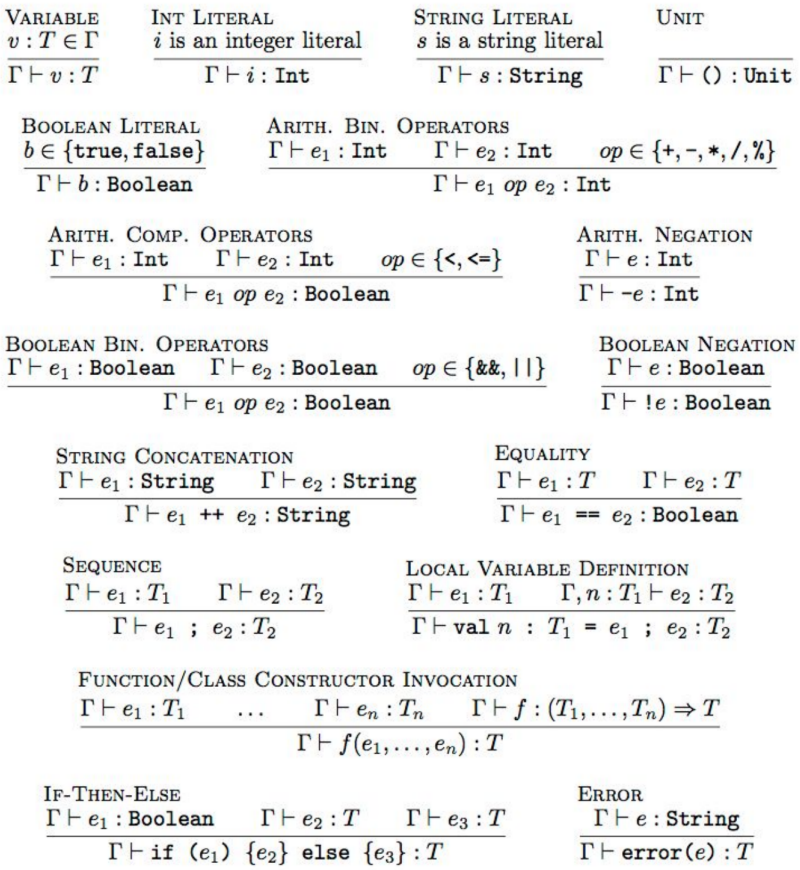
The meta-variable `f` refers to names of user-defined function and `x` refers to names of variables. You may assume that you have a fixed environment `e` which contains information about user-defined functions (i.e. the function arguments, their types, the function body and the result type).

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

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

Exercise 2

Consider the following typing rules.



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:

$$\frac{n \text{ is an integer literal}}{\Gamma \vdash n \,:\, \texttt{Int}} \qquad \frac{\Gamma \vdash e_1 \,:\, \texttt{Int} \qquad \Gamma \vdash e_2 \,:\, \texttt{Int}}{\Gamma \vdash e_1 + e_2 \,:\, \texttt{Int}} \qquad \frac{\Gamma \vdash e_1 \,:\, \texttt{Int} \qquad \Gamma \vdash e_2 \,:\, \texttt{Int}}{\Gamma \vdash e_1 \cdot e_2 \,:\, \texttt{Int}}$$

$$\frac{\Gamma \vdash e_1 \,:\, T_1 \qquad \Gamma \vdash e_2 \,:\, T_2}{\Gamma \vdash (e_1, e_2) \,:\, (T_1, T_2)} \qquad \frac{\Gamma \vdash e \,:\, (T_1, T_2)}{\Gamma \vdash \mathit{fst}(e) \,:\, T_1} \qquad \frac{\Gamma \vdash e \,:\, (T_1, T_2)}{\Gamma \vdash \mathit{snd}(e) \,:\, T_2}$$

$$\frac{\Gamma \oplus \{(x, T_1)\} \vdash e \,:\, T_2}{\Gamma \vdash x \Rightarrow e \,:\, T_1 \Rightarrow T_2} \qquad \frac{\Gamma \vdash e_1 \,:\, T_1 \Rightarrow T_2 \qquad \Gamma \vdash e_2 \,:\, T_1}{\Gamma \vdash e_1(e_2) \,:\, T_2} \qquad \frac{(x, T) \in \Gamma}{\Gamma \vdash x \,:\, T}$$

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

$$\dfrac{\dfrac{\dfrac{(x, \mathbf{T_2}) \in \Gamma}{\Gamma \vdash x \,:\, \mathbf{T_2}}}{\Gamma \vdash \mathit{fst}(x) \,:\, \mathbf{T_4}} \qquad \dfrac{\dfrac{(x, \mathbf{T_2}) \in \Gamma}{\Gamma \vdash x \,:\, \mathbf{T_2}}}{\Gamma \vdash \mathit{snd}(x) \,:\, \mathbf{T_5}}}{\dfrac{\Gamma \vdash \mathit{fst}(x)(\mathit{snd}(x)) \,:\, \mathbf{T_3}}{\Gamma_0 \vdash x \Rightarrow \mathit{fst}(x)(\mathit{snd}(x)) \,:\, \mathbf{T_1}}}$$

Circle all the correct answers:

A. There are no assignments of $\mathbf{T_1}, \ldots, \mathbf{T_5}$ such that the resulting derivation is valid.

B. In all valid derivations, $\mathbf{T_3}$ is equal to $\mathbf{T_5}$.

C. There does **not** exist valid derivations where $\mathbf{T_1}$ is `Int`.

D. In all valid derivations, $\mathbf{T_2}$ is equal to $(\mathbf{T_4}, \mathbf{T_5})$.

E. In all valid derivations, $\mathbf{T_3}$ is equal to $\mathbf{T_2} \Rightarrow \mathbf{T_1}$.

Exercise 4

Infer the type of the following expressions:

1. `x => x + 5`
2. `x => y => x + y`
3. `x => (y => x + 5)`
4. `x => if (x > 0) x else -x`
5. `x => if (x) 1 else x`
6. `x => if (x) (y => 0) else (y => y)`
7. `x => y => x(y) && y(0)`
8. `x => y => i => x(y) + y(i) + i`