Ex 1

Part 1

$$\frac{}{\Gamma \vdash \text{true} : \text{Boolean}} \qquad \frac{}{\Gamma \vdash \text{false} : \text{Boolean}}$$

$$\frac{c_e \text{ is an integer literal}}{\Gamma \vdash c_e : \text{Integer}} \qquad \frac{\Gamma \vdash t_1 : \text{Integer} \quad \Gamma \vdash t_2 : \text{Integer}}{\Gamma \vdash t_1 == t_2 : \text{Boolean}}$$

$$\frac{\Gamma \vdash t_1 : \text{Boolean} \quad \Gamma \vdash t_2 : \text{Boolean}}{\Gamma \vdash t_1 == t_2 : \text{Boolean}} \qquad \frac{\Gamma \vdash t_1 : \text{Integer} \quad \Gamma \vdash t_2 : \text{Integer}}{\Gamma \vdash t_1 + t_2 : \text{Integer}}$$

$$\frac{\Gamma \vdash t_1 : \text{Boolean} \quad \Gamma \vdash t_2 : \text{Boolean}}{\Gamma \vdash t_1 \;\&\&\; t_2 : \text{Boolean}} \qquad \frac{(x, T) \in \Gamma}{\Gamma \vdash x : T}$$

$$\frac{\Gamma \vdash t_1 : \text{Boolean} \quad \Gamma \vdash t_2 : T \quad \Gamma \vdash t_3 : T}{\Gamma \vdash \text{if } (t_1)\ t_2 \text{ else } t_3 : T}$$

$$\frac{\Gamma \vdash f : (T_1, \ldots, T_n) \Rightarrow T \quad \Gamma \vdash x_1 : T_1 \quad \ldots \quad \Gamma \vdash x_n : T_n}{\Gamma \vdash f(x_1 \ldots x_n) : T}$$

Part 2

We dende by

$$e_1 [X := e_2] \mapsto e_3$$

The fact that e_1 with all free occurences of X replaced by e_2 is e_3.

We have the following rules:

$$\frac{}{x\,[x := e] \mapsto e} \qquad \frac{x \neq y}{x\,[y := e] \mapsto x}$$

$$\frac{}{\text{true}\,[x := e] \mapsto \text{true}} \qquad \frac{}{\text{false}\,[x := e] \mapsto \text{false}}$$

$$\frac{}{c_e\,[x := e] \mapsto c_e}$$

$$\frac{e_1 [x := e_3] \mapsto e_1' \quad e_2 [x := e_3] \mapsto e_2'}{(e_1 == e_2)[x := e_3] \mapsto (e_1' == e_2')}$$

(similarly for $+$, $\&\&$, if-else and function application)

We show that, for any environment Γ, variable x, expressions e_1, e_2, e_3, and types T_1, T_2, if we have that:

1) $\Gamma \vdash e_1 : T_1$
2) $(x, T_2) \in \Gamma$
3) $\Gamma \vdash e_2 : T_2$
4) $e_1[x := e_2] \mapsto e_3$

Then, we have that $\Gamma \vdash e_3 : T_1$

We prove this by structural induction on e_1.

- Case $e_1 = x$:

  We must have that $e_3 = e_2$ as well, as the only valid substitution is

  $$x[x := e_2] \mapsto e_2$$

  Thus, to show $\Gamma \vdash e_3 : T_1$ we show $\Gamma \vdash e_2 : T_1$.

  From 1), we have that $\Gamma \vdash x : T_1$, which can only be derived from

  $$\frac{(x, T_1) \in \Gamma}{\Gamma \vdash x : T_1}$$

  Therefore, we must have that

  $$(x, T_1) \in \Gamma$$

  We also have that $(x, T_2) \in \Gamma$, from 2). Thus, we must have that $T_1 = T_2$. Since, from 3), $\Gamma \vdash e_2 : T_2$, then also $\Gamma \vdash e_2 : T_1$ #

- Case $e_1 = y \neq x$, or
  $e_1 = \text{true}$ or
  $e_1 = \text{false}$ or
  $e_1 = c_e$: In those cases, the proof is trivial since e_1 and e_3 must be the same, as the only valid substitutions are of the form
  $$e_1[x := e_2] \mapsto e_1$$

- Case $e_1 = (e_4 == e_5)$: We have that
  1) $\Gamma \vdash (e_4 == e_5) : T_1$
  2) $(x, T_2) \in \Gamma$
  3) $\Gamma \vdash e_2 : T_2$
  4) $(e_4 == e_5)[x := e_2] \mapsto e_3$

  We must have that $e_3 = (e_4' == e_5')$ for some e_4' and e_5', and that $e_4[x := e_2] \mapsto e_4'$ and $e_5[x := e_2] \mapsto e_5'$ from the only applicable derivation rule:

$$\frac{e_4[x := e_2] \mapsto e_4' \quad e_5[x := e_2] \mapsto e_5'}{(e_4 == e_5)[x := e_2] \mapsto (e_4' == e_5')}$$

  Moreover, we must have that $\Gamma \vdash (e_4 == e_5) : \text{Boolean}$, since the only applicable rules all match T_1 with Boolean. In addition, we have, for some type $T_3 \in \{\text{Integer}, \text{Boolean}\}$, that $\Gamma \vdash e_4 : T_3$ and $\Gamma \vdash e_5 : T_3$.

From induction hypothesis, we have then that

$$\Gamma \vdash e_4' : T_3 \quad \text{and that} \quad \Gamma \vdash e_5' : T_3$$

(Since conditions 1-4 are satisfied by e_4' and e_5').

Therefore, we can derive that:

$$\frac{\Gamma \vdash e_4' : T_3 \quad \Gamma \vdash e_5' : T_3}{\Gamma \vdash (e_4' == e_5') : \text{Boolean}}$$

(Since T_3 in {Boolean, Integer})

And thus we have $\Gamma \vdash (e_4' == e_5') : T_1$, which gives us

$$\Gamma \vdash e_3 : T_1 \quad \#$$

- All other cases are similar to the previous one.

#

Part 3

The only rule that differs from lectures is function application:

$$\frac{b \text{ body of } f,\ x_1 \ldots x_n \text{ args of } f \qquad b[x_1 := e_1, \ldots, x_n := e_n] \mapsto b'}{f(e_1, \ldots, e_n) \rightsquigarrow b'}$$

Where $b[x_1 := e_1, \ldots, x_n := e_n] \mapsto b'$ denotes that b' is the result of the substitution of x_1 by e_1, x_2 by e_2 and so on in b.

Note that in previous part we only treated the case when only a single variable is replaced. This is a generalisation for more than one variable.

Part 4

Progress: Proof from lecture barely changes. We just plug-in the new operational semantics rule for function application instead of the old one.

Preservation: We must show that, if $\Gamma \vdash f(e_1, \ldots, e_n) : T$ and $f(e_1, \ldots, e_n) \rightsquigarrow e$, then $\Gamma \vdash e : T$.

The only applicable typing rule is:

$$\frac{\Gamma \vdash f : (T_1, \ldots, T_n) \Rightarrow T \quad \Gamma \vdash e_1 : T_1 \ldots \Gamma \vdash e_n : T_n}{\Gamma \vdash f(e_1, \ldots, e_n) : T}$$

Therefore $\Gamma \vdash f : (T_1, \ldots, T_n) \Rightarrow T \quad \Gamma \vdash e_1 : T_1 \ldots \Gamma \vdash e_n : T_n$

Also, the only applicable operational semantics rule is:

$$\frac{b\,[x_1 := e_1, \ldots, x_n := e_n] \mapsto b'}{f(e_1, \ldots, e_n) \rightsquigarrow b'}$$

where b is the body of f and $x_1 \ldots x_n$ its arguments.

We must thus show that $\Gamma \vdash b' : T$

We assume that $\Gamma = \Gamma_0 \oplus \Gamma_1$ for some Γ_1.

(The program environment remains visible in Γ). Let $\Gamma_2 = \{(x_1, T_1), \ldots, (x_n, T_n)\}$.

Let's assume $\Gamma_0, \Gamma_1, \Gamma_2$ disjoint.

We have that $\Gamma_0 \oplus \Gamma_2 \vdash b : T$ as the body of f types checks.

We thus have that $\Gamma_0 \oplus \Gamma_1 \oplus \Gamma_2 \vdash b : T$, as we have just added irrelevant bindings.

We thus have, from (an adapted version of) Part 2's lemma:

$$\Gamma_0 \oplus \Gamma_1 \oplus \Gamma_2 \vdash b' : T$$

Since after substitution all occurences of $x_1 \dots x_n$ have been replaced in b', we have that

$$\Gamma_0 \oplus \Gamma_1 \vdash b' : T ,$$

as all bindings in Γ_2 are unused.

Thus, we have that

$$\Gamma \vdash e : T \quad \#$$

Part 2

- 1+1 can have type Pos or Integer.

$$\frac{\Gamma \vdash 1 : Pos \quad \Gamma \vdash 1 : Pos}{\Gamma \vdash 1+1 : Pos} \quad \text{and} \quad \frac{\Gamma \vdash 1 : Integer \quad \Gamma \vdash 1 : Integer}{\Gamma \vdash 1+1 : Integer}$$

- -2 * 4 can have type Neg or Integer.
- -1 * (2 + -1) can only have type Integer.
- 7 / (18 + -1) can not be typed.

Part 3

We have:

$$Pos <: Integer$$
$$Neg <: Integer$$

But also, we could have:

$$Pos <: Pos, \quad Neg <: Neg, \quad Integer <: Integer$$

(But we chose not to for simplicity here).

Part 4

$$\frac{\Gamma \vdash e : T_1 \quad T_1 <: T_2}{\Gamma \vdash e : T_2}$$

One could change some rules. For instance, the integer literal rules could be:

$$\frac{}{\Gamma \vdash 0 : \text{Integer}} \qquad \frac{c_e > 0}{\Gamma \vdash c_e : \text{Pos}} \qquad \frac{c_e < 0}{\Gamma \vdash c_e : \text{Neg}}$$

Part 5

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash 7 : \text{Pos}} \; \text{Pos} <: \text{Integer}}{\Gamma \vdash 7 : \text{Integer}} \quad \dfrac{}{\Gamma \vdash 2 : \text{Pos}}}{\Gamma \vdash 7/2 : \text{Integer}} \quad \dfrac{\dfrac{}{\Gamma \vdash 3 : \text{Pos}} \; \text{Pos} <: \text{Integer}}{\Gamma \vdash 3 : \text{Integer}}}{\Gamma \vdash \text{power}(7/2, 3) : \text{Integer}}$$

is one possible type derivation.

If we were to add reflexive subtyping rules, such as $\text{Pos} <: \text{Pos}$, we could generate more derivations.