Ex. 1: Convert Haskell types → MinHS types, give example values:

① `data Direction = East | West | North | South` ≅ $1 + (1 + (1 + 1))$

In MinHS, North is $\text{InR}(\text{InR}(\text{InL}\ ()))$

② `data Foo = Foo Int Bool Int` ≅ ~~$1 \times$~~ $\text{Int} \times (\text{Bool} \times \text{Int})$

In MinHS, `Foo 1 True 2` is $(1, (\text{True}, 2))$

③ `data Tree = Node Tree Tree | Leaf` ≅ $\text{rec}\ t.\ (t \times t) + 1$

In MinHS, Node Leaf Leaf is $\text{roll}(\text{InL}(\text{roll}(\text{InR}\ ()), \text{roll}(\text{InR}\ ())))$

Proof: We have $\text{Leaf} = \text{roll}(\text{InR}\ ())$

Choose T_1 in InR intro

$\text{InR}\ () : (\text{rec}\ t.\ (t \times t) + 1,\ \text{rec}\ t.\ (t \times t) + 1) + 1$

$\text{roll}(\text{InR}\ ()) : \text{rec}\ t.\ (t \times t) + 1$

$(\text{roll}, \text{InR}\ (), \ldots) : (\text{rec}\ t.\ (t \times t) + 1) \times \ldots$

$\text{InL}(\ \_\_\ ) : (\text{rec}\ t.\ (t \times t) + 1 \times \ldots) + 1$

$\text{roll}(\ \_\_\_\_\ ) : \text{rec}\ t.\ (t \times t) + 1.$

Choose as T_2 in InL intro.

Ex. 2: Min HS value of these types:

① $\text{rec } t.\ (\text{Int} + t)$

② $\text{rec } t.\ (\text{Int} \times \text{Bool})$

③ $\text{rec } t.\ (\text{Int} \times t)$ is infinite, but $\text{recfun } f\ x = \text{roll}\ (x, f\ x)$

Ex. 3: Find the type isom. classes: assume strict semantics

$\tau_A = \text{rec } t.\ (t \times t)$ ⟶ can't construct strictly

$\tau_B = 0$

$\tau_C = 1$

$\tau_D = 1 \times 0$

$\tau_E = \text{rec } t.\ (t \times 1)$ ⟶ can't construct strictly

$\tau_F = \text{rec } t.\ (t + 1)$ ⟶ is "isom." to $\mathbb{N}$, i.e. countably infinite

Ex. 4: Give a type for these terms:

① $\text{InL True} : \text{Bool} + \alpha$ for all α

② $\text{InR True} :$

③ $\text{InL (InL True)} :$

④ $\text{roll (InR True)} :$

⑤ $((), \text{InL}\ ()) :$

Ex. 5: Is there a MinHS function with this type?

① $(a \to b) \to (b \to c) \to (a \to c) \xrightarrow{C.H.} (A \Rightarrow B) \Rightarrow (B \Rightarrow C) \Rightarrow (A \Rightarrow C)$

Ans: Yes

Why? The logical prop^n is just transitivity of $\Rightarrow$

② $((a \times b) \to c) \to (a \to c) \xrightarrow{C.H.} ((A \wedge B) \Rightarrow C) \Rightarrow (A \Rightarrow C)$

Ans: NO

Why? If $A = T$, $B = C = \bot$, then we get $T \Rightarrow \bot$ which is false, i.e. the logical prop^n is not a tautology. So not provable.

③ $(a \to c) \to ((a \times b) \to c) \xrightarrow{C.H.} (A \Rightarrow C) \Rightarrow ((A \wedge B) \Rightarrow C)$

Ans: Yes

Why? We can just impl. The function. Not sure about a pure logic answer here.

Ex. 6: Write a MinHS program which proves $(A \vee B) \Rightarrow (B \vee A)$

Ex. 8: Difference between two polymorphic functions:

$\forall a.\ a \to a$ vs. $(\forall a.a) \to (\forall a.a)$

Ex. 9: Deductions based on type signature:

① $f_1 : \forall a.\ [a] \to a$

② $f_2 : \forall a\ b.\ [a] \to [b]$

③ $f_3 : \forall a\ b.\ [a] \to b$

④ $f_4 : [Int] \to [Int]$