1 Warmup

- Is the program \( x = 5 \) in the grammar?
  If so, does it typecheck?

- Is the program \((x : \text{Int}) \Rightarrow (x = 5)\) in the grammar?
  If so, does it typecheck?
2 Call-By-Wha?!?!

- Walk through the evaluation steps for \( ((\text{var} \ x : \ 	ext{Ptr[Int]}) \Rightarrow (x = \text{new}(3); \ *x))(\text{new}(5)) \). How many addresses have been allocated and assignment by the end of evaluation? Why?

- Walk through the evaluation steps for \( ((\text{ref} \ x : \ 	ext{Int}) \Rightarrow (x = 3))(\text{new}(5)) \). How many addresses have been allocated and assignment by the end of evaluation? Why?

- Write a function using the \text{var} annotation that takes a pointer to some integer as a parameter and reassigns the pointer to the integer value 3 (in other words, rewrite the previous program using \text{var} instead of \text{ref}). Apply this function to a \text{Ptr[Int]} and walk through the evaluation steps.

- A potential problem with this language is that using the construct \text{if} \ (e_1) \ e_2 \ \text{else} \ e_3 \ in \ a \ function \ can \ behave \ unexpectedly \ if \ we \ choose \ the \ wrong \ kind \ of \ function \ parameter \ annotation. \ For \ example, \ say \ we \ wrote \ the \ following \ program:

\[
(((\text{tst} : \ 	ext{Boolean}) \Rightarrow
((\text{s0} : \ 	ext{Int}) \Rightarrow ((\text{s1} : \ 	ext{Int}) \Rightarrow \text{if} \ (b) \ \text{s0} \ \text{else} \ \text{s1}))) \ \text{true}))(\text{print}(5); \ 7)) \ (\text{print}(6); \ 8))
\]

What we would like is to have the program print only 5 and evaluate to 7, but currently it is printing both 5 and 6 before evaluating to 7. What is the problem? How can we fix it?