Interpret functions:

Create a closure: a) formal parameters

b) function body

c) A function that creates the state/environment that is active for the function call.

When you see a function definition, create a closure for the function and store that closure in the state, bound to the function name.

When you get a function call, you need to get the closure from the state,

1. Create the state for the function call, by calling the function in the closure.
2. Add a layer to that, evaluate each of the actual parameters (in the current state), bind them to the formal parameters, and place in the function’s state.
3. Execute the function body, setting a new return continuation.
4. (If not using boxes) when the function returns you need to update the current state with the changes made to the function’s state.

Memory Management

Memory errors:

1. Memory leak
   1. When you allocated memory, but you lose the reference to it, so you can no deallocate it.
2. Dangling pointer
   1. You deallocate memory, but you have a reference to the memory, and then it is an error to try to dereference that pointer.
3. Segmentation violation
   1. Accessing memory that your process is not permitted to access.
4. Stack smash
   1. A stack allocated buffer, but you write past the buffer overwriting parts of the stack.