

Names + referring — We give names to functions so we can use it more than once. ( $\rightarrow$  Lambda + definite integrals).

$\hookrightarrow$  Recursion is a/b solving equations.  
Don't need the quadratic equation,  
you just need the fixed-point.

PA#1: Problem 3 — Think of random choice as outputting a procedure (strategy).  
Higher-order procedures. ~~puts~~

Inputs a ship-state (randomly chooses 1 of 2 strategies), a procedure, and outputs another procedure/strategy.

random-choice: (strategy \* strategy)  
 $\downarrow$   
strategy

Problem #4

high-choice: (strategy \* strategy \* int)  $\rightarrow$  strategy

#5 choice: (strategy \* strategy \* shipstate  $\rightarrow$  <sup>boolean</sup>)

→ strategy

Curry

\* Programs are ~~types~~ proofs, & types are theorems.

$f: A \rightarrow B$  }  $f$  is a procedure turning input of type  $A$  into output of type  $B$   
(or)  $f$  is a proof of theorem  $A \rightarrow B$

$x: A$

... Modus ponens

$A \rightarrow A : (\lambda(x)x)$  } the identity fn

\* Lambda expressions evaluate to procedures \*