Derivation Exercise

it's your turn

How to execute the Racket factorial program given these parts?

- ○ factorial-in-Racket program
- ○ Racket-to-Python-translator-in-Python program
- ○ Python-interpreter-in-C program
- ○ C-to-x86-translator-in-x86 program
- ○ x86 computer (i.e., x86 interpreter machine)

Warning: cannot start the following way:

factorial machine (I)
- ❑ factorial-in-Racket program
- ❑ Racket interpreter machine (I)
  ....

Why not?
The derivation would need to begin:

factorial machine (I)
- ❑ factorial-in-Racket program
- ❑ Racket interpreter machine (I)
  - ○ Racket-interpreter-in-L program
    ...
  - ○ L interpreter machine
    ...

But the parts don't include Racket-interpreter-in-L program **for any L!**

What to do? Explore translating the factorial-in-Racket program **to a** factorial-in-L program **for some L for which we *can* make an interpreter machine!**

Derivation Exercise: Solution

it's your turn

How to execute the Racket factorial program given these parts?

- ○ factorial-in-Racket program
- ○ Racket-to-Python-translator-in-Python program
- ○ Python-interpreter-in-C program
- ○ C-to-x86-translator-in-x86 program
- ○ x86 computer (i.e., x86 interpreter machine)

SOLUTION:

factorial machine (I)
- ❑ factorial-in-Python program (T)
  - ✧ factorial-in-Racket program
  - ✧ Racket-to-Python translation machine (I)
    - ➢ Racket-to-Python-translator-in-Python program
    - ➢ Python interpreter machine (I)
      - ◆ Python-interpreter-in-x86 program (T)
        - ○ Python-interpreter-in-C program
        - ○ C-to-x86-translator machine (I)
          - • C-to-x86-translator-in-x86 program
          - • x86 computer (= x86 interpreter machine
      - ◆ x86 computer (= x86 interpreter machine)
- ❑ Python interpreter machine (I)
  # Derivation already given above; no need to rederive it!
  # A reused derivation is a lemma, which corresponds to
  # a helper function in programming