| HW2: Exercises 2.3, 2.4, 2.5, 2.6, 2.8, and 2.9 from Watt & Brown, due on January 31. |
| Bootstrapping (from ch. 13 of Magrjos's book) |
Suppose we want ML \times 86

Suppose we have ML \xrightarrow{\times 86} ARM

Option 1: do binary translation, i.e. build ARM \times 86 ML
We can then translate the compiler we have using the binary translator.

\[ \text{P} \quad \text{ML} \quad \text{ML} \quad \text{ARM} \quad \text{ML} \quad \text{ARM} \quad \text{ARM} \quad \text{ARM} \quad \text{x86} \quad \text{x86} \quad \text{ML} \]

\[ \text{P} \quad \text{ML} \quad \text{ARM} \quad \text{x86} \quad \text{x86} \quad \text{ML} \]

\[ \text{P} \quad \text{ML} \quad \text{ARM} \quad \text{x86} \quad \text{x86} \quad \text{ML} \]

But: requires an extra pass.

And the ARM-x86 needs to run on the x86 mode.
So, instead of binary translation, it usually better to solve this problem by half-bootstrapping.

We want \( \text{ML} \times \text{ARM} \) \( \Rightarrow \) \( \text{P} \rightarrow \text{x86} \)

We have: \( \text{ML} \rightarrow \text{ARM} \)
We write $\text{ML} \xrightarrow{\text{X86}} \text{ML}$, compile it:

$\text{ML X86}$ $\text{ARM}$ $\text{ML X86}$

$\text{ML}$ $\text{ARM}$ $\text{ARM}$

$\text{ARM}$ $\text{ARM}$

$\text{ARM}$ $\text{ARM}$

A cross-compiled which are used to compile the "real" compiler written in ML,
You can then compile again, using the native compiler for testing purposes. If we do not get the same object code, we have an error.
Now: full-bootstrap. In half-bootstrap, we had compiler for the desired language but running on a different machine.

We want a compiler for the new language M+.
We write the "real" compiler

We write a QAD ("Quick and Dirty")

translator

We compile the

QAD compiler.

M+ ARM

M+

M+ ML

M+ ML

M+ ML

M+ ML

M+ ML

ML

ML

ML

ML

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM

ARM
M+ ARM  
M+ M+ ML ML
  ARM ARM

This is functionally what we want.
Recompile for efficiency.
Alternatively, one could do full bootstrapping using an interpreter.

Want: \[ \text{ML} \rightarrow \text{ARM} \]
\[ \rightarrow \text{ARM} \]

Have: \[ \text{ML} \rightarrow \text{ARM} \]
\[ \rightarrow \text{ARM} \]
Write the "real" compiler

Write a QAD interpreter

Compile it:
M+ ARM
M+ M+ ARM ARM
M+ ARM

Done!

Since the "real" compiler was used here, no need to recompile for efficiency.

Sometimes, M+ is easier than M+ (viz. harder) ML