Compiler Construction

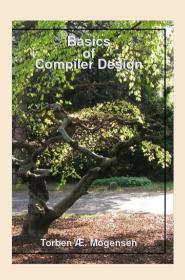
Lecture 16



Bootstrapping

Jeremy Yallop, Lent 2024 jeremy.yallop@cl.cam.ac.uk

Recommended book



Chapter 13 of
Basics of Compiler Design
Torben Ægidius Mogensen
http://hjemmesider.diku.dk/~torbenm/Basics/

Notation

Notation: programs, interpreters, machines

Notation



Examples

Compiling compilers

Full bootstrap A program

f

An interpreter

L2 L1 A machine

L

Computes function f written in language L

Interprets language L2 written in language L1

Executes code in language L

Notation: compilers

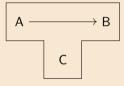
Notation



Examples

Compiling compilers

Full bootstrap A compiler



Translates language A into language B Written in language C

Examples

Executing programs

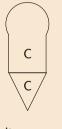
Notation

Examples

•0000¢

Compiling compilers

Full bootstrap To execute a program



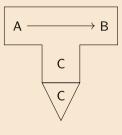
we run it on a machine

To execute an interpreter



we run it on a machine

To execute a compiler



we run it on a machine

Interpreting a program

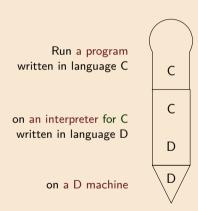
Notation

Examples

•000¢

Compiling compilers

Full bootstrap



(Note: the languages must match)

Interpreting a Java program

Notation

Examples

ullet

Compiling compilers

Full bootstrap Run a program written in Java byte code

on an interpreter for Java byte code written in x86 code

on a x86 machine



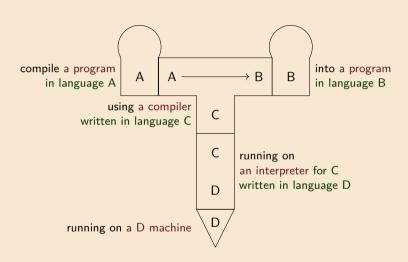
Running a compiler on an interpreter

Notation

Examples



Compiling compilers

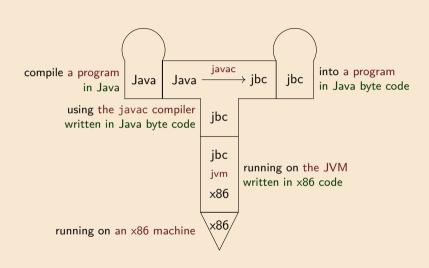


Running javac on the JVM

Notation

Examples

Compiling compilers



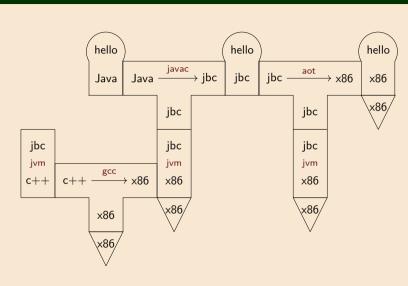
Ahead-of-time compilation for Java

Notation

Examples

Compiling compilers

Full bootstrap



Thanks to David Greaves for the example

Compiling

compilers

Notation

Examples

Compiling compilers



Full bootstrap

The OCaml compiler is written in OCaml



Puzzle: how was the compiler compiled?

Translating translators

Notation

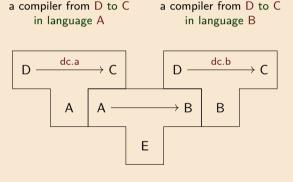
Compilers can be translated, just like any other program:

Examples

Compiling compilers



Full bootstrap



compile programs from A to B

that runs on x86

Notation

We have: a compiler from ML to arm that runs on arm

ML o arm

We want: a compiler from ML to x86

 $\begin{array}{c}
 \text{ML} \rightarrow \times 86 \\
 \times 86
\end{array}$

Examples

Compiling compilers



Notation

We have: a compiler from ML to arm

that runs on arm

 $ML \rightarrow arm$ arm

We want: a compiler from ML to x86

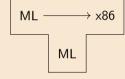
 $ML \rightarrow x86$ x86 that runs on x86

Examples

1. write an ML-to-x86 compiler in ML

Compiling compilers





Notation

We have: a compiler from ML to arm that runs on arm

 $ML \rightarrow arm$ arm

We want: a compiler from ML to x86

 $ML \rightarrow x86$ x86 that runs on x86

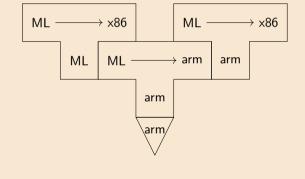
Examples

1. write an ML-to-x86 compiler in ML

Compiling compilers

...

Full bootstrap 2. compile the compiler for arm



that runs on x86

Notation

We have: a compiler from ML to arm that runs on arm ML → arm

We want: a compiler from ML to x86

 $\begin{array}{c} \text{ML} \rightarrow \times 86 \\ \hline \times 86 \end{array}$

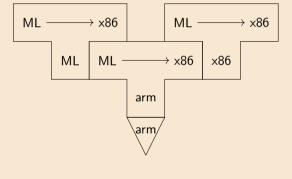
Examples

Compiling compilers



Full bootstrap 1. write an ML-to-x86 compiler in ML

- 2. compile the compiler for arm
- 3. run the compiler on arm to compile itself



Half and full bootstraps

Notation

Previous example: *half bootstrap* (needs existing compiler for the language).

New example: full bootstrap (no existing ML compiler for the language)

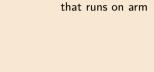
Examples

Compiling compilers





Notation



a compiler from ML to arm

We have:

arm

 $\mathsf{ML} o \mathsf{arm}$

a compiler from **XL** to arm that runs on arm

We want:



 $\mathsf{XL} o \mathsf{arm}$





 $\bullet \bullet \circ \circ$





Notation

 $\begin{tabular}{ll} \begin{tabular}{ll} \beg$

 $\mathsf{ML} o \mathsf{arm}$

We want: a compiler from XL to arm that runs on arm

 $\mathsf{XL} o \mathsf{arm}$

Examples

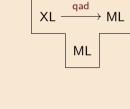
XL-to-ML compiler in ML

1. write a quick-and-dirty (QAD)











We have: a compiler from ML to arm that runs on arm

 $\mathsf{ML} o \mathsf{arm}$

We want: a compiler from XL to arm that runs on arm

arm

 \rightarrow ML

qad

XL

 $XL \rightarrow arm$

Examples

Compiling compilers

XL-to-ML compiler in ML

2. compile the QAD compiler for arm

1. write a quick-and-dirty (QAD)

Z. compile

r for arm

\arm

qad

ML

XL



 $XL \rightarrow arm$

arm

We want:

that runs on arm

a compiler from XL to arm

real

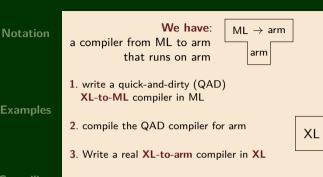
XL

→ arm



Full bootstrap

 \bullet \bullet \circ



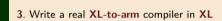
 $XL \rightarrow arm$

arm

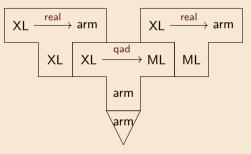


Notation

a compiler from ML to arm that runs on arm 1. write a quick-and-dirty (QAD) XL-to-ML compiler in ML 2. compile the QAD compiler for arm



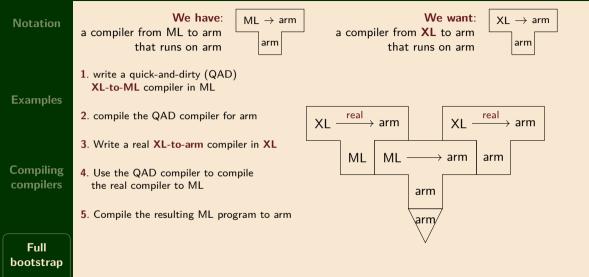
4. Use the QAD compiler to compile the real compiler to ML



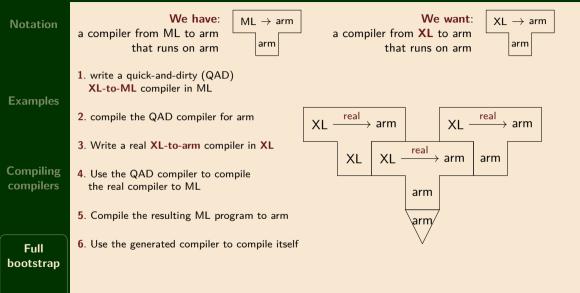
We want:

that runs on arm

a compiler from XL to arm



 $\bullet \bullet \circ \circ$



Observations

Notation

Examples

Compiling compilers

Full bootstrap The *speed* of the quick-and-dirty compiler does not matter much (We could even use a quick-and-dirty interpreter instead)

We don't need to give the quick-and-dirty compiler to users

Once the real compiler works, we can discard the quick-and-dirty compiler altogether

