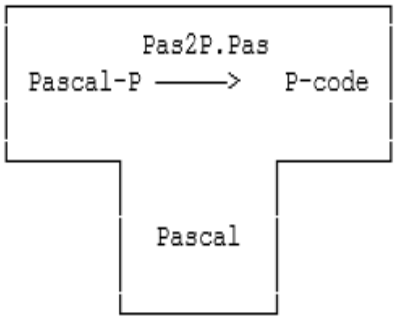
Programming Language Translation Lecture 5

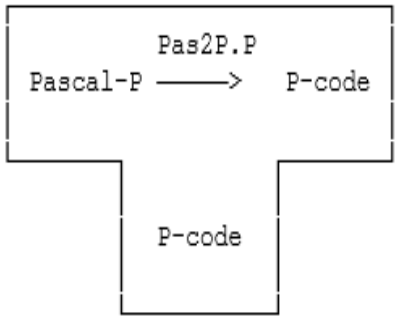
Karen Bradshaw

Chapter 2 (pp. 21–24) and Chapter 3

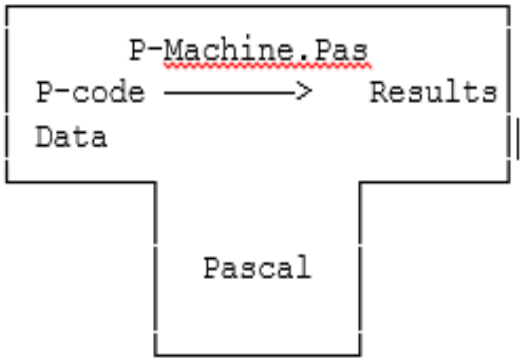
The Zürich P-System kit

- Source code of a Pascal to P-code compiler, written in Pascal
- Object code version of the same Pascal compiler, in P-code



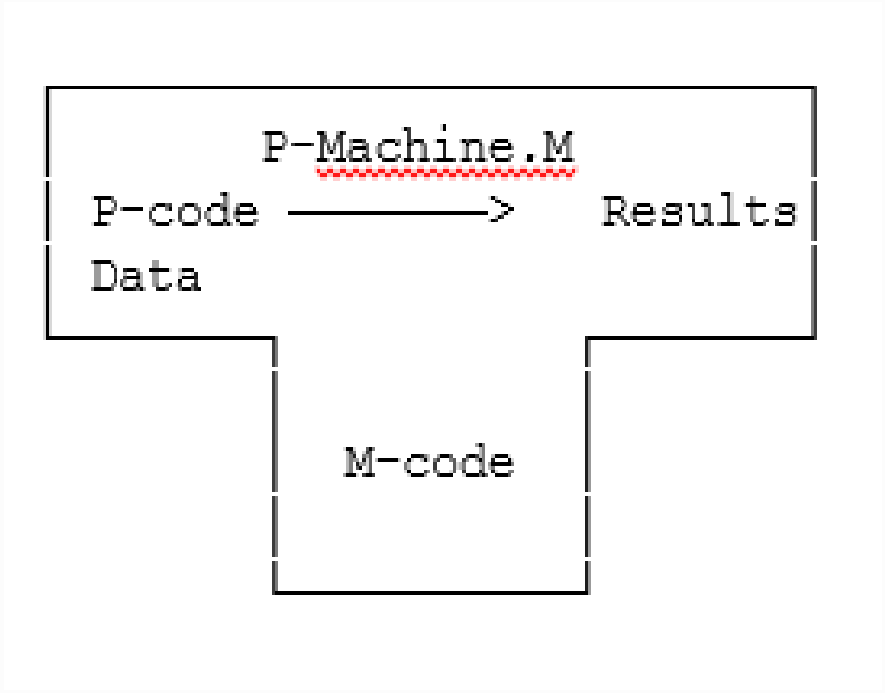


- Source code for a P-Machine emulator, written in Pascal

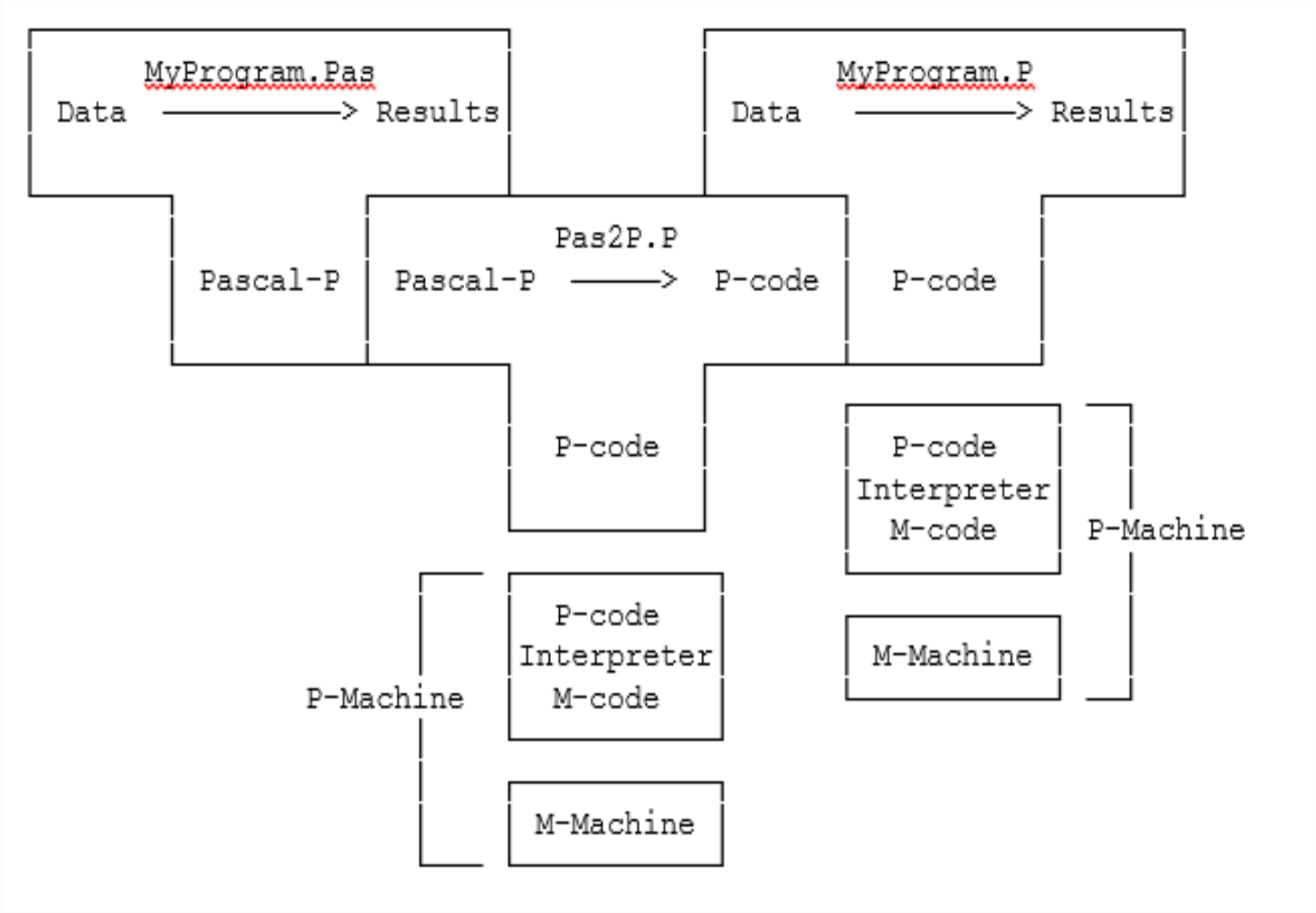


The Zürich P-System kit (2)

- To use the kit it was necessary to develop a native-code version of the P-Machine emulator using some locally available host language (Fortran, Assembler …)



Compilation and execution using the P-system

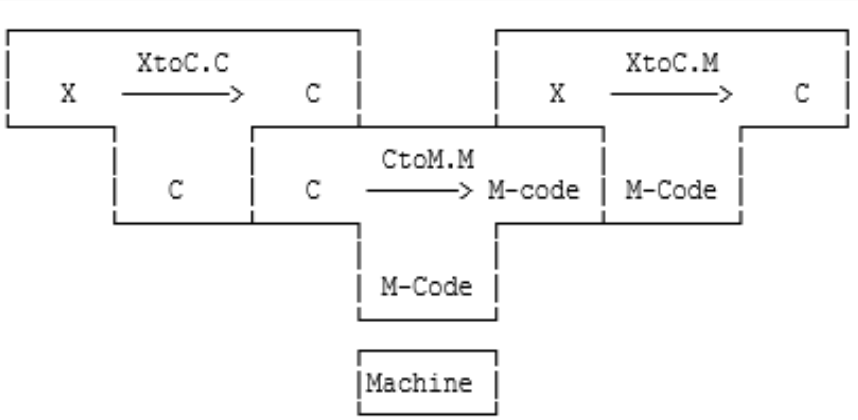


The big question

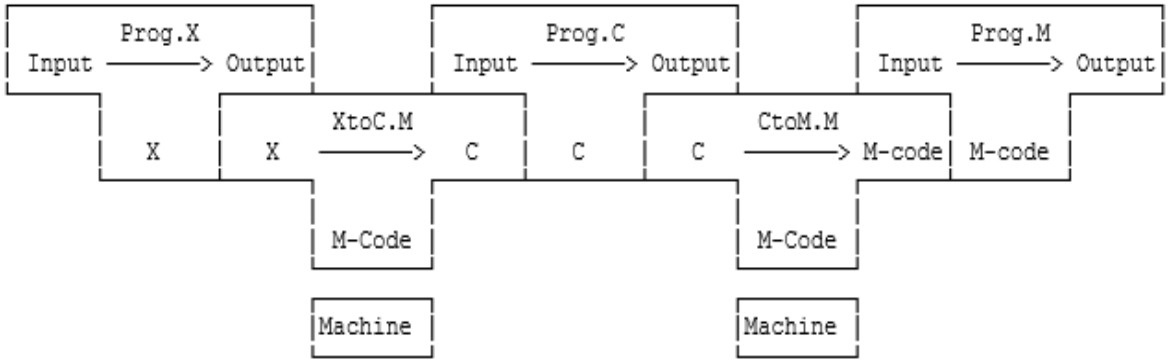
HOW DO WE DEVELOP THE FIRST COMPILER FOR A NEW PROGRAMMING LANGUAGE?

Porting and using a high-level translator

- Porting a high-level (X to C) compiler to a new machine using an existing C compiler on the new machine

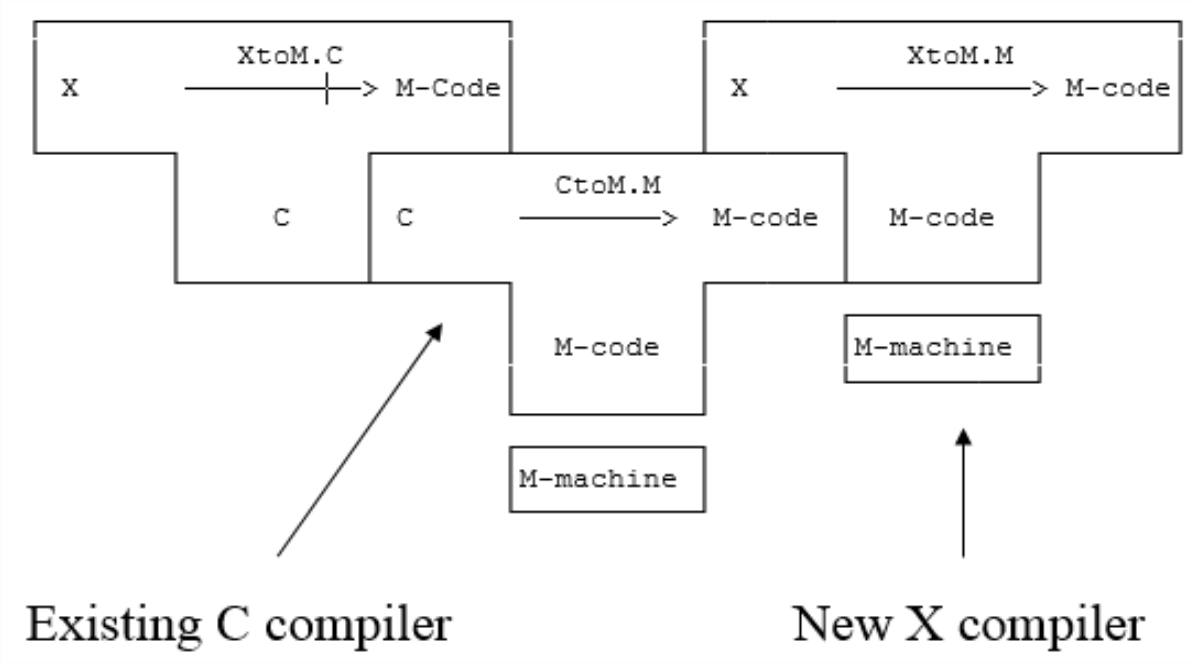


- Using the high-level compiler as the first stage of a two-stage compiler, with the C compiler providing the final stage



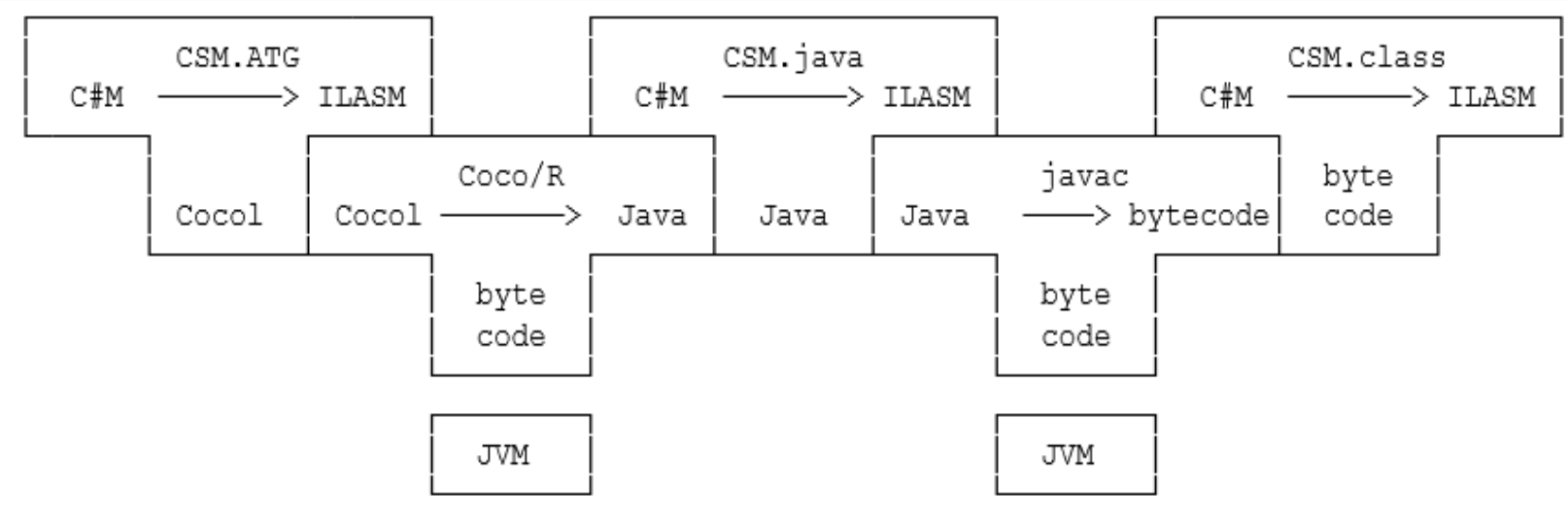
Use of C as implementation language

- Most computers are provided with a C compiler, which can be used to develop further compilers by developing those compilers using C as the host language:



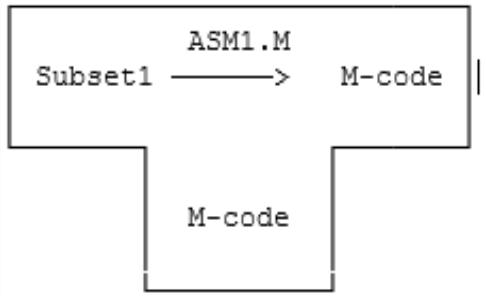
Development with the aid of a compiler generator

- Modern compilers are often developed using a compiler generator that takes as input a formal description of the language and generates source code for part of the compiler from this:

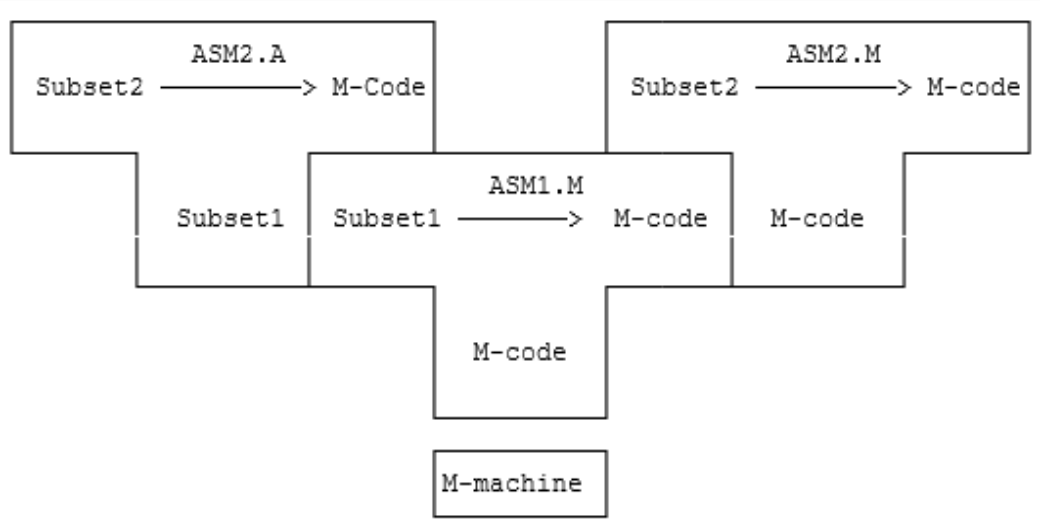


Full bootstrap of an assembler

- The simplest version is developed the hard way:

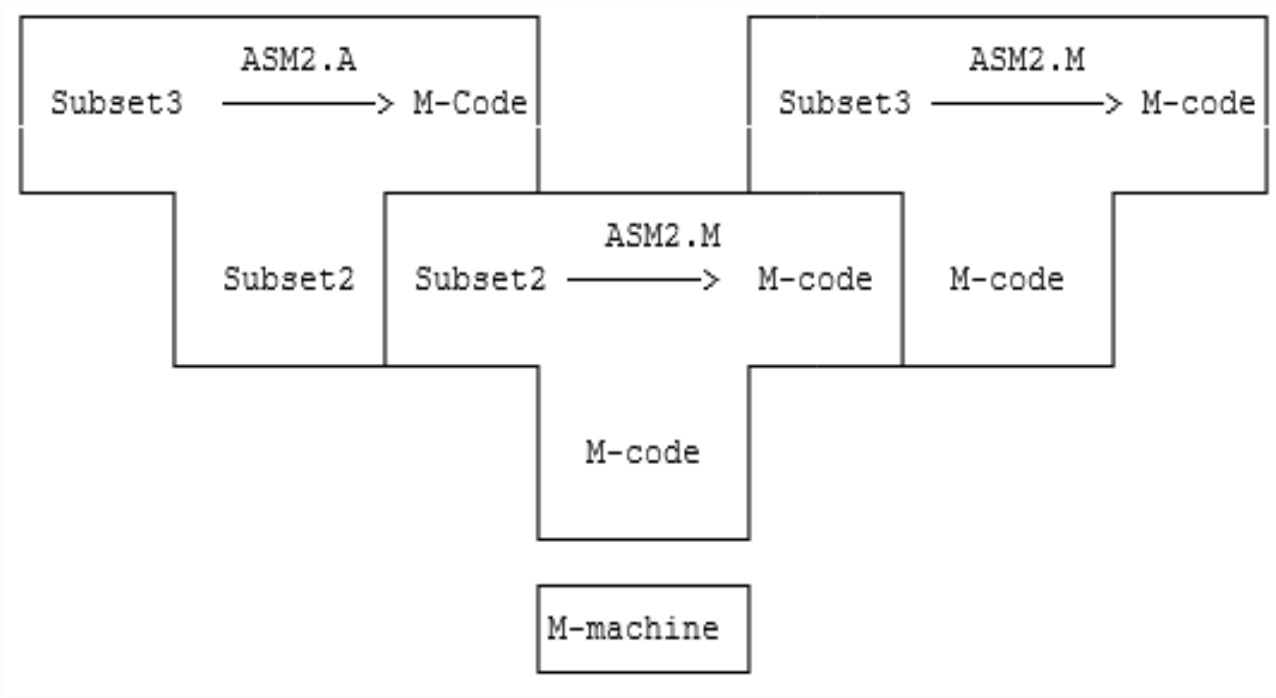


- Once we have a base version we can start to do further development in Assembler, rather than in M-code



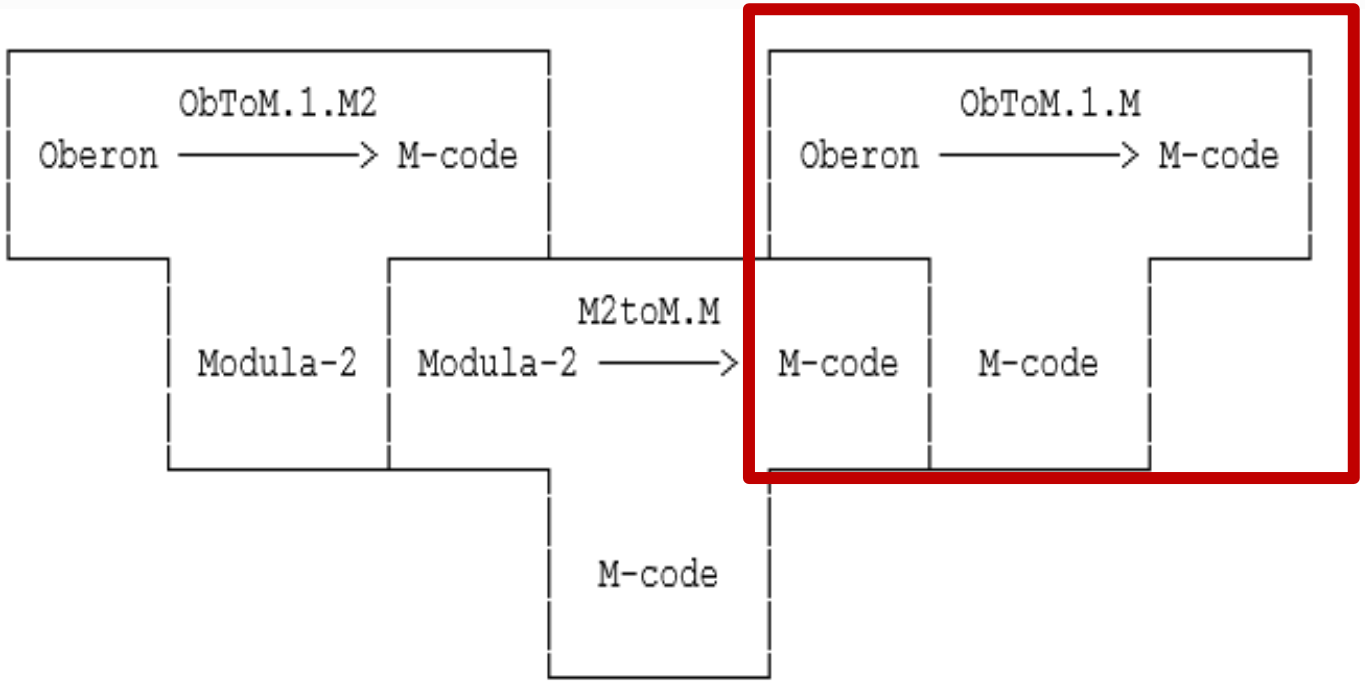
Full bootstrap of an assembler (2)

- Once we have a more powerful version we can produce still more powerful versions:



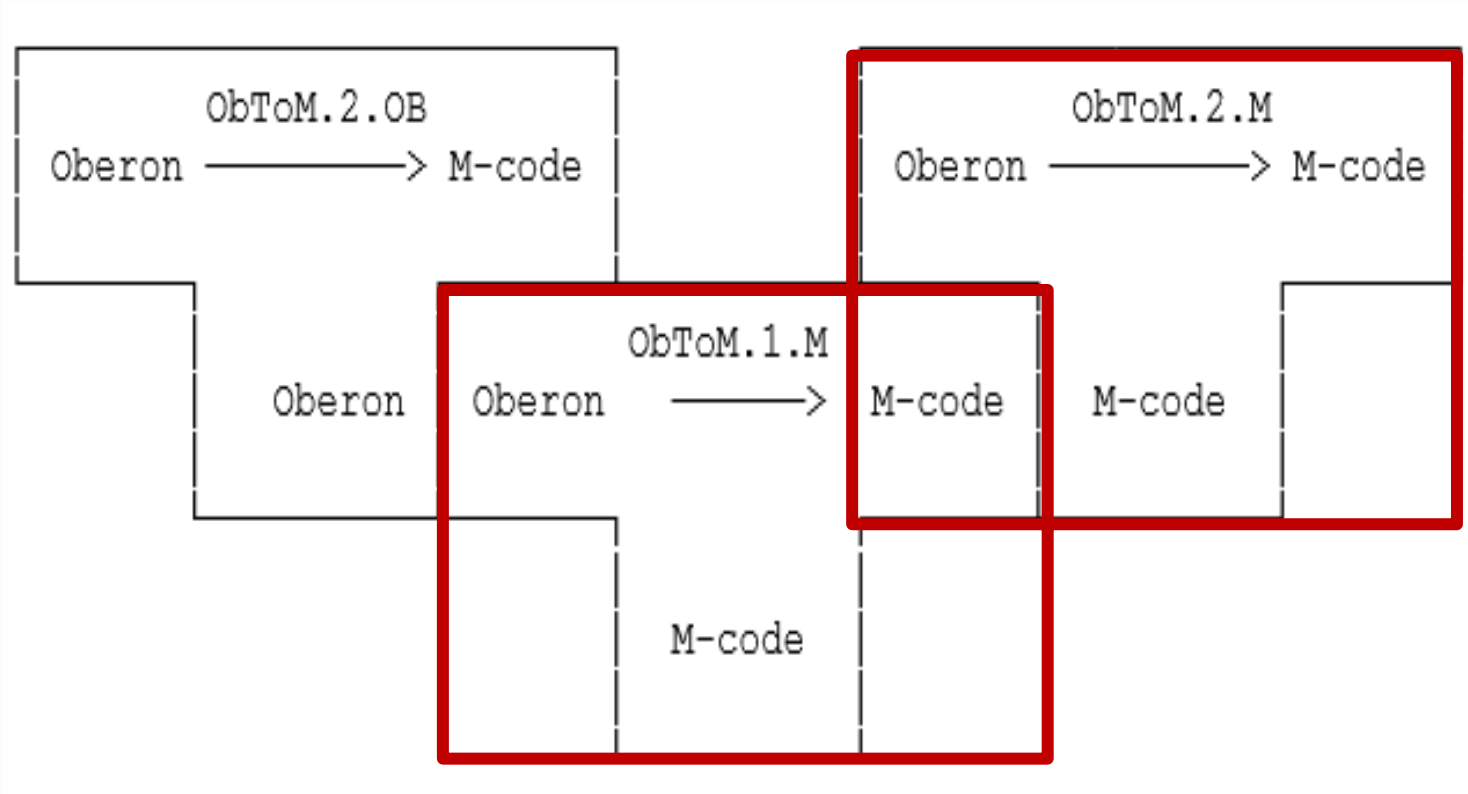
Self-compiling compilers

- The first version of a compiler is usually developed using a compiler for another language. This other language is the original host language. Development of the first version of the compiler can be represented:



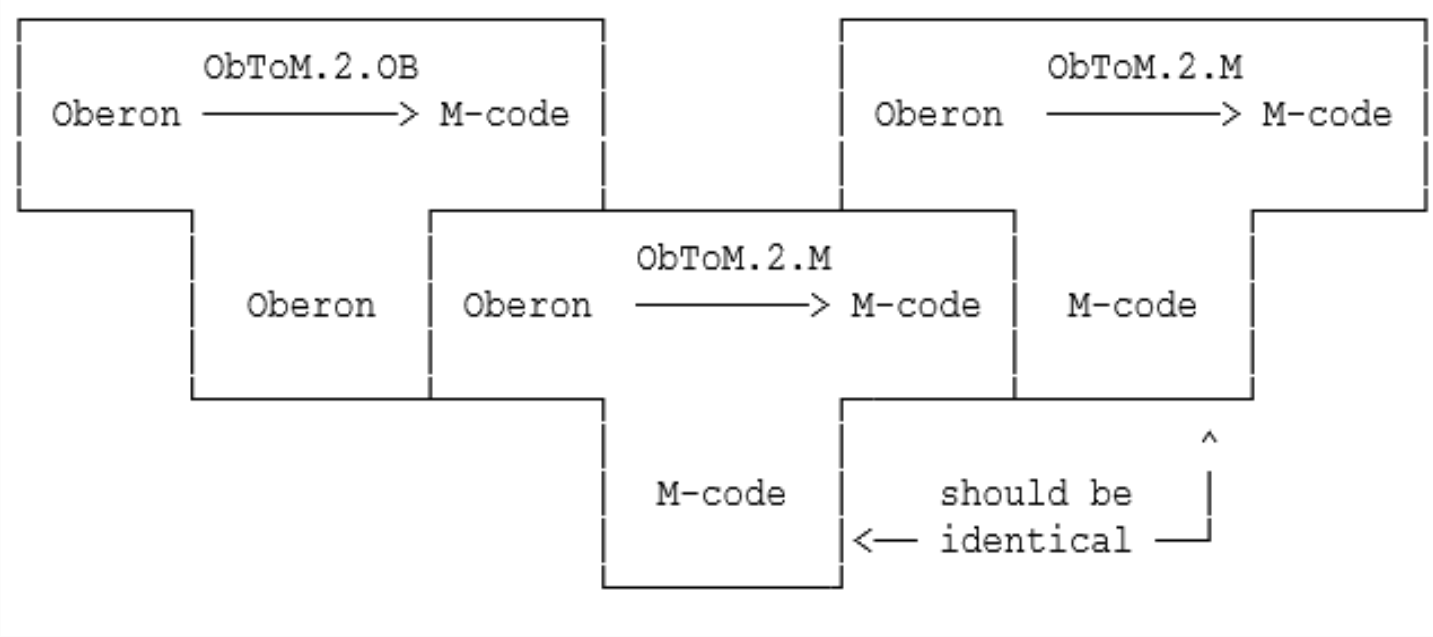
Self-compiling compilers (2)

- We then produce a second source code version of the required compiler, using the source language as the host language, and compile it with the first version:



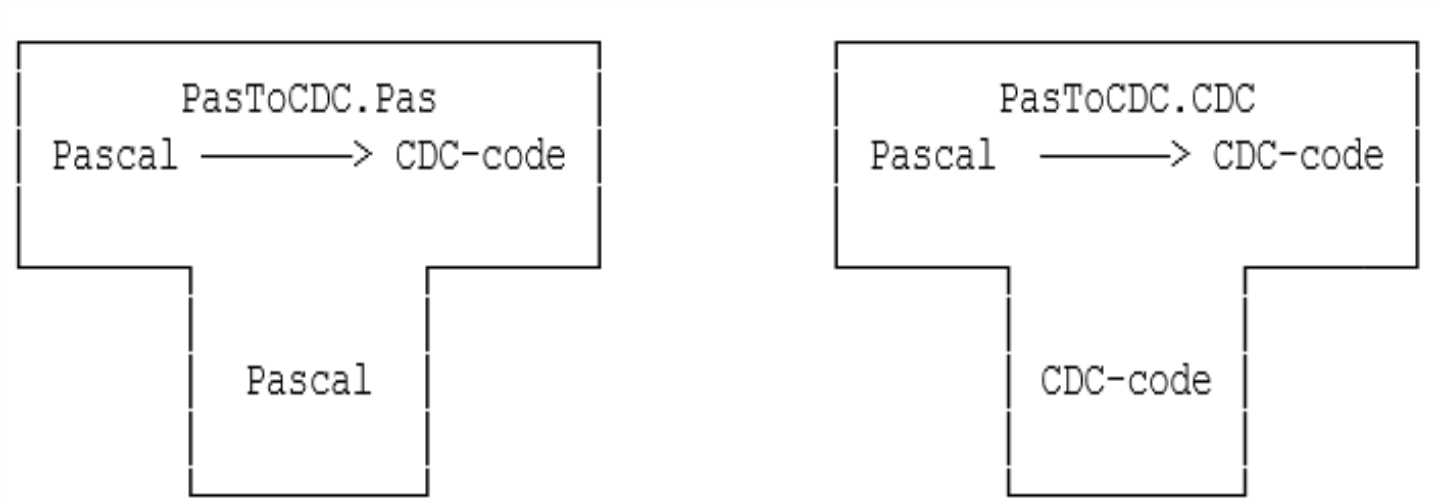
Self-compiling compilers (3)

- If we now use the object version of this compiler to compile its own source code we should find that it reproduces the same object code!



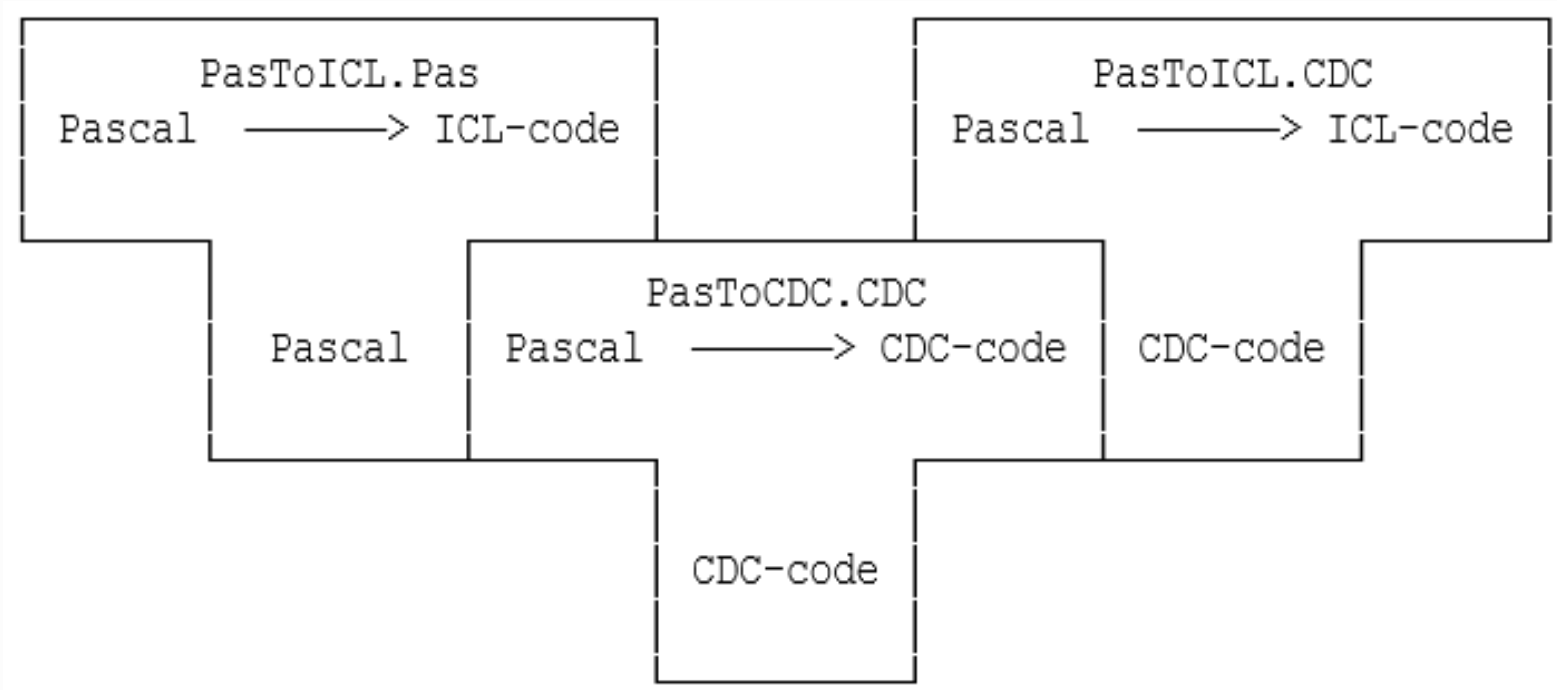
The half bootstrap

- The first Pascal compilers were developed in Zürich on a CDC mainframe. The self-compiling compiler at the end of this process existed in two forms:



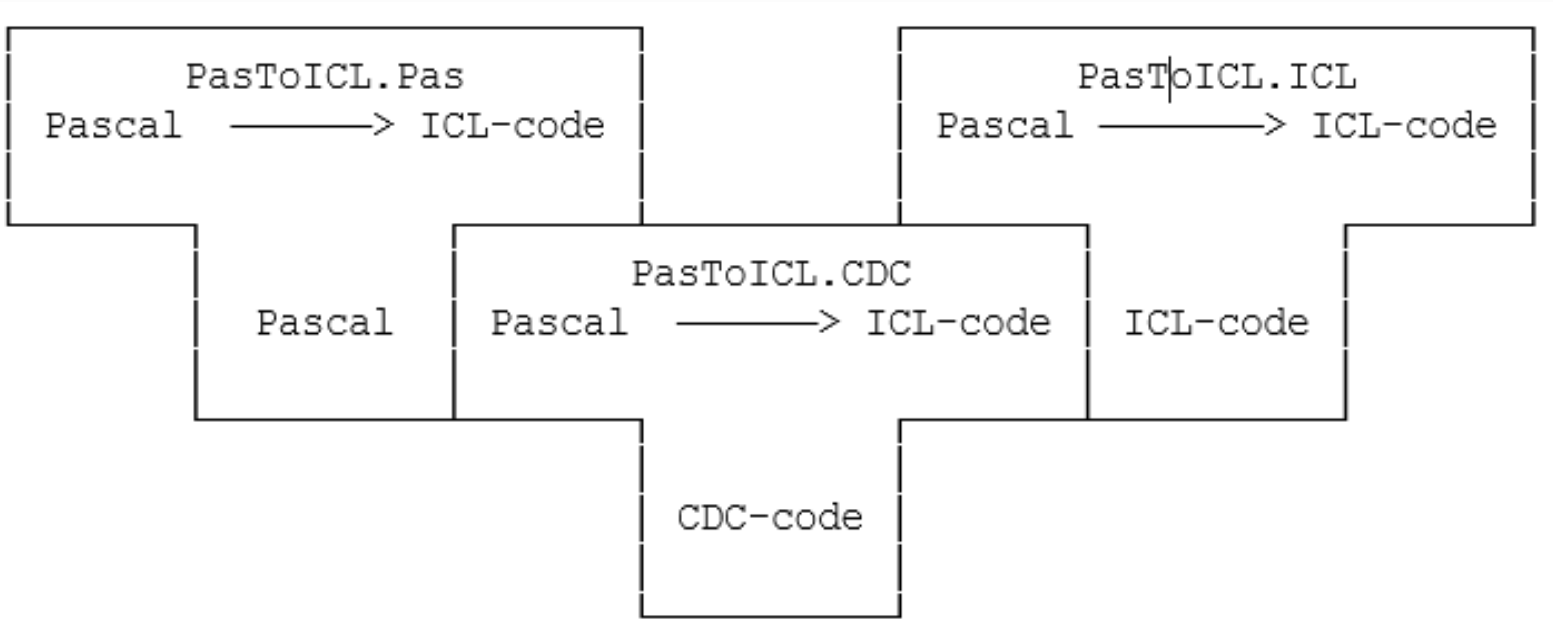
The half bootstrap (2)

- When a Pascal compiler was needed for an ICL mainframe in Belfast the first stage of the bootstrap involved retargeting the back end to produce a cross compiler:



The half bootstrap (3)

- Running the cross compiler of the CDC machine produced the object code version for the ICL machine:



Next lecture …

- Please read Chapter 4, pp. 34 – 38