Exercises 3 (Compilers and interpreters) – Solutions

3A. *(Translators)*

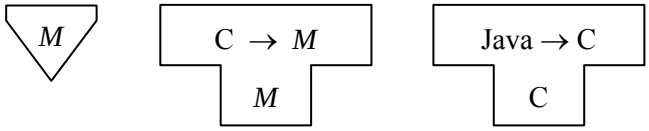
(a) A Java → C translator would be useful, enabling Java programs to be compiled (via C) to real machine code. The hardest problem would be to translate Java's OO features into C code, but that problem should be surmountable.

(b) A C → Java translator would have no useful purpose. It would be practically impossible to translate C's low-level features (such as pointer handling) into Java code.

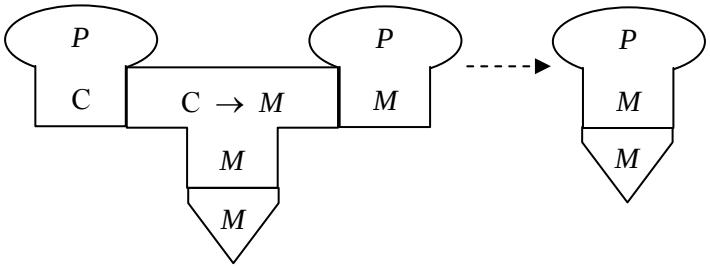
(c) A machine code → C decompiler might be useful in reverse engineering of software whose source code has been lost. However, it would be very difficult to generate readable C code from machine code.

3B. *(Compilers)*

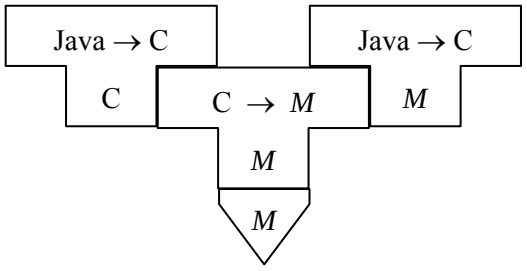
(a) Tombstones representing a machine M; a C compiler that runs on machine *M* and generates machine code *M*; and a Java → C translator expressed in C:



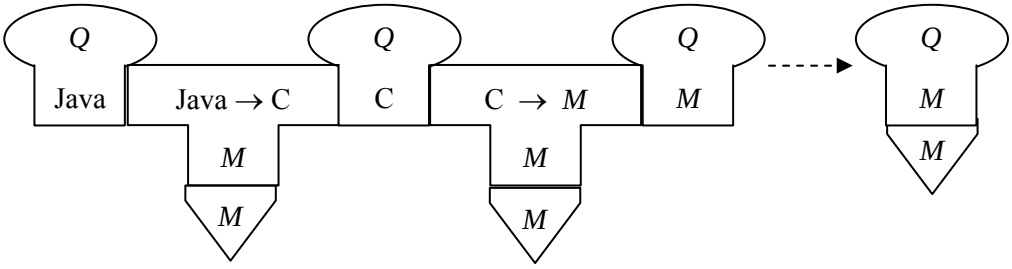
(b) To compile and run a program *P* expressed in C:



(c) To compile the Java → C translator into machine code:

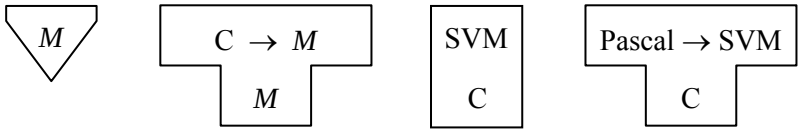


(d) To compile and run a program *Q* expressed in Java:

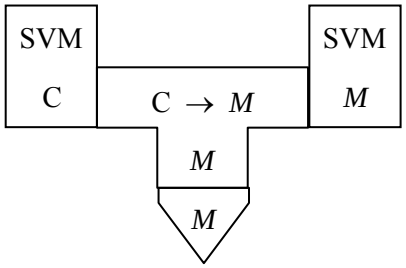


3C. *(Interpretive compiler)*

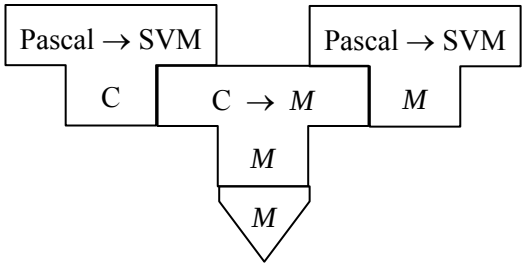
(a) Tombstones representing a machine *M*; a C compiler that runs on machine *M* and generates machine code *M*; an SVM interpreter expressed in C; and a Pascal → SVM compiler expressed in C:



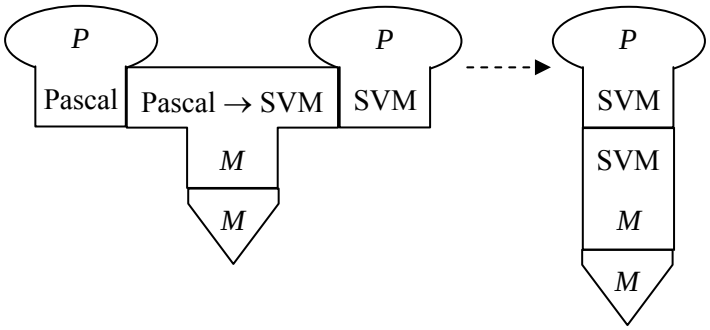
(b) To compile the SVM interpreter into machine code:



(c) To compile the Pascal → SVM compiler into machine code:

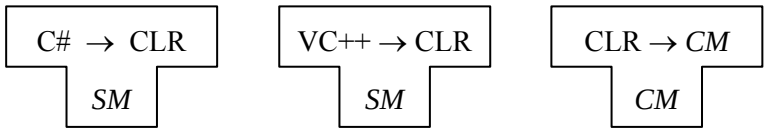


(d) To compile and run a program *P* expressed in Pascal:

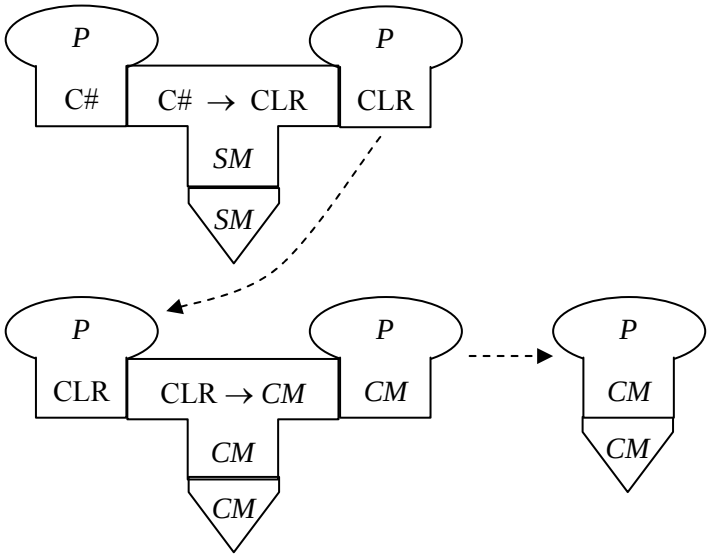


3D. *(CLR)*

(a) Tombstones representing compilers for C# and Visual C++ (VC++), running on a server *SM*; and a JIT compiler running on a client *CM*:



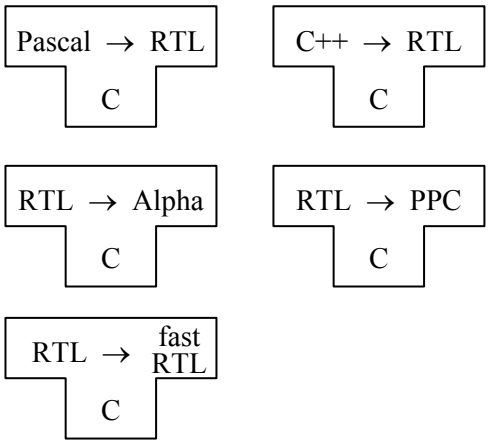
(b) To compile a program *P* expressed in C# and run it on *CM*:



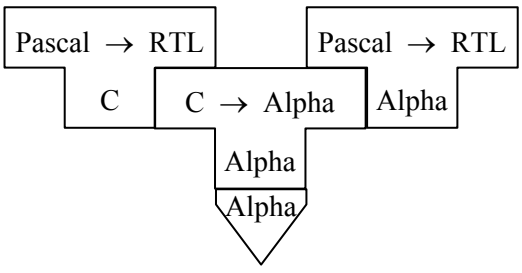
(c) The JIT compiler could be run immediately after downloading the CLR code, or it could be run immediately before execution.

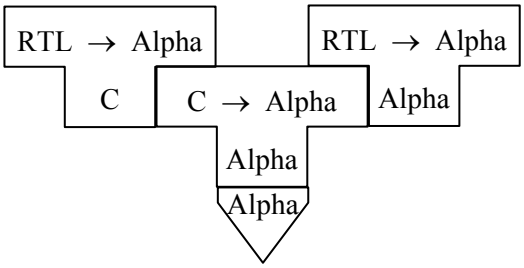
3E. *(Gnu compiler kit)*

(a) Tombstones representing translators from Pascal and C++ into RTL; translators from RTL into Alpha and PPC machine codes; and the RTL optimizer:

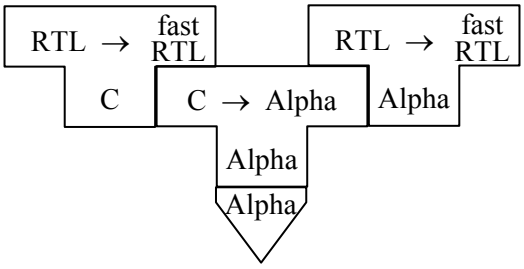


(b) To install these components on an Alpha machine, given a C compiler for the Alpha:

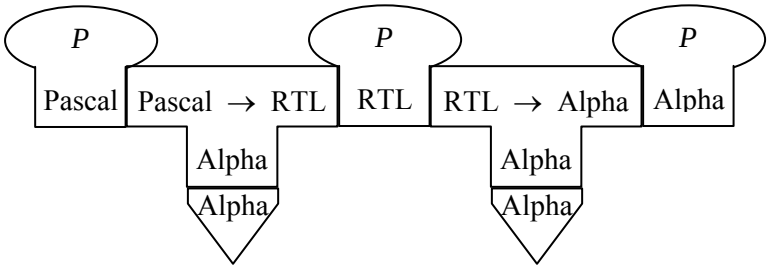




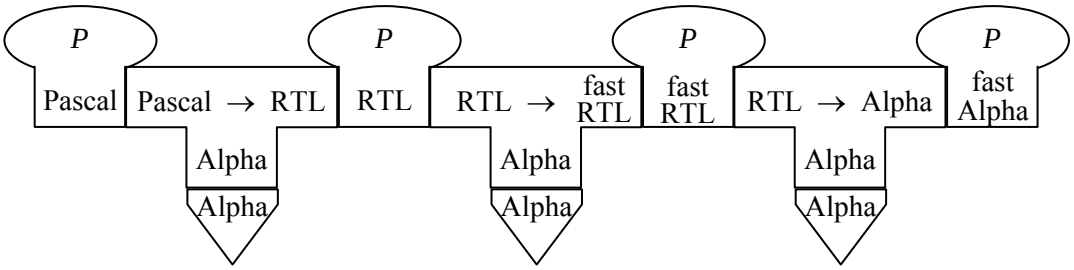
– and similarly for the RTL → PPC translator



(c) To compile a program *P*, expressed in Pascal, into Alpha machine code:



(d) To compile the same program, but using the RTL optimizer to generate more efficient object code:



(e) To cross-compile a program *Q*, expressed in C++, into PPC machine code: