Exercises 3 (Compilers and interpreters)

3A. *(Translators)*

Consider each of the following (hypothetical) translators. Do you think the translator might be useful in practice? Explain your answer. Also, what difficulties could be anticipated in making the translator generate good-quality object code?

(a) a Java → C translator;

(b) a C → Java translator;

(c) a machine code → C decompiler.

3B. *(Compilers)*

Suppose that you are given the following components: a machine *M*; a C compiler that runs on machine *M* and generates machine code *M*; and a Java → C translator expressed in C.

(a) Draw tombstones to represent these components.

Also show how you would use these components to:

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

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

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

3C. *(Interpretive compiler)*

Suppose that you are given the following components: 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.

(a) Draw tombstones to represent these components.

Also show how you would use these components to:

(b) compile the SVM interpreter into machine code;

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

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

3D. *(CLR)*

The Microsoft common language run-time (also called .NET) is based on a general-purpose intermediate language, CLR. Compilers for a variety of programming languages (such as C# and Visual C++) all generate CLR code. A CLR object program can subsequently be downloaded to any client machine where a JIT compiler is available. The JIT compiler translates the CLR object program into native machine code before execution.

Note that CLR code is *never* interpreted.

(a) Draw tombstones to represent the following components: compilers for C# and Visual C++, both running on a server *SM*; and a JIT compiler running on a client *CM*.

(b) Show how you would use these components to compile a program *P* expressed in C# and run it on *CM*. Make a clear distinction between compile-time and run-time.

(c) The CLR code must be translated by the JIT compiler before execution. What are the options?

3E. *(Gnu compiler kit)*

The Gnu compiler kit uses a machine-independent register transfer language, RTL, as an intermediate language. The kit includes translators from several high-level languages (such as Pascal and C++) into RTL, and translators from RTL into several machine codes (such as Alpha and PPC). It also includes an RTL "optimizer", i.e., a program that translates RTL code into more efficient RTL code. All of these translators are expressed in C.

(a) Draw tombstones to represent the above components.

(b) Show how you would install these components on an Alpha machine, given a C compiler for the Alpha.

Now show how you would use these components to:

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

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

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