Table of Contents

Source-to-source compiler

Source-to-source compiler (S2S compiler), transcompiler or transpiler is a type of compiler that takes the source code of a program written in a programming language as its input and produces the equivalent source code in the same or a different programming language.

*Snippet from Wikipedia: **Source-to-source compiler***

> A **source-to-source translator**, **source-to-source compiler** (**S2S compiler**), **transcompiler**, or **transpiler** is a type of translator that takes the source code of a program written in a programming language as its input and produces an equivalent source code in the same or a different programming language. A source-to-source translator converts between programming languages that operate at approximately the same level of abstraction, while a traditional compiler translates from a higher level programming language to a lower level programming language. For example, a source-to-source translator may perform a translation of a program from Python to JavaScript, while a traditional compiler translates from a language like C to assembly or Java to bytecode. An automatic parallelizing compiler will frequently take in a high level language program as an input and then transform the code and annotate it with parallel code annotations (e.g., OpenMP) or language constructs (e.g. Fortran's `forall` statements).
>
> Another purpose of source-to-source-compiling is translating legacy code to use the next version of the underlying programming language or an API that breaks backward compatibility. It will perform automatic code refactoring which is useful when the programs to refactor are outside the control of the original implementer (for example, converting programs from Python 2 to Python 3, or converting programs from an old API to the new API) or when the size of the program makes it impractical or time-consuming to refactor it by hand.
>
> Transcompilers may either keep translated code structure as close to the source code as possible to ease development and debugging of the original source code or may change the structure of the original code so much that the translated code does not look like the source code. There are also debugging utilities that map the transcompiled source code back to the original code; for example, the JavaScript Source Map standard allows mapping of the JavaScript code executed by a web browser back to the original source when the JavaScript code was, for example, minified or produced by a transcompiled-to-JavaScript language.
>
> Examples include Closure Compiler, CoffeeScript, Dart, Haxe, Opal, TypeScript and Emscripten.

kb, programming