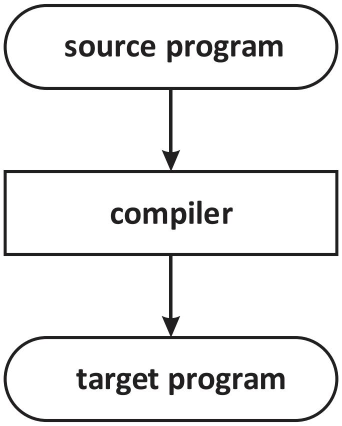


Figure 1. A compiler

compiler a computer program that translates a *source* program into a *target* program
source language the language in which the *source* program is written
target language the language in which the *target* program is written

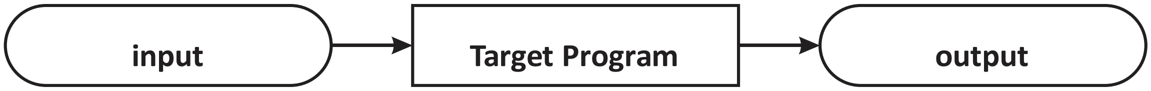


Figure 2. Executing the target program

If the target program in an executable, machine-language program, it can be called by the user to process inputs and produce outputs.

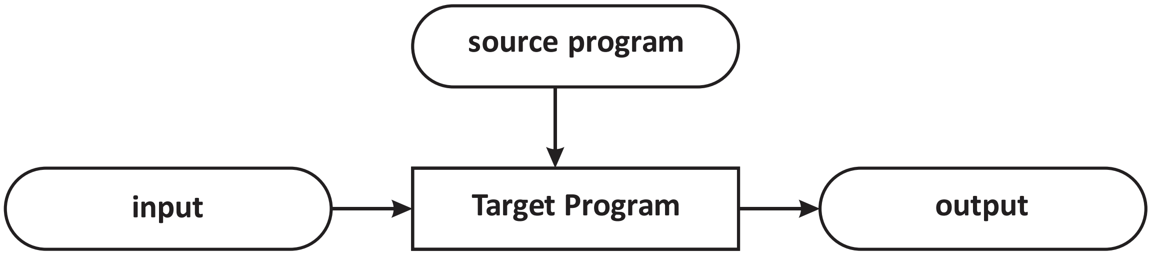


Figure 3. An interpreter

interpreter an interpreter is a combination of a compiler and the target program produced by a compiler. An interpreter reads both the input and the source program to produce the output.

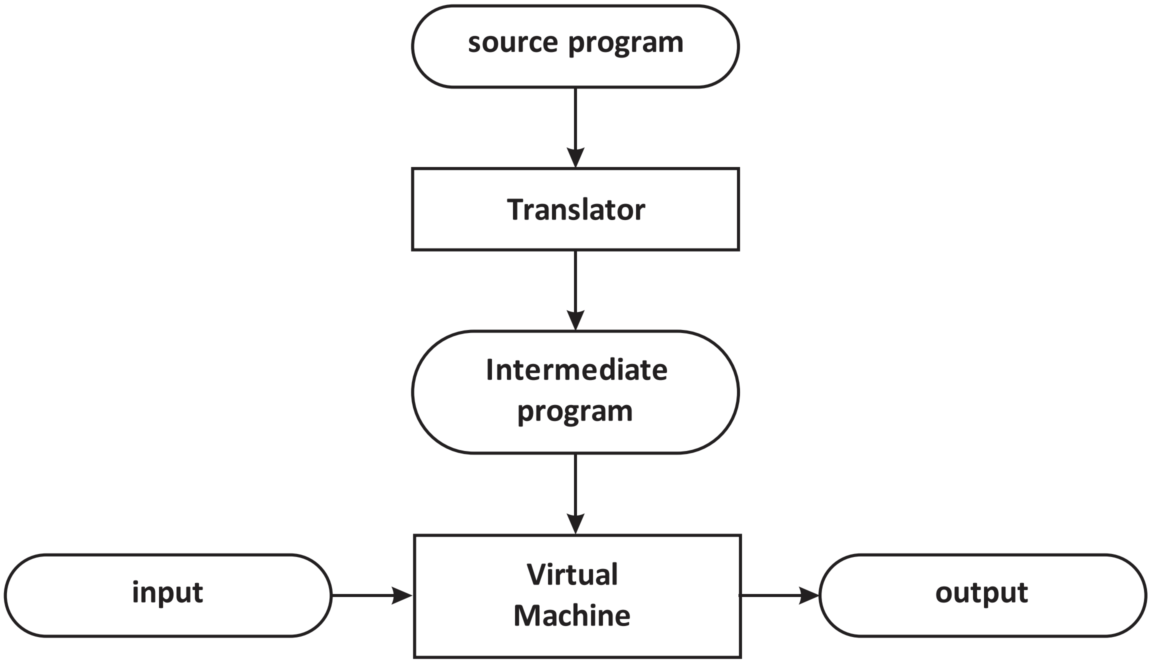


Figure 4. A hybrid system

Hybrid system example 1: Certain Pascal compilers produce an intermediate language called P-Code. P-Code programs are executed by a virtual machine called a P-Machine.

Hybrid system example 2: Nearly all Java language processors conform to the diagram in figure 4. Java programs are translated to an intermediate form call bytecodes. Bytecode programs are executed by a Java virtual machine.

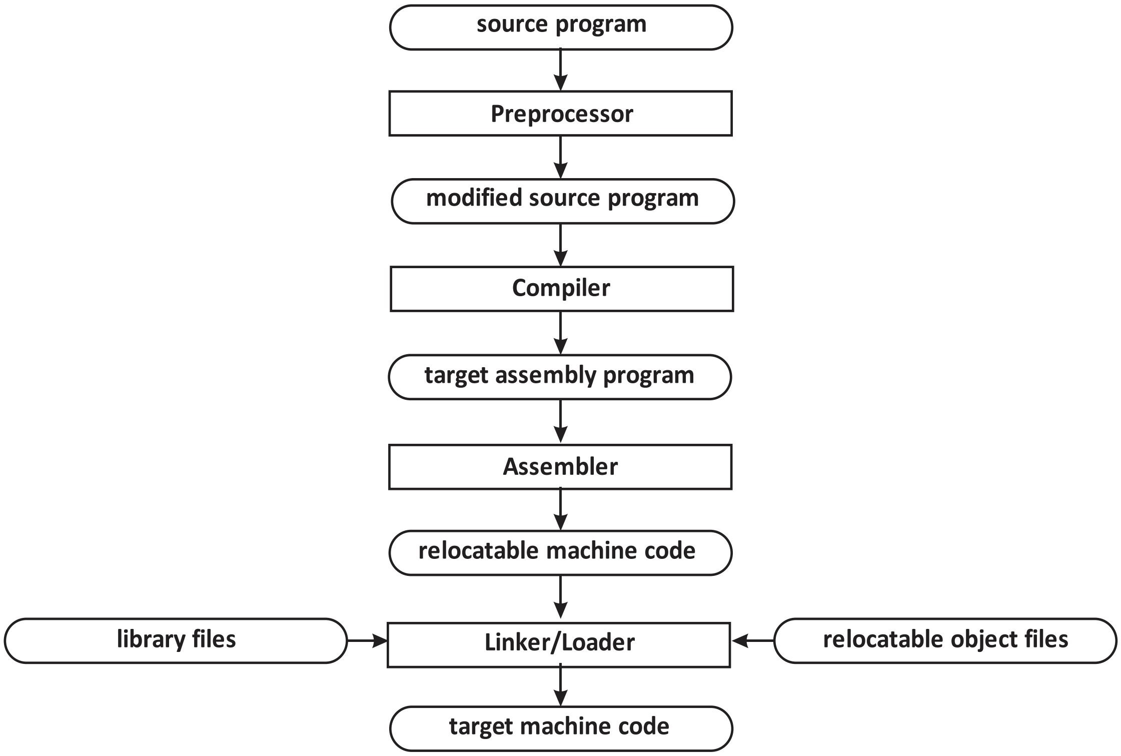


Figure 5. A language-processing system

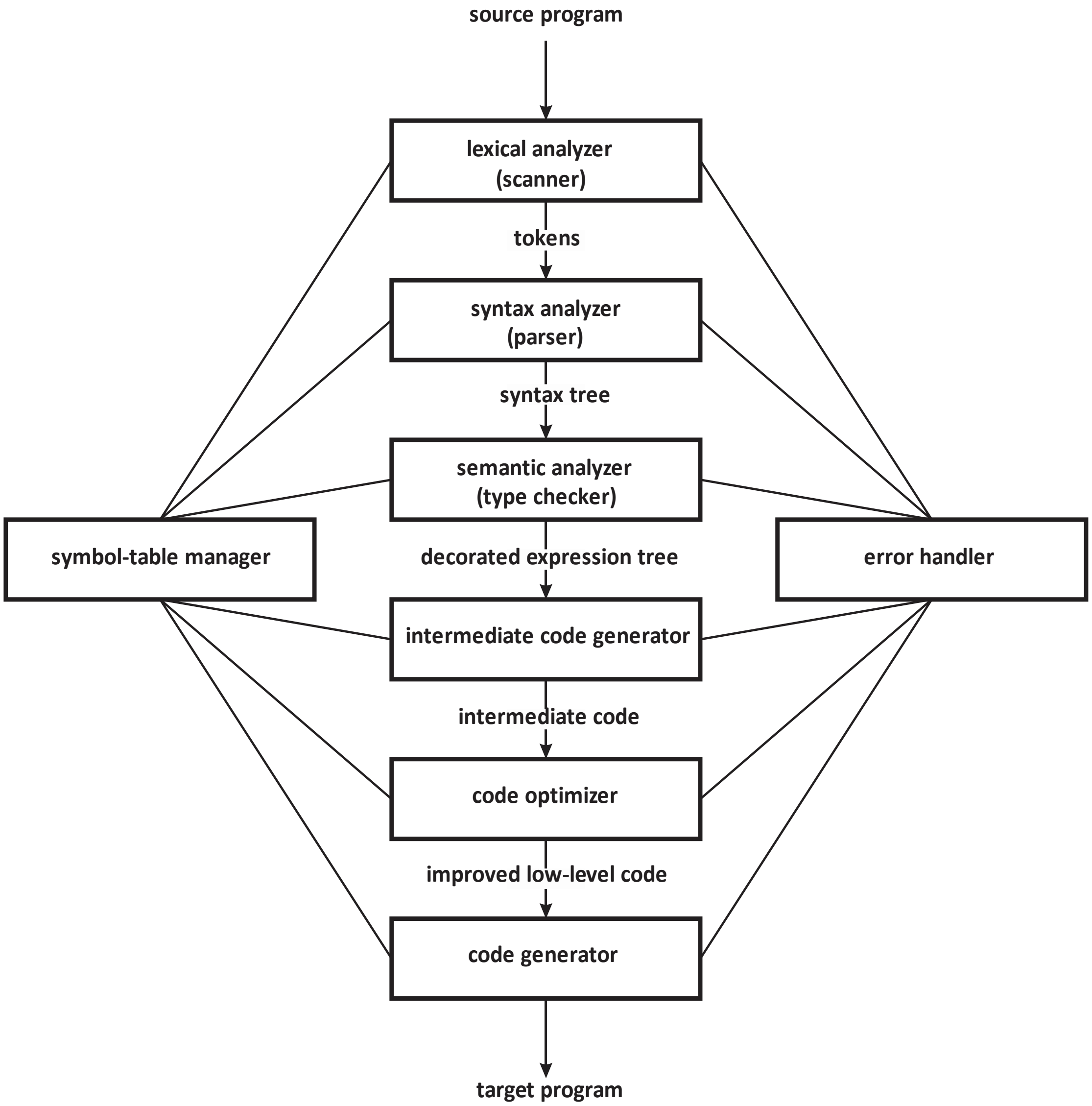
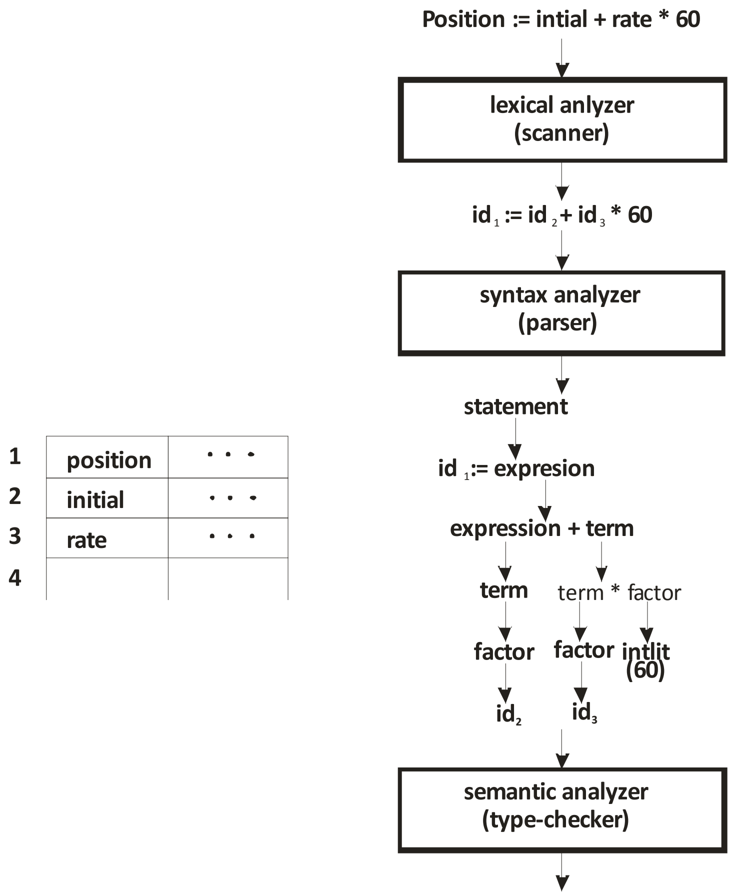
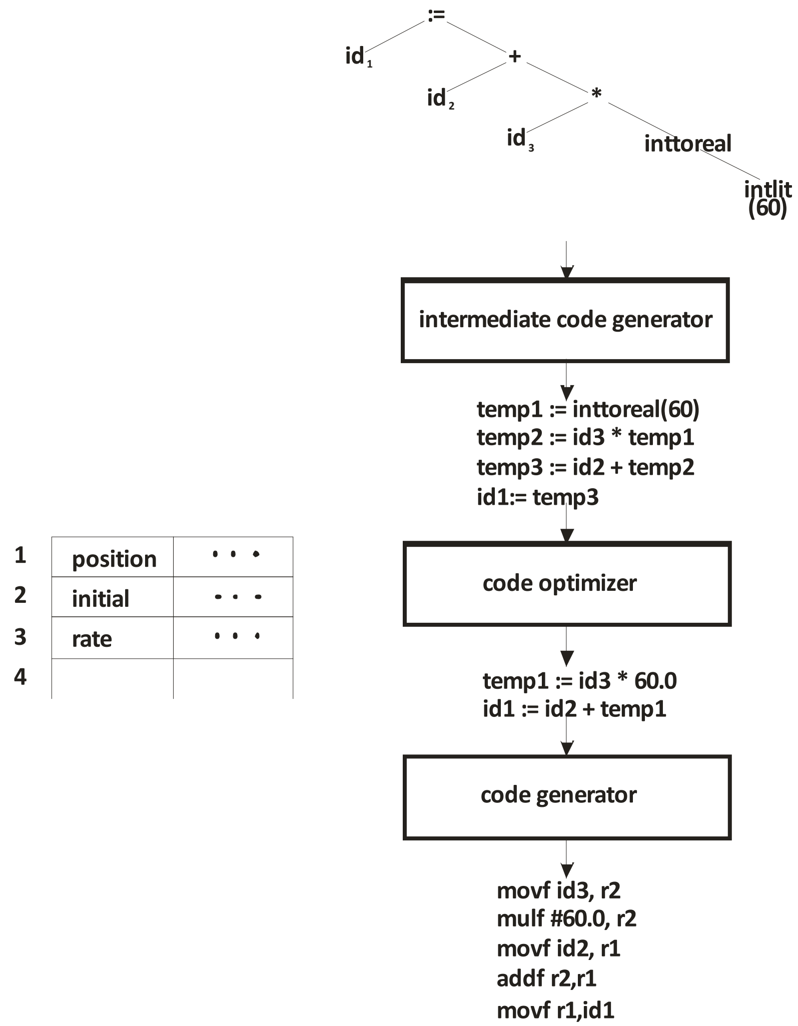


Figure 6. Anatomy of a Compiler



1	position	. . .
2	initial	. . .
3	rate	. . .
4		

Figure 7. Translation of a statement.



```
temp1 := inttoreal(60)
temp2 := id3 * temp1
temp3 := id2 + temp2
id1:= temp3
```

1	position	• • •
2	initial	• • •
3	rate	• • •
4		

```
temp1 := id3 * 60.0
id1 := id2 + temp1
```

```
movf id3, r2
mulf #60.0, r2
movf id2, r1
addf r2,r1
movf r1,id1
```

Figure 7. Translation of a statement (continued)

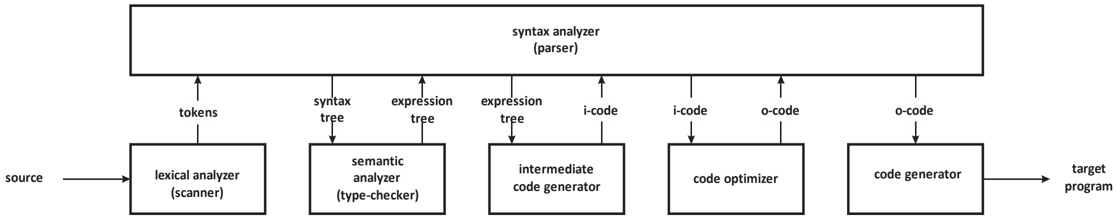


Figure 8. Compiler organization