

Chapter 1

Introduction to Compilers

Content

- ♥ Introduction to compilers
- ♥ Introduction to compiler generators
- ♥ Introduction to automatic tool generators

Programming Languages

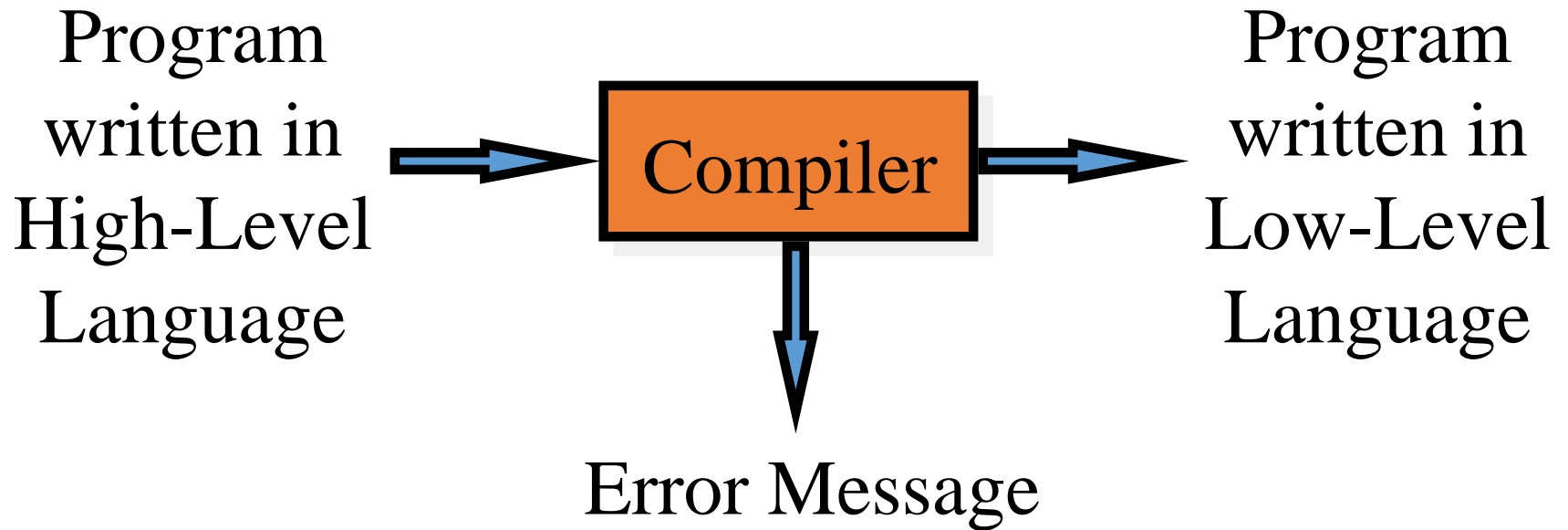
♥ Human uses human languages to **communicate** with each other

- Chinese, English, French

♥ Human uses programming languages to **communicate** with computers

- Fortran, C, Java

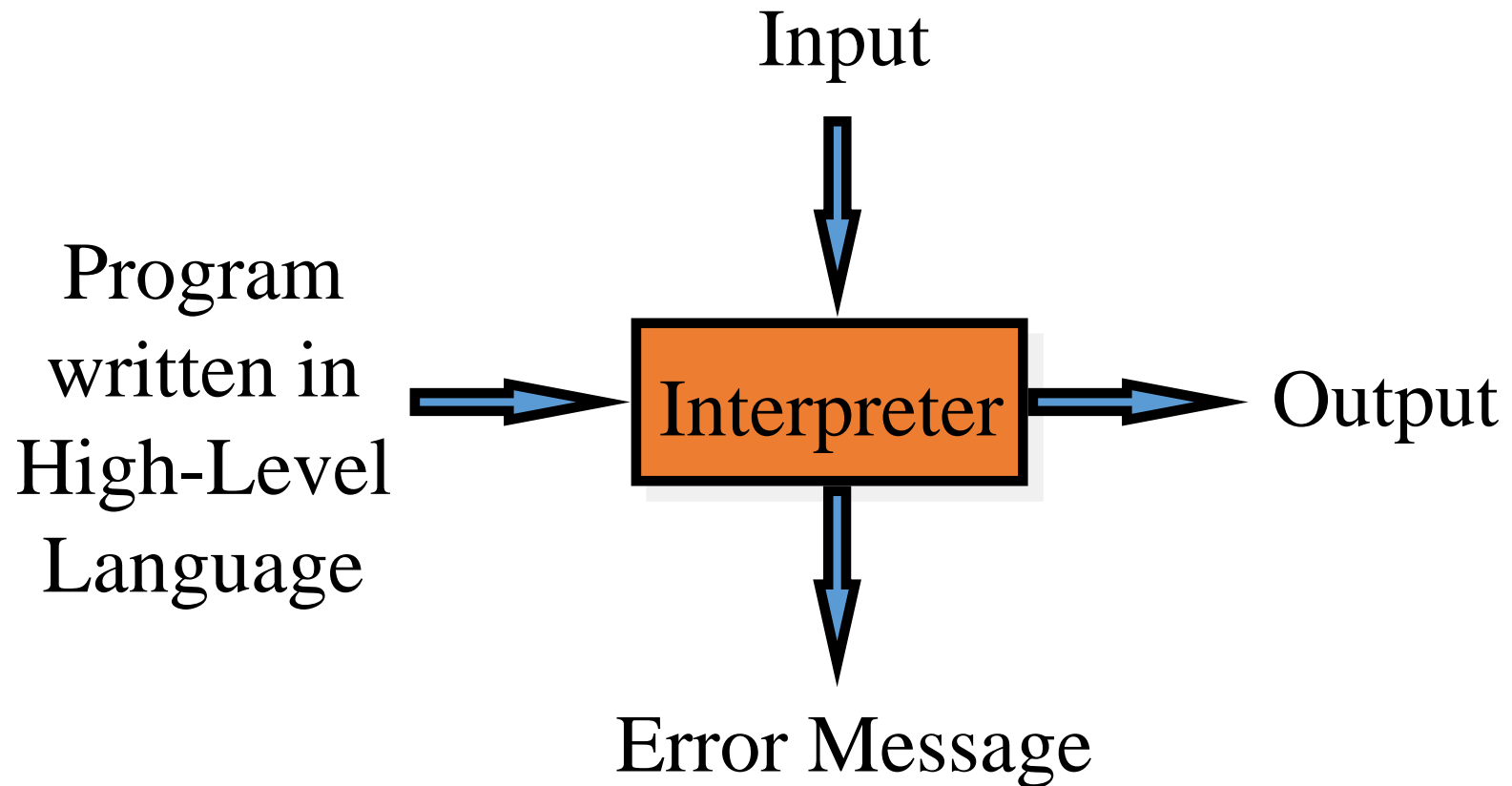
Compilers



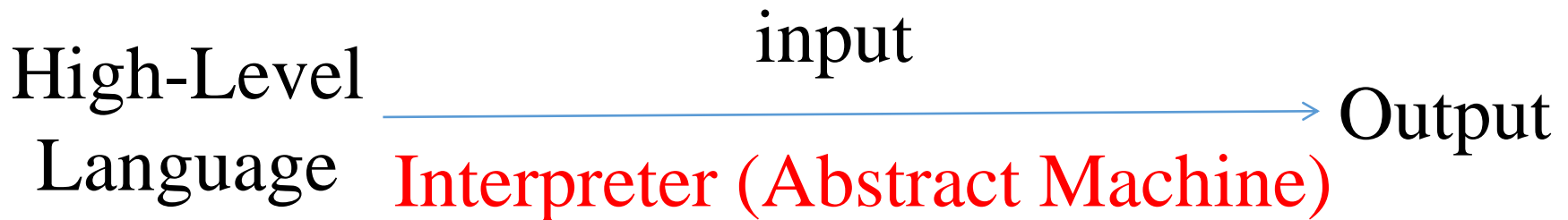
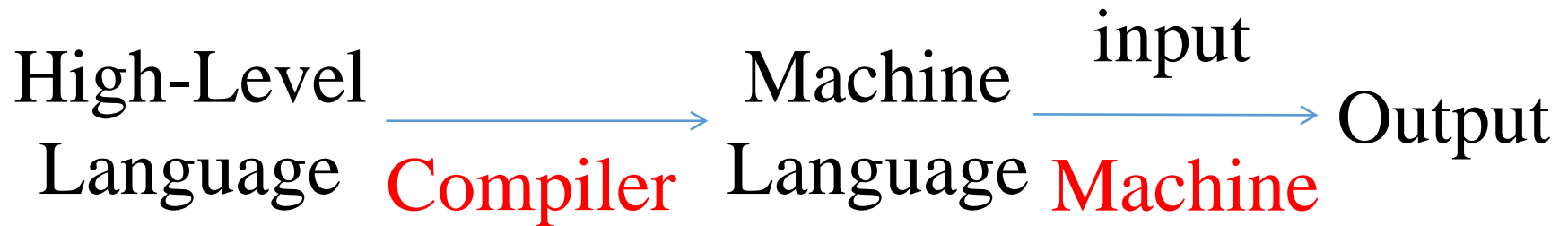
Computer Organization

Applications (High Level Language)
Compiler
Operating System (Low Level Language)
Hardware Machine

Interpreters



Compilers and Interpreters



Components of a Compiler

♥Analysis

- Lexical Analysis
- Syntax Analysis
- Semantic Analysis

♥Synthesis

- Intermediate Code Generation
- Code Optimization
- Code Generation

Lexical Analysis

S o m e o n e b r e a k s
t h e i c e



Lexical Analysis



Someone breaks the ice

final := initial + rate * 60



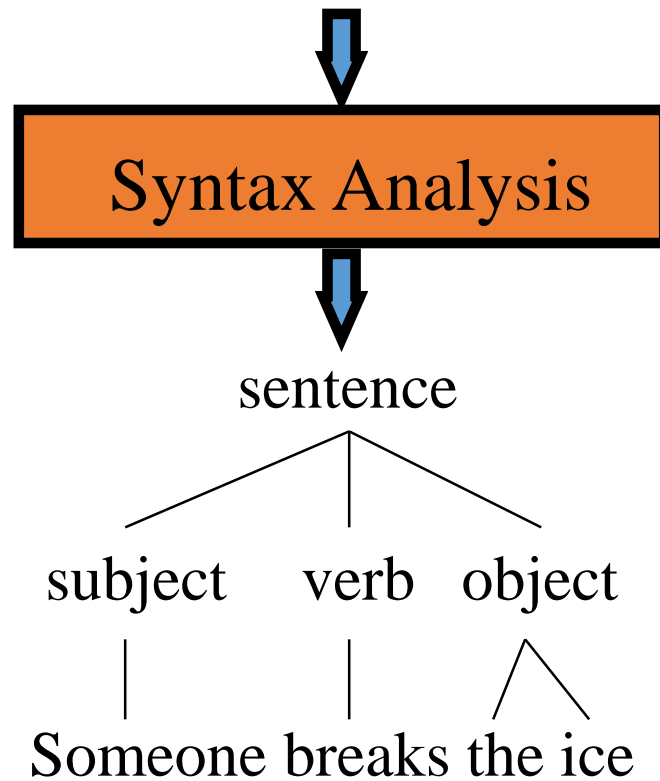
Lexical Analysis



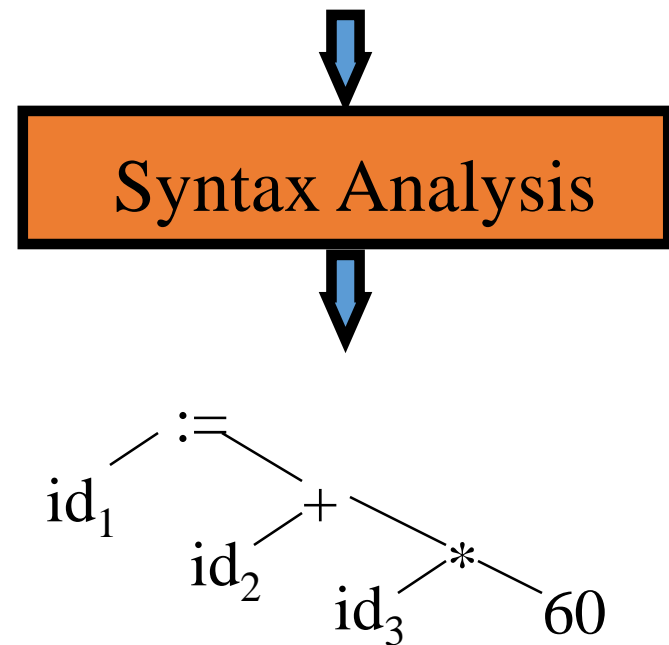
id₁ := id₂ + id₃ * 60

Syntax Analysis

Someone breaks the ice



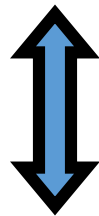
$\text{id}_1 := \text{id}_2 + \text{id}_3 * 60$



Semantic Analysis

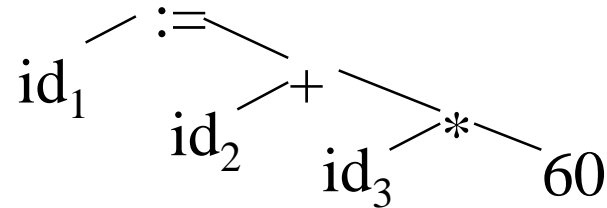
Someone plays the piano

(meaningful)

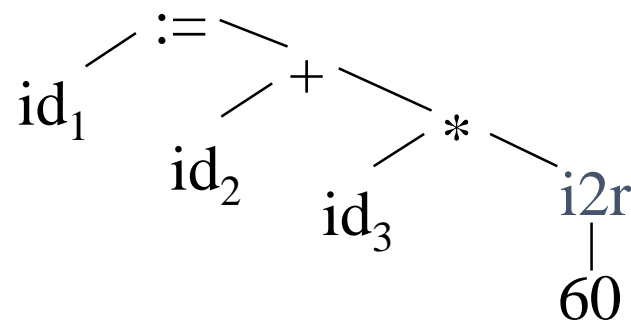


The piano plays someone

(meaningless)



Semantic Analysis

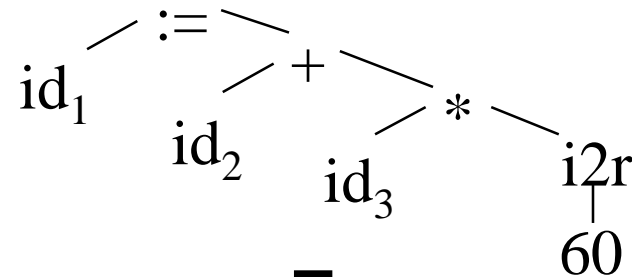


Intermediate Code Generation

Someone breaks the ice



有人打破冰



Intermediate Code Generation



```
temp1 := i2r ( 60 )  
temp2 := id3 * temp1  
temp3 := id2 + temp2  
id1 := temp3
```

Code Optimization

有人打破冰



Code Optimization



有人打破沉默

```
temp1 := i2r ( 60 )  
temp2 := id3 * temp1  
temp3 := id2 + temp2  
id1 := temp3
```



Code Optimization



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

Code Generation

有人打破沉默



有人打破沉默

$\text{temp1} := \text{id}_3 * 60.0$
 $\text{id}_1 := \text{id}_2 + \text{temp1}$



Code Generation



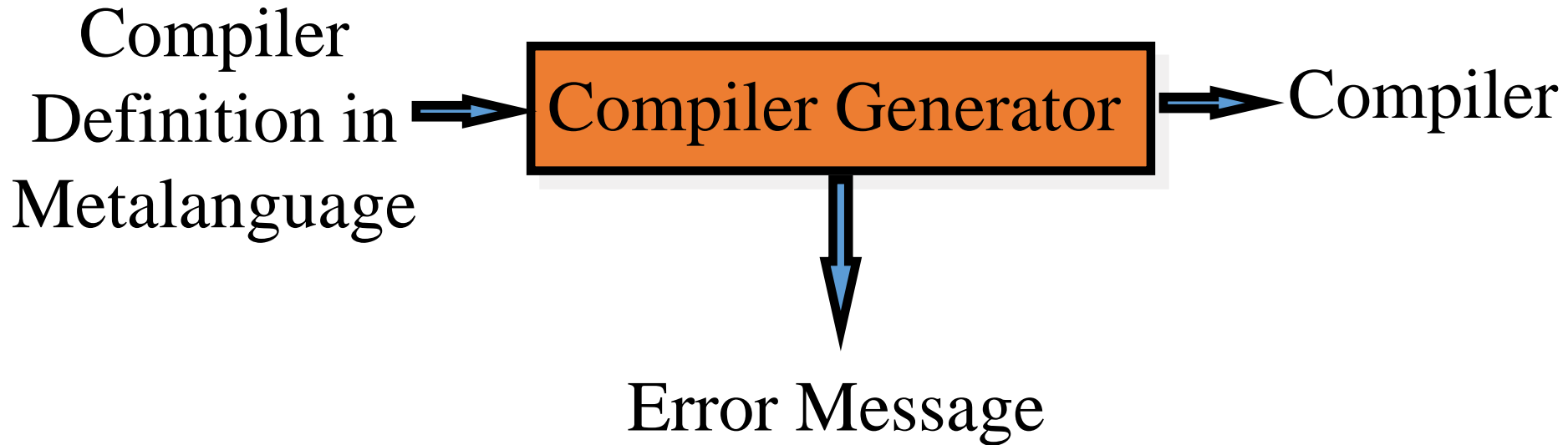
```
movf    id3, r2
mulf    #60.0, r2
movf    id2, r1
addf    r2, r1
movf    r1, id1
```

Metalanguages

♥ **Metalanguage:** a language used to define another language

We will use different *metalanguages* to define the various components of a programming language so that these components can be generated automatically

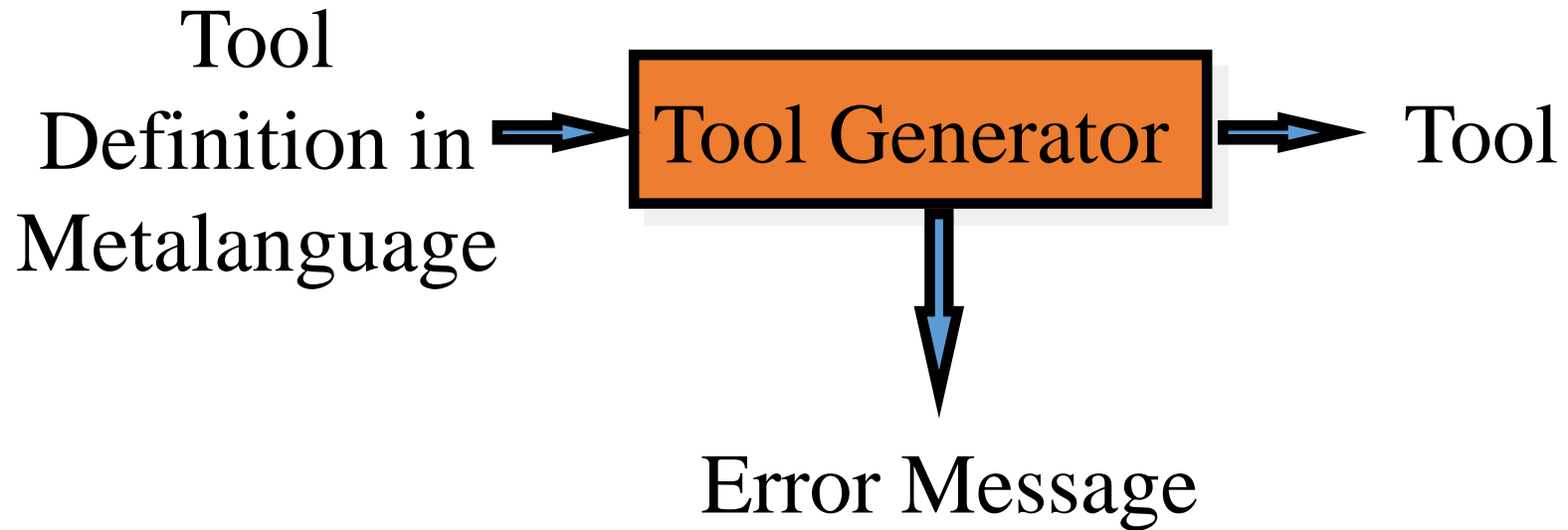
Compiler Generators



Definition of Programming Languages

- ♥ *Lexical analysis*: regular expressions
- ♥ *Syntax analysis*: context free grammars
- ♥ *Semantics analysis*: attribute grammars
- ♥ *Intermediate code generation*:
attribute grammars
- ♥ *Code generation*: tree grammars

Automatic Tool Generators



Applications of Compilation Techniques

- ♥ Web Browsers (HTML, XML, ...)
- ♥ Word Processors (postscript, pdf, ...)
- ♥ Computer-Aided Software Engineering (UML)
- ♥ Computer-Aided Design (VHDL, Verilog, ...)
- ♥ Computer-Aided Manufacturing (APT, G-code)

Outline of This Course

- ♥Lexical analysis
- ♥Syntax analysis
- ♥Semantic analysis
- ♥Intermediate code generation
- ♥Code generation