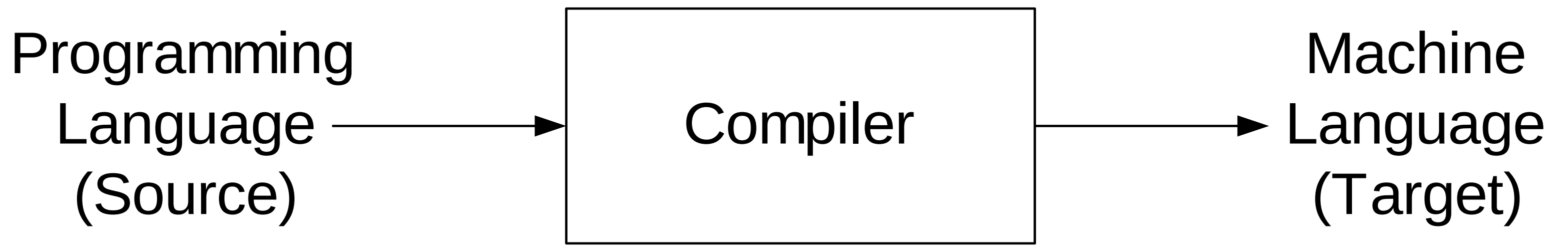
Compiler

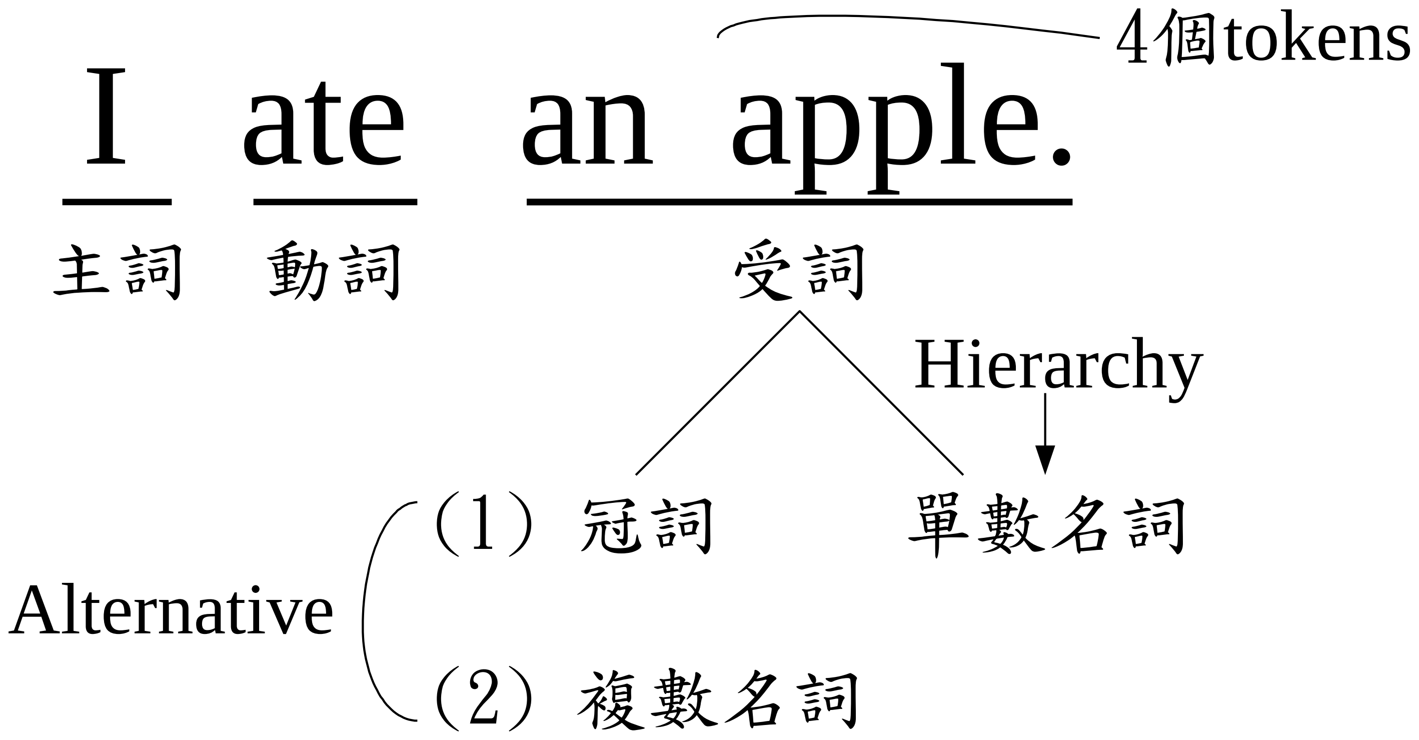
<<Compiler.ppt>>

Compiler

- A compiler allows virtually all computer users to <u>ignore</u> the <u>machine dependent</u> details of machine language.
  - (Sun workstation C ? PC C ? NO !!)
- Compilers therefore allow programs and programming expertise to be *machine-independent.*

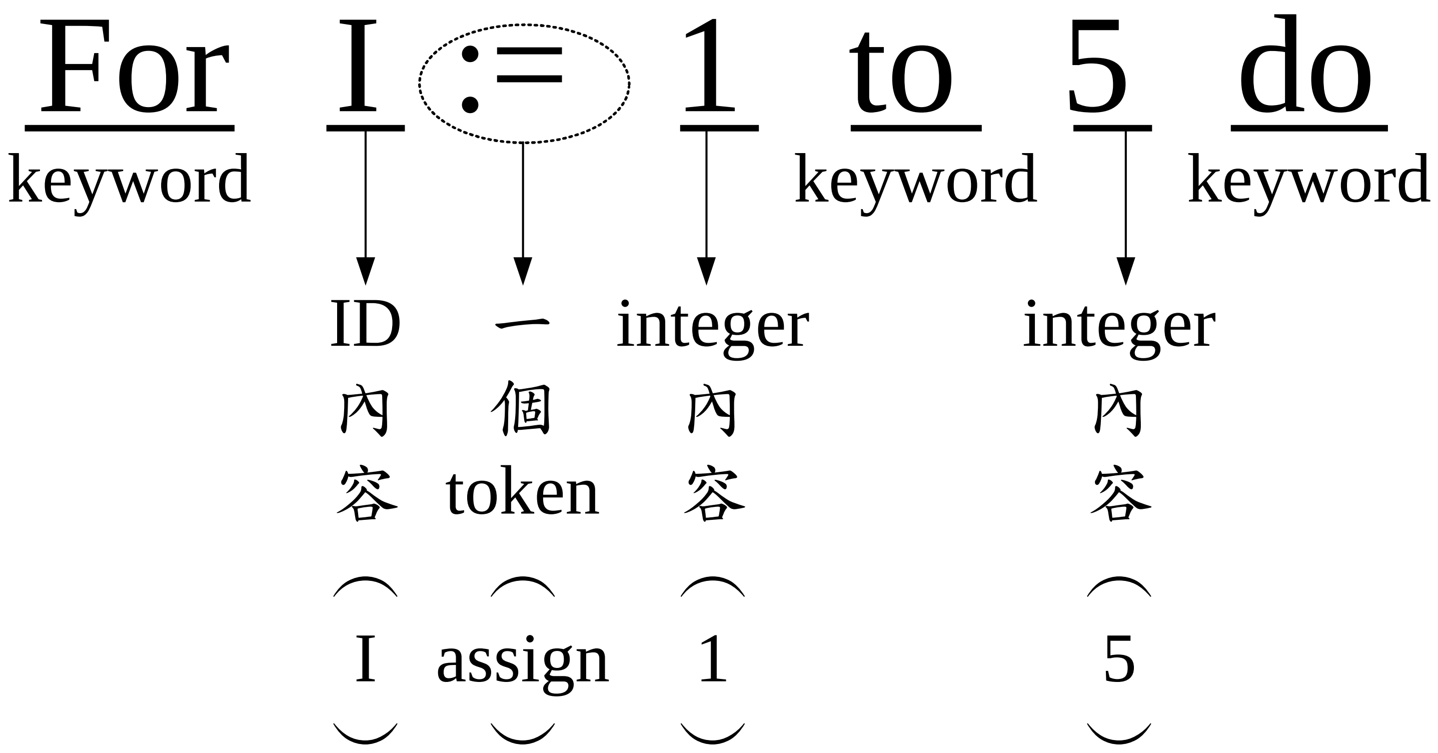


- Translator: assembler, compiler, interpreter, preprocessor.
- Compiler vs. Interpreter
  - portability, execution speed, with/without object code, with/without optimization, debugging capability.
- Hardware → OS → Compiler → Application.
- **1st compiler: 1950s IBM Fortran (written in a machine language by using some software tool).**



Syntax: 文法 (parsing ->YACC)
幫你做parser

Token: 文法處理單元 (type與內容)



Lexical Analysis : 找出 token (Lex, 幫你辨認token)

- Input: a sequence of characters.
- Output: a sequence of tokens

```
    If  (X=1)  Then
         If  (Y=2)  Then
         Begin
               Writeln('Hi');
               Z := X+Y;
         End
         Else
               Writeln('Bad')
    Else
         Writeln('Bye');
```

- Grammar: BNF grammar, grammar chart.
  - Terminal/Non-Terminal symbols.
  - EX.

  <ID>::= <letter> | <ID> <letter>| <ID> <digit>

  <letter> ::=A|B|…|Z|a|b|…|z

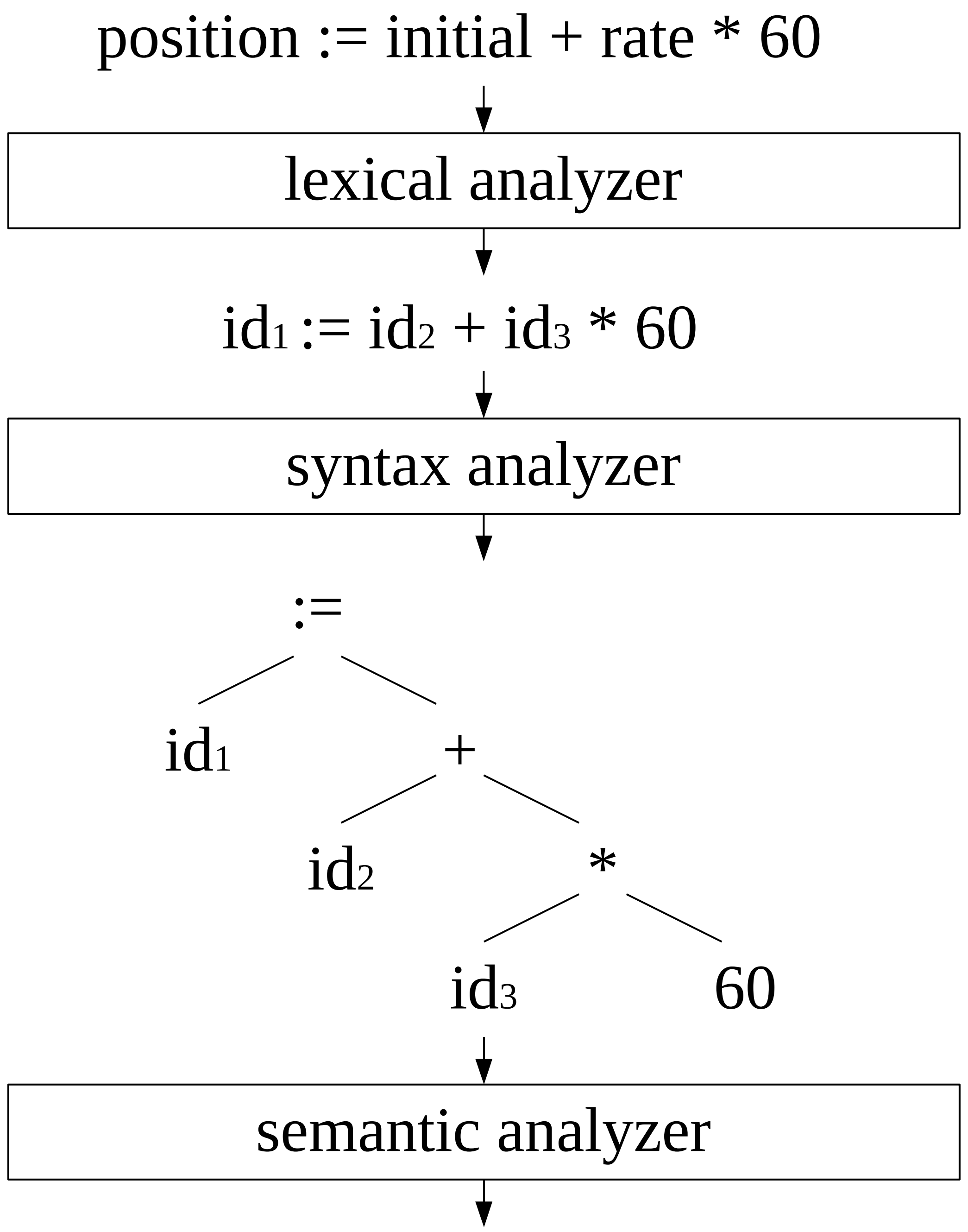
  <digit> ::= 0|1|…|9

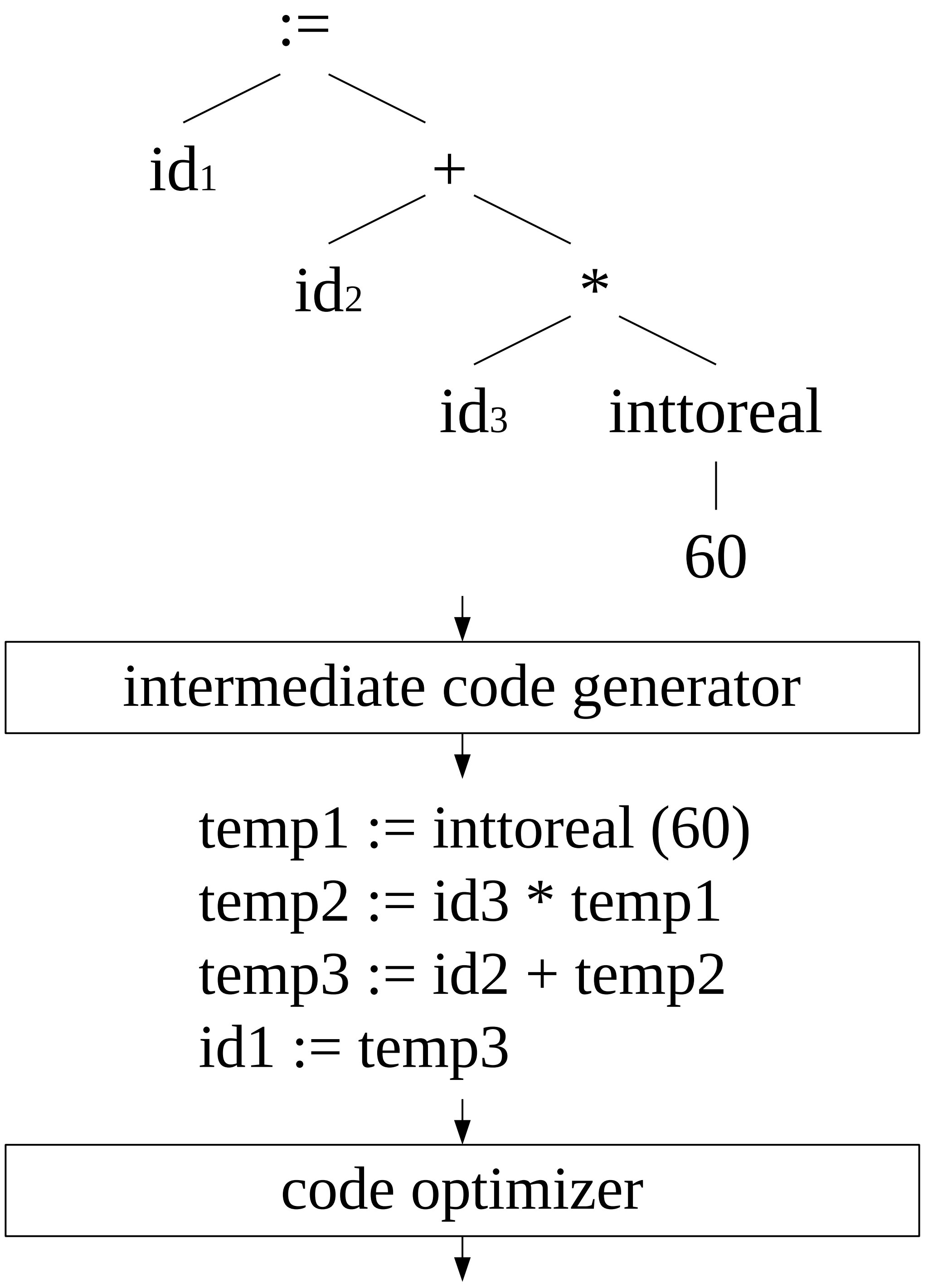
編譯程式簡介

- 編譯程式的七個階段中，語彙分析(Lexical analysis)、語法分析(Syntax analysis)、解釋(Interpretation)及與機器無關的最佳化(Machine independent optimization)等前面四個階段是”與機器無關而與語言有關” 。**(Analysis)**

編譯程式簡介

- 編譯程式的七個階段中，儲存位置的分配(Storage assignment)、數碼產生(Code generation)及組合並輸出(Assembly and output)等後面三個階段是與機器有關而與語言無關”　。**(Synthesis)**


position := initial + rate * 60
lexical analyzer
$id_1 := id_2 + id_3 * 60$
syntax analyzer
:=
id_1
+
id_2
*
id_3
60
semantic analyzer



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

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

$\downarrow$

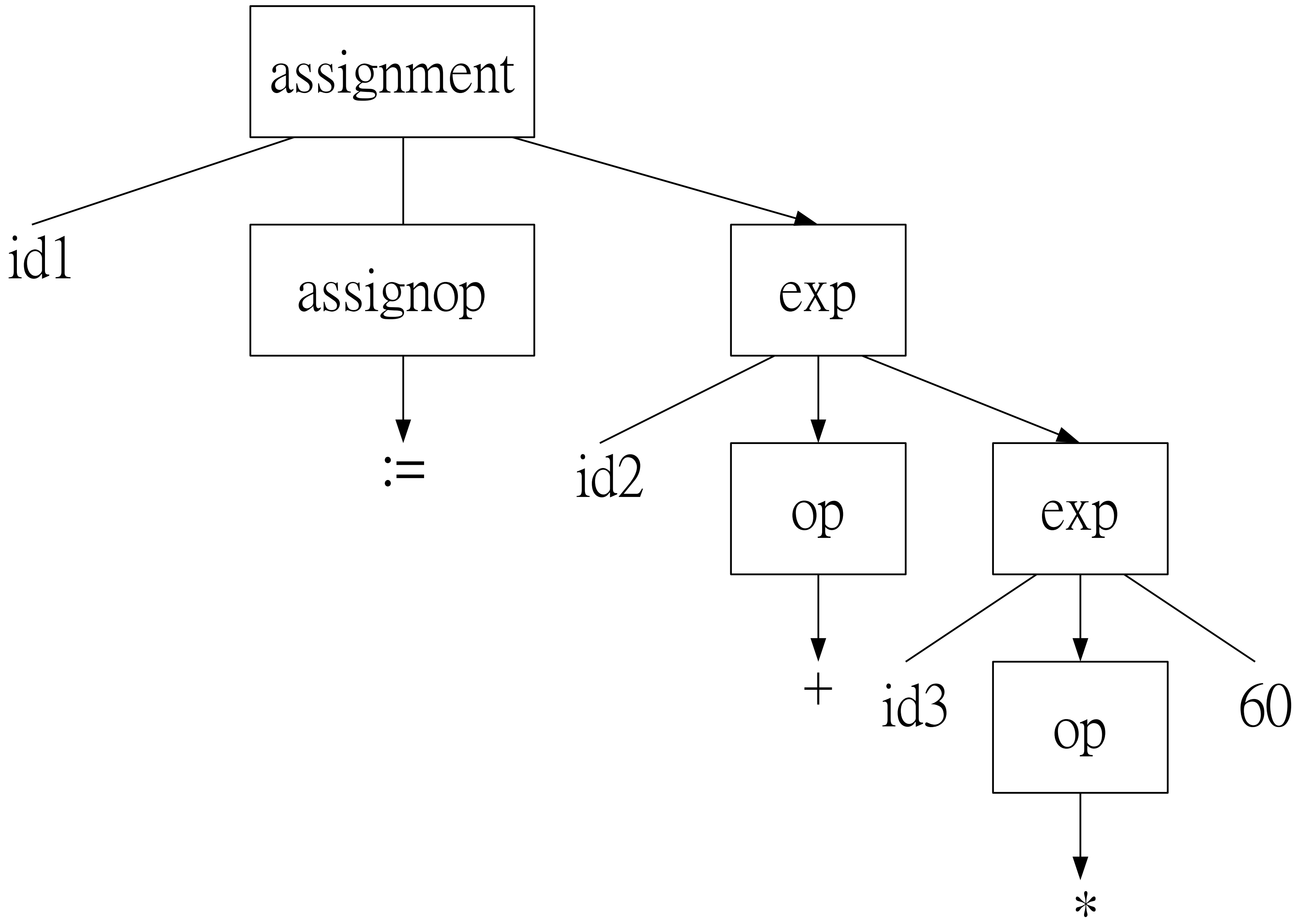
$\boxed{\text{code generator}}$

$\downarrow$

```
MOVF id3, R2
MULF #60.0,  R2
MOVF id2, R1
ADDF R2, R1
MOVF R1, id1
```

SYMBOL TABLE

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


assignment
id1
assignop
exp
:=
id2
op
exp
+
id3
op
60
*

- A Parsing Tree (for syntax analysis): decide the execution order (operator priority, association).
- Semantic analyzer: type checking, type conversion.
  - A = B+5; (* a syntax error in PASCAL*)
  - A:= B+5; (* A: real; B: integer; ➔ need type conversion; require different integer/real ADD operator *)

- A:=B+5; (* A: integer; B: real; ➔ a semantic error; type checking ; A[1], A[1.5] ?? *)
- One/Two/N pass(es) compiler.
- Cross compiler: A compiler that runs on one machine and produces the object code for another machine.
- Error detection/reporting.