编译原理: 概述

魏恒峰

hfwei@nju.edu.cn

2020 年 10 月 31 日

$$8 \text{ 周} = 7.5 \text{ 周 (授课)} + 0.5 \text{ 周 (期末测试)}$$

8 周 = 7.5 周 (授课) + 0.5 周 (期末测试)

期末测试 (40 分): 2020 年 12 月 31 日; 时长 2 小时; 闭卷

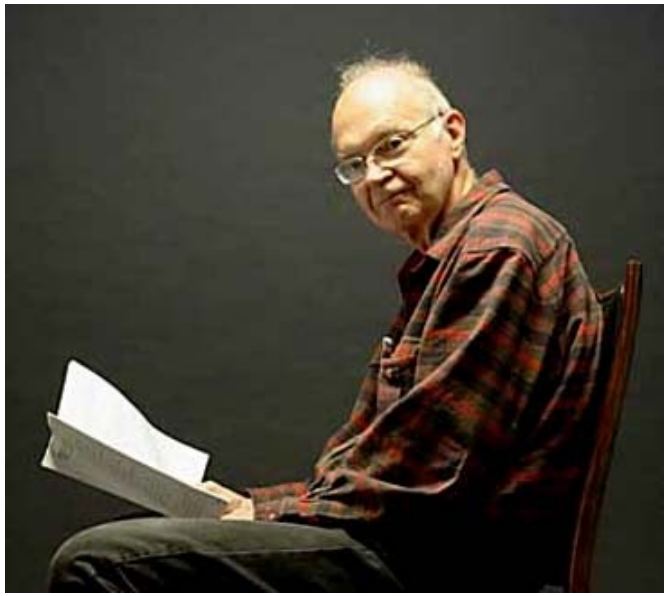
作业 (15 分): 7 次必做作业 + 1 次选做报告 (+3 分)

实验 (45 分): 4 次必做实验 + 1 次选做实验 (+5 分)

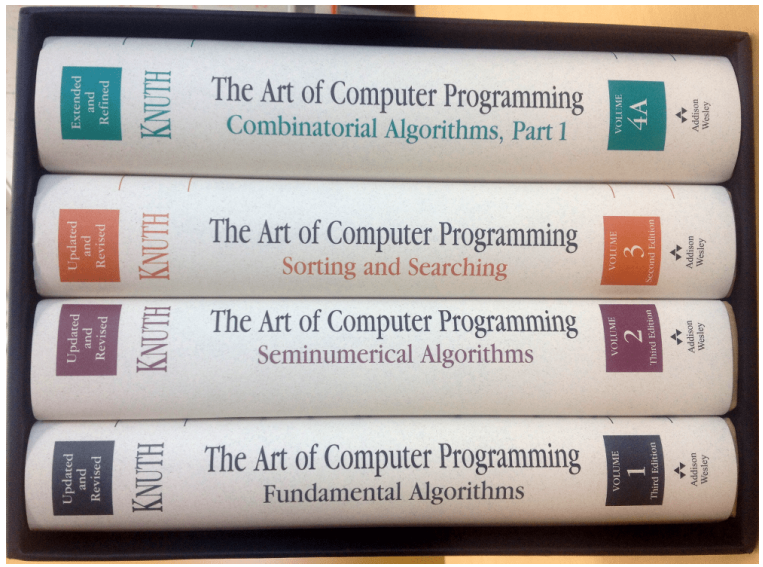
$$45 = 5 + 15 + 15 + 10$$

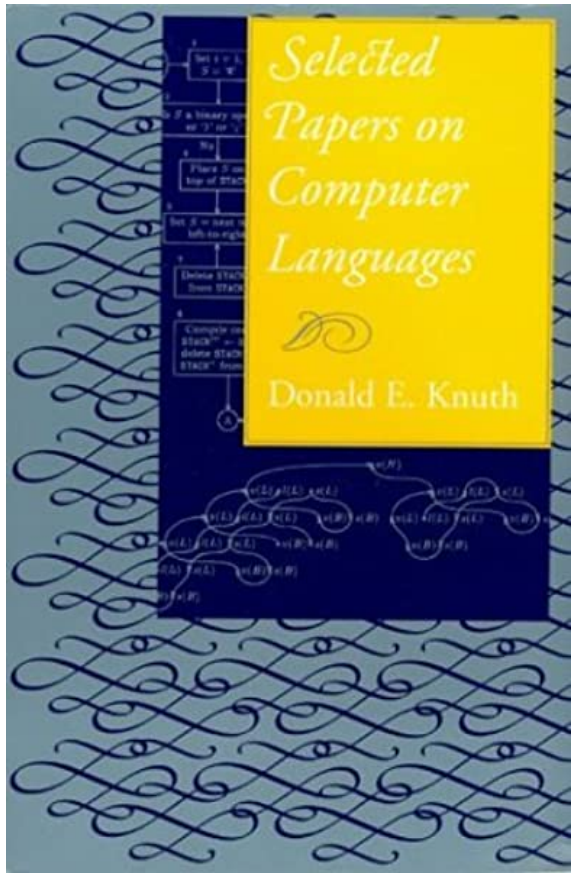
一次：当次作业计零分　　两次：所有作业计零分　　三次：总评零分

“father of the analysis of algorithms”

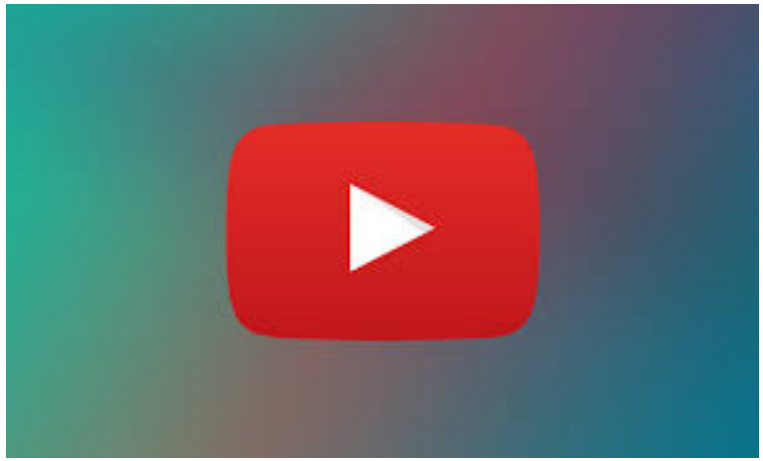


Donald E. Knuth (1938 ∼)





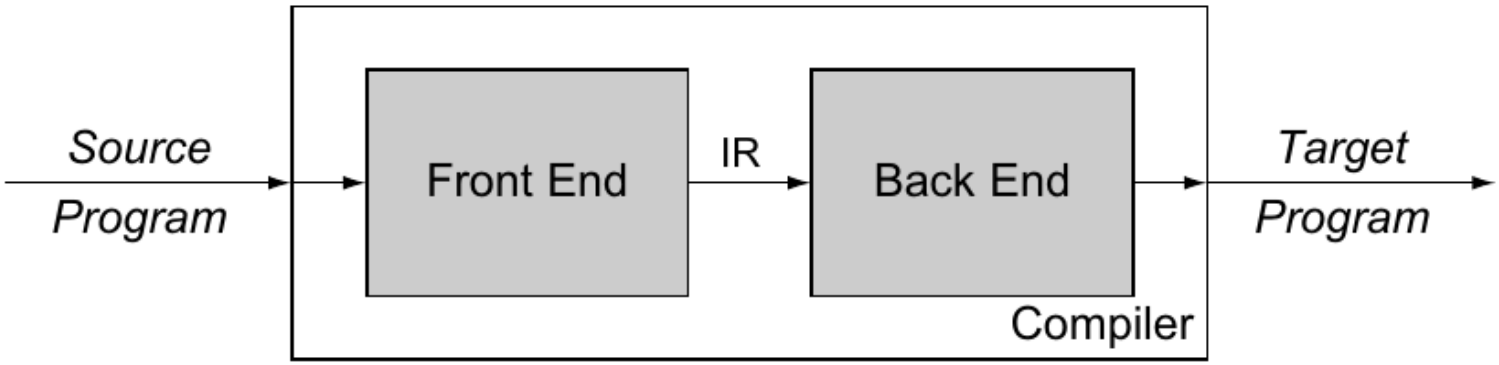
ALGO Compiler **LR Parser** **Attribute Grammar**



Writing a Compiler for the Burroughs Corporation

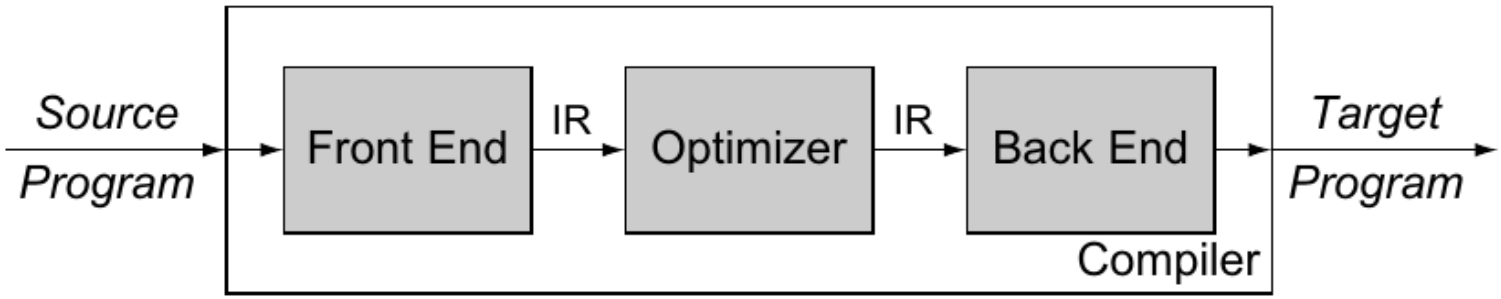
“高级”语言 $\Longrightarrow$ “低级”语言 (如, 汇编语言)

两个月的“编译器设计原理”之旅



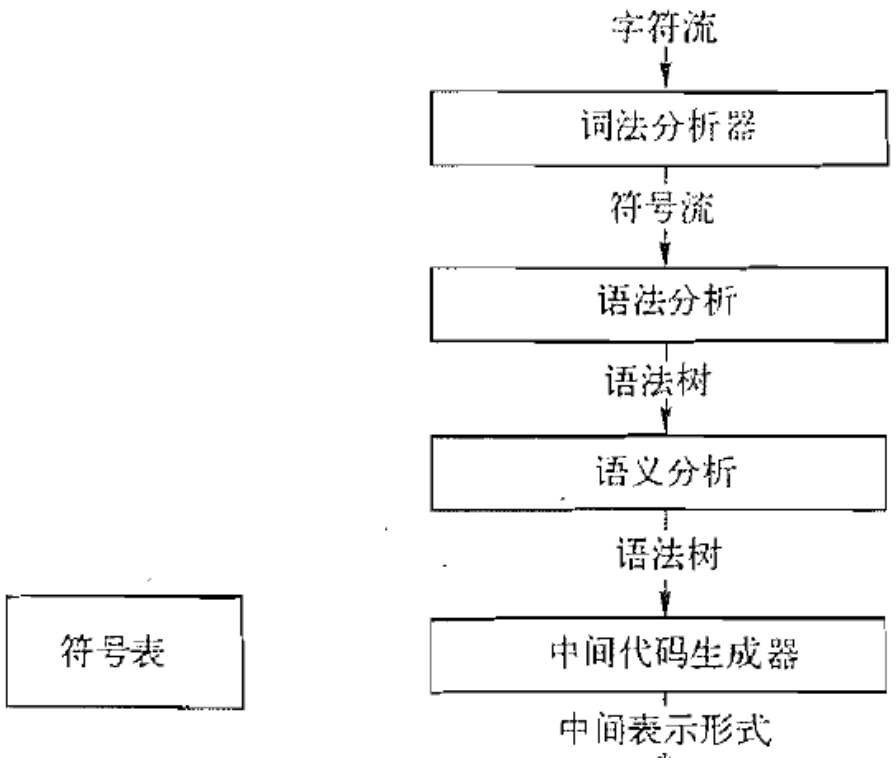
IR: Intermediate Representation (中间表示)

模块化设计

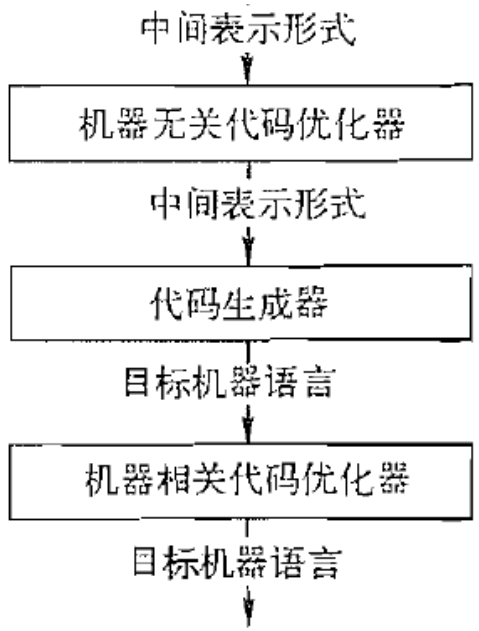


机器无关的中间表示优化

编译器前端：分析阶段



编译器后端：综合阶段



词法分析器 (Scanner): 将**字符**流转化为**词法单元** (token) 流。

词法分析 (Lexical Analysis)

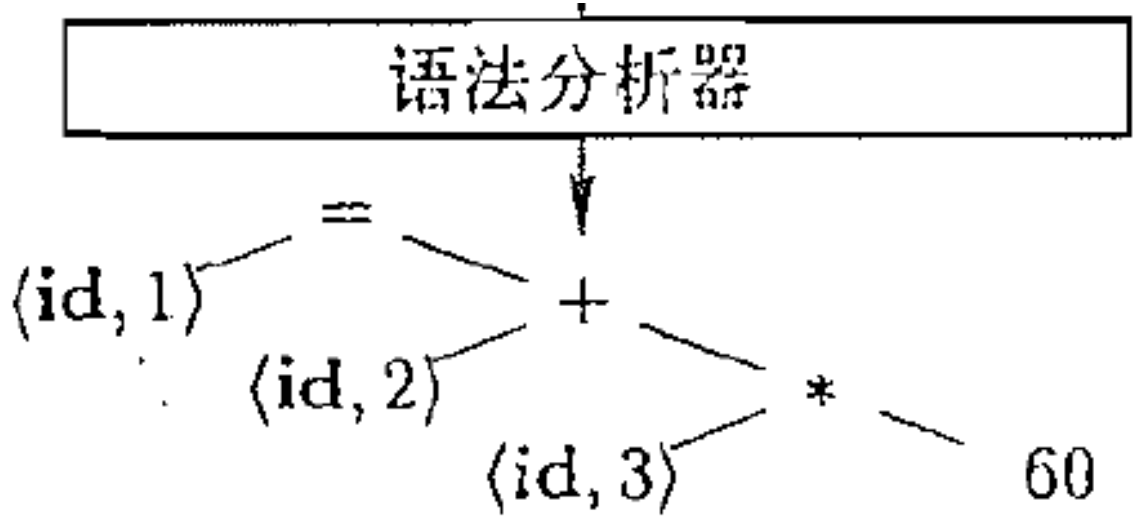
```
position = initial + rate * 60
```

token : $\langle$token-class, attribute-value$\rangle$

$\langle \mathbf{id}, 1\rangle$ $\langle \mathbf{ws}\rangle$ $\langle \mathbf{assign}\rangle$ $\langle \mathbf{ws}\rangle$ $\langle \mathbf{id}, 2\rangle$ $\langle \mathbf{ws}\rangle$
$\langle + \rangle$ $\langle \mathbf{ws}\rangle$ $\langle \mathbf{id}, 3\rangle$ $\langle \mathbf{ws}\rangle$ $\langle * \rangle$ $\langle \mathbf{ws}\rangle$ $\langle \mathbf{num}, 4\rangle$

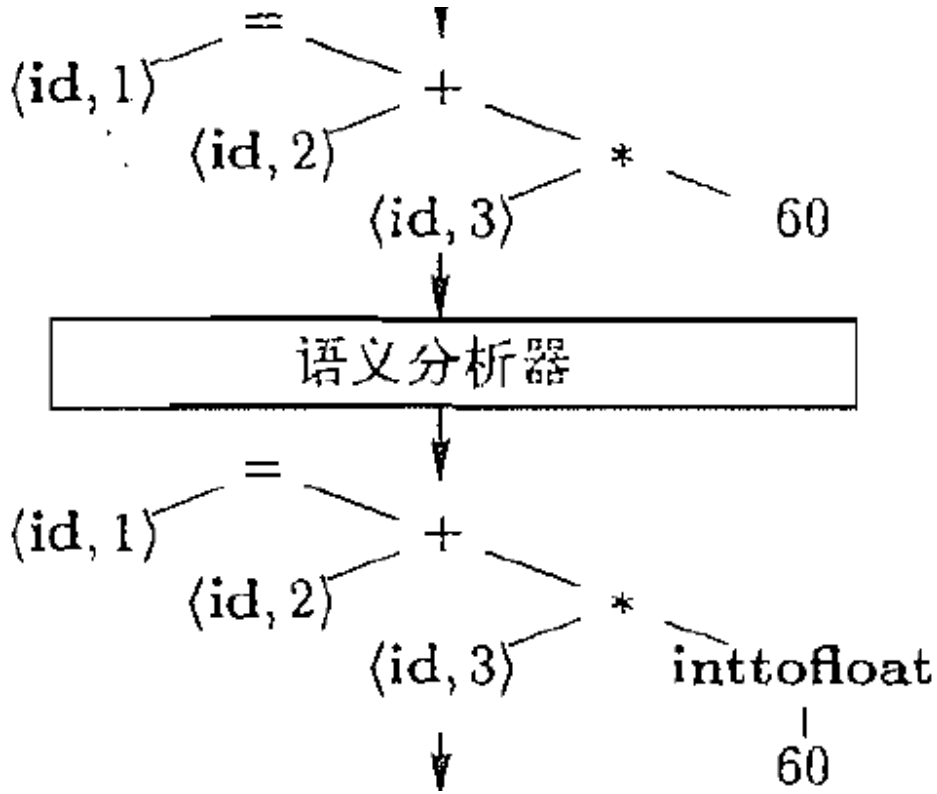
语法分析器 (Parser): 构建**词法单元**之间的**语法结构**

语法分析 (Syntax Analysis)



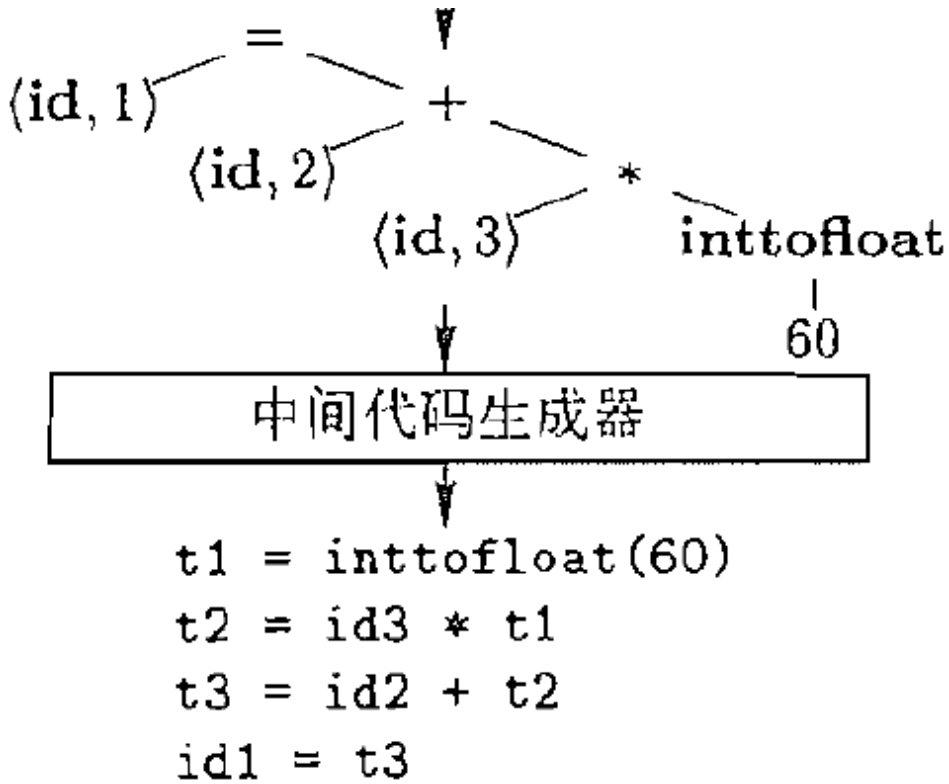
语义分析器: 语义检查, 如类型检查、一致性约束检查

语义分析 (Semantic Analysis)



中间代码生成器: 生成中间代码, 如 “**三地址代码**”

中间代码生成



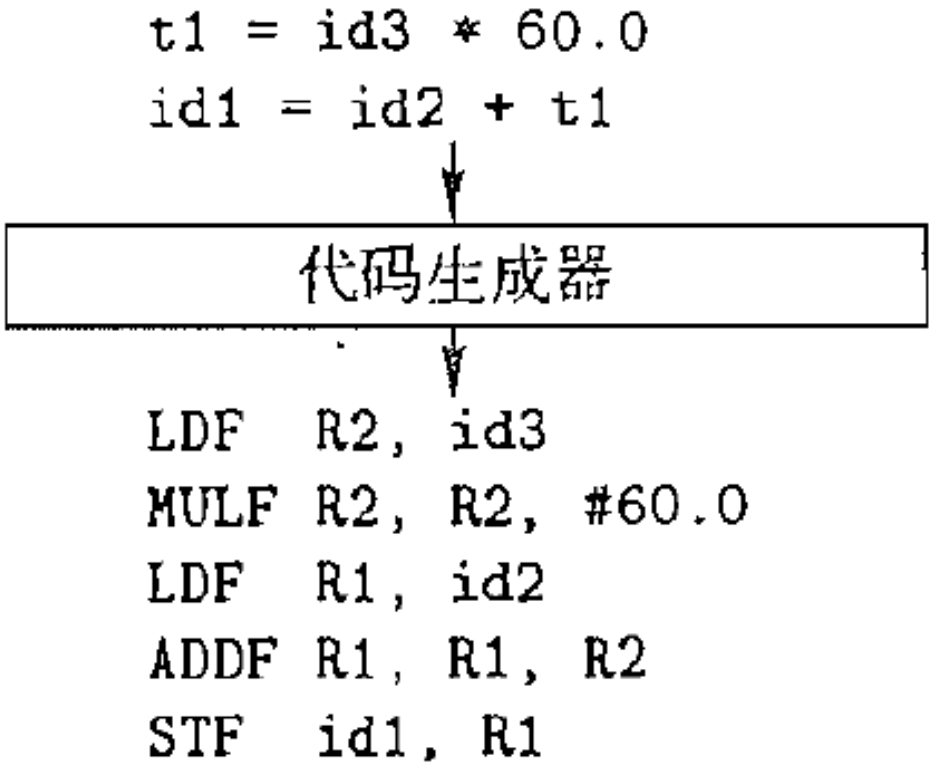
中间代码优化

```
t1 = inttofloat(60)
t2 = id3 * t1
t3 = id2 + t2
id1 = t3
```

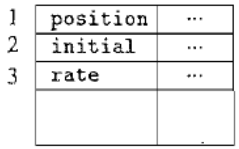

```
t1 = id3 * 60.0
id1 = id2 + t1
```

代码生成器：生成目标代码，主要任务包括**指令选择、寄存器分配**

中间代码生成

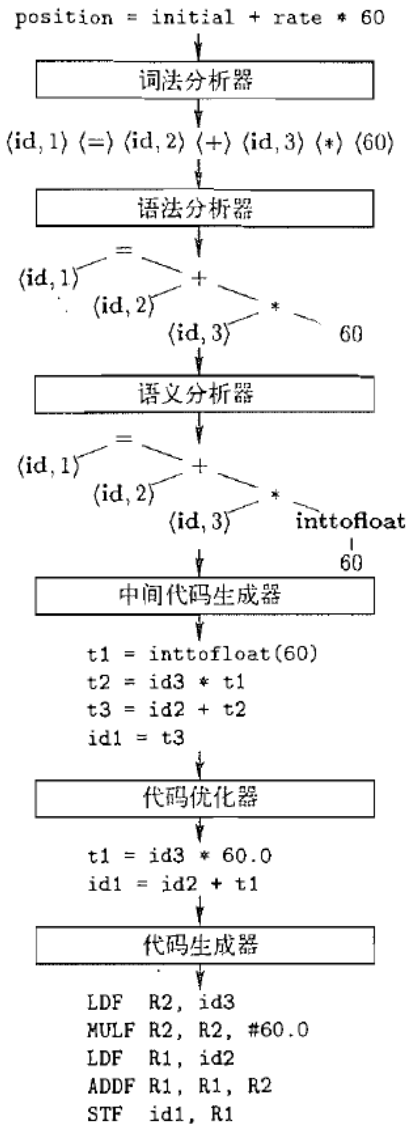


符号表：收集并管理变量名/函数名相关的信息



1	position	···
2	initial	···
3	rate	···

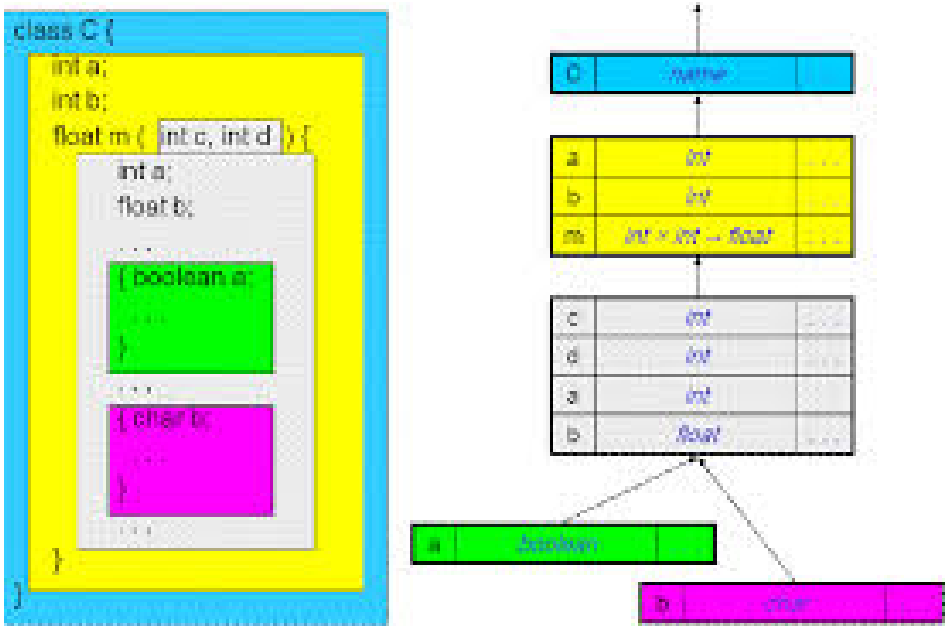
符号表



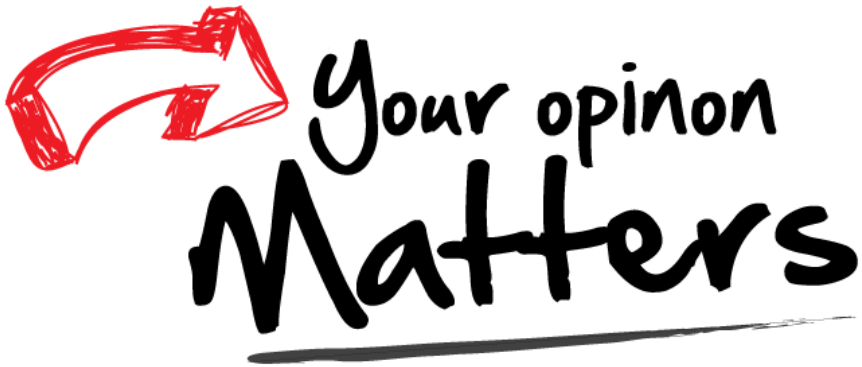
public class ST<Key extends Comparable<Key>, Value>		
	ST()	*create an empty symbol table*
void	put(Key key, Value val)	*associate* `val` *with* `key`
Value	get(Key key)	*value associated with* `key`
void	remove(Key key)	*remove* `key` *(and its associated value)*
boolean	contains(Key key)	*is there a value associated with* `key`*?*
int	size()	*number of key–value pairs*
Iterable<Key>	keys()	*all keys in the symbol table*

红黑树 (RB-Tree)、哈希表 (Hashtable)

符号表可能有多个


class C {
int a;
int b;
float m (int c, int d) {
int a;
float b;
{ boolean a;
}
{ char b;
}
}
}
C
name
a
int
b
int
m
int × int → float
c
int
d
int
a
int
b
float
a
boolean
b
char


Thank
You!



Office 926

hfwei@nju.edu.cn