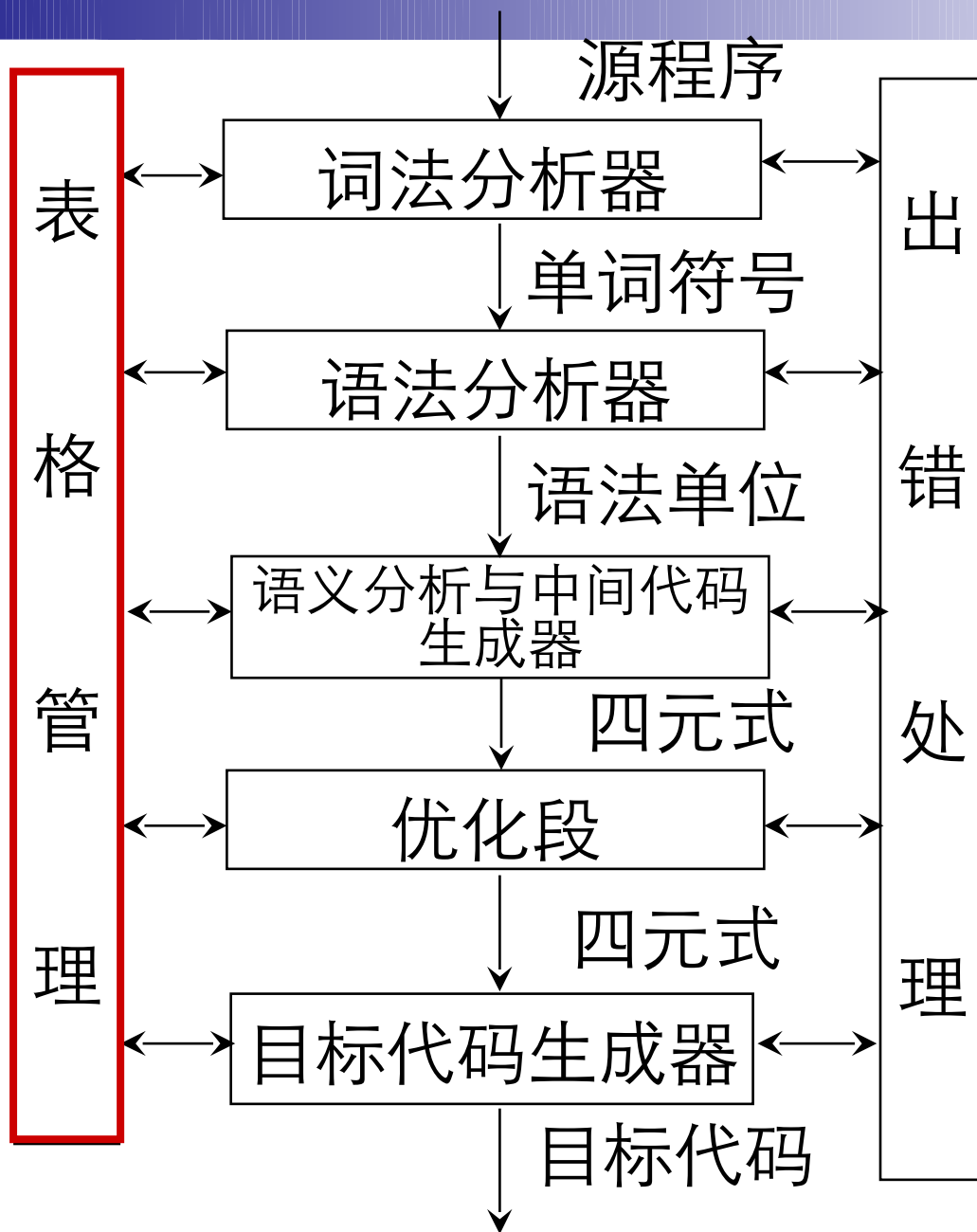




# 编译原理

## 第八章 符号表

# 编译程序总框



# 第八章 符号表

- 符号表的组织与作用
- 整理和查找
- 符号表的内容
- 名字的作用范围

# 第八章 符号表

- 符号表的组织与作用
- 整理和查找
- 符号表的内容
- 名字的作用范围

# 第八章 符号表

- 符号表的作用
  - 一致性检查和作用域分析
  - 辅助代码生成

# 8.1 符号表的组织与作用

- 符号表的每一项（入口）包含两大栏
  - 名字栏，也称主栏，关键字栏
  - 信息栏，记录相应的不同属性，分为若干子栏

名字	信息
----	----

# 8.1 符号表的组织与作用

## ■ 对符号表的操作

- 填入名称
- 查找名字
- 访问信息
- 填写修改信息
- 删除

符号名表

	NAME	INFORMATION
(1)	index	整型, 变量
(2)	score	实型, 变量
(3)	p	整型, 形式参数

# 8.1 符号表的组织与作用

- 对符号表进行操作时机

- 定义性出现

- `int index`

- 使用性出现

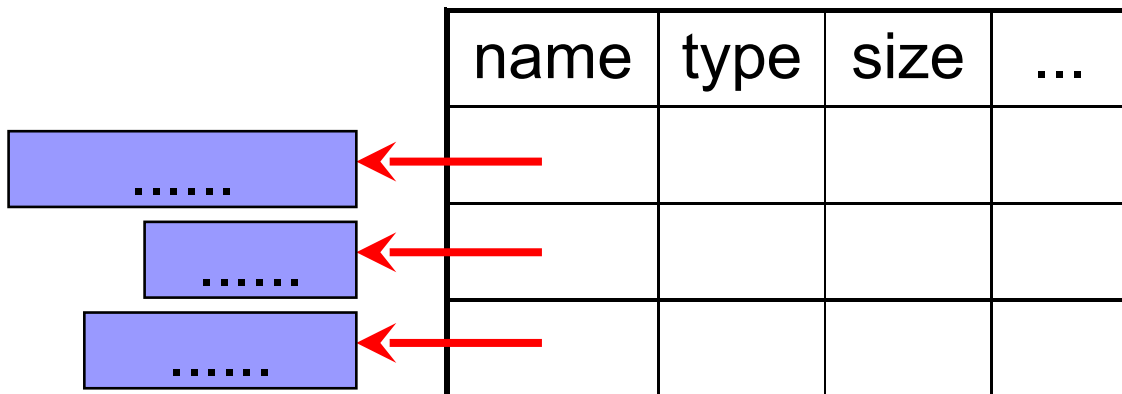
- `if index < 100 ...`



# 8.1 符号表的组织与作用

- 按名字的不同种属建立多张符号表，如常数表、变量名表、过程名表、...
- 符号表的组织方式
  - 安排各项各栏的存储单元为固定长度
  - 用间接方式安排各栏存储单元

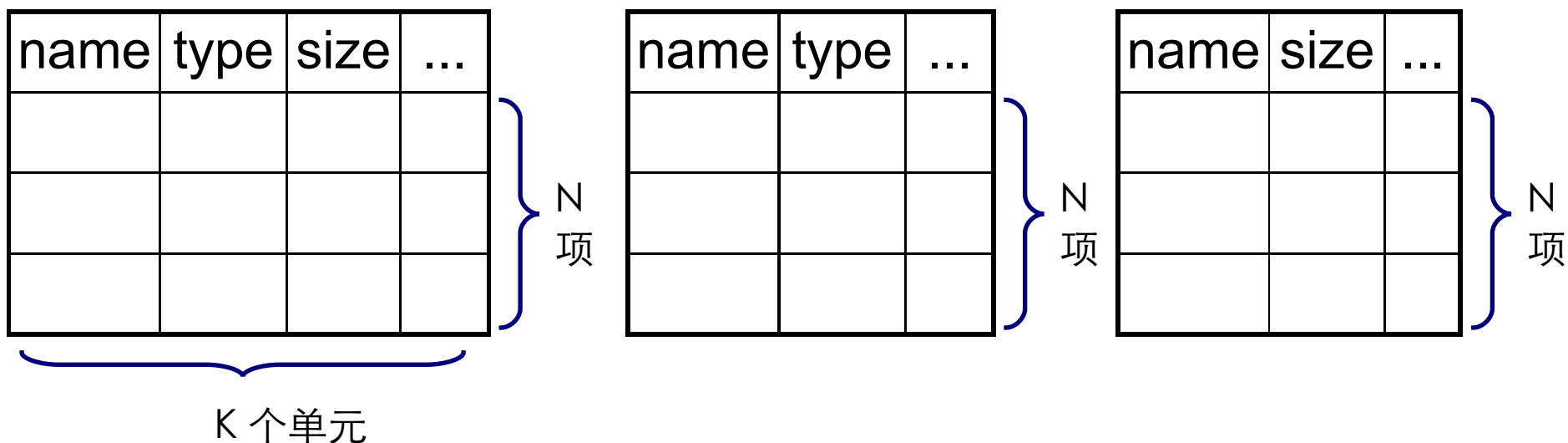
name	type	size	...



# 8.1 符号表的组织与作用

## ■ 符号表的存放次序

- 把每一项置于连续  $K$  存储单元中，构成一张  $K*N$  的表
- 把整个符号表分成  $m$  个子表，如  $T_1, T_2, \dots, T_m$ ，每个子表含有  $N$  项。



# PASCAL 程序段示例

```
PROCEDURE INCWAP(M , N:INTEGER);  
LABEL START;  
VAR  
  K:INTEGER;  
BEGIN  
START:  
  K:=M+1;  
  M:=N+4;  
  N:=K;  
END.
```

```

PROCEDURE
  INCWAP(M, N:INTEGER);
LABEL START;
VAR
  K:INTEGER;
BEGIN
START:
  K:=M+1;
  M:=N+4;
  N:=K;
END.

```

符号名表

	NAME	INFORMATION
(1)	M	形式参数，整型，值参数
(2)	N	形式参数，整型，值参数
(3)	K	整型，变量

```
PROCEDURE
  INCWAP(M, N:INTEGER);
LABEL START;
VAR
  K:INTEGER;
BEGIN
START:
  K:=M+1;
  M:=N+4;
  N:=K;
END.
```

常数表

	值 (VALUE)
(1)	1
(2)	4

```

PROCEDURE
  INCWAP(M, N:INTEGER);
LABEL START;
VAR
  K:INTEGER;
BEGIN
START:
  K:=M+1;
  M:=N+4;
  N:=K;
END.

```

四元式表

	OPR	OPN1	OPN2	RESULT
(1)	Link			
(2)	par	INCWAP	1	M
(3)	par	INCWAP	2	N
(4)	+	M	1	K
(5)	+	N	4	M
(6)	:=	K		N
(7)	return			

```

PROCEDURE
  INCWAP(M, N:INTEGER);
LABEL START;
VAR
  K:INTEGER;
BEGIN
START:
  K:=M+1;
  M:=N+4;
  N:=K;
END.

```

四元式表

	OPR	OPN1	OPN2	RESULT
(1)	Link			
(2)	par	INCWAP	1	M
(3)	par	INCWAP	2	N
(4)	+	M	1	K
(5)	+	N	4	M
(6)	:=	K		N
(7)	return			

入口名表

	NAME	INFORMATION
(1)	INCWAP	二目子程序，入口四元式：

(1)

PROCEDURE  
INCWAP(M, N:INTEGER); 四元式表

LABEL START;

VAR

K:INTEGER;

BEGIN

START:

K:=M+1;

M:=N+4;

N:=K;

END.

	OPR	OPN1	OPN2	RESULT
(1)	Link			
(2)	par	INCWAP	1	M
(3)	par	INCWAP	2	N
(4)	+	M	1	K
(5)	+	N	4	M
(6)	:=	K		N
(7)	return			

标号表

	NAME	INFORMATION
(1)	START	四元式 : (4)



# 第八章 符号表

- 符号表的组织与作用
- 整理和查找
- 符号表的内容
- 名字的作用范围

## 8.2 整理和查找

- 线性查找
- 二分查找
- 杂凑查找 (HASH 技术)

# 线性查找

- 按关键字出现的顺序填写各项

- 填表快，查找慢。

- 结构简单，节省空间，效率低，查找时间复杂度： $O(n)$

- 改进：自适应线性表

name	type	size	...



# 二分查找

- 表格中的项按名字的“大小”顺序整理排列
- 填表慢，查找快。
- 查找时间复杂度： $O(\log_2 n)$
- 改进：组织成二叉树

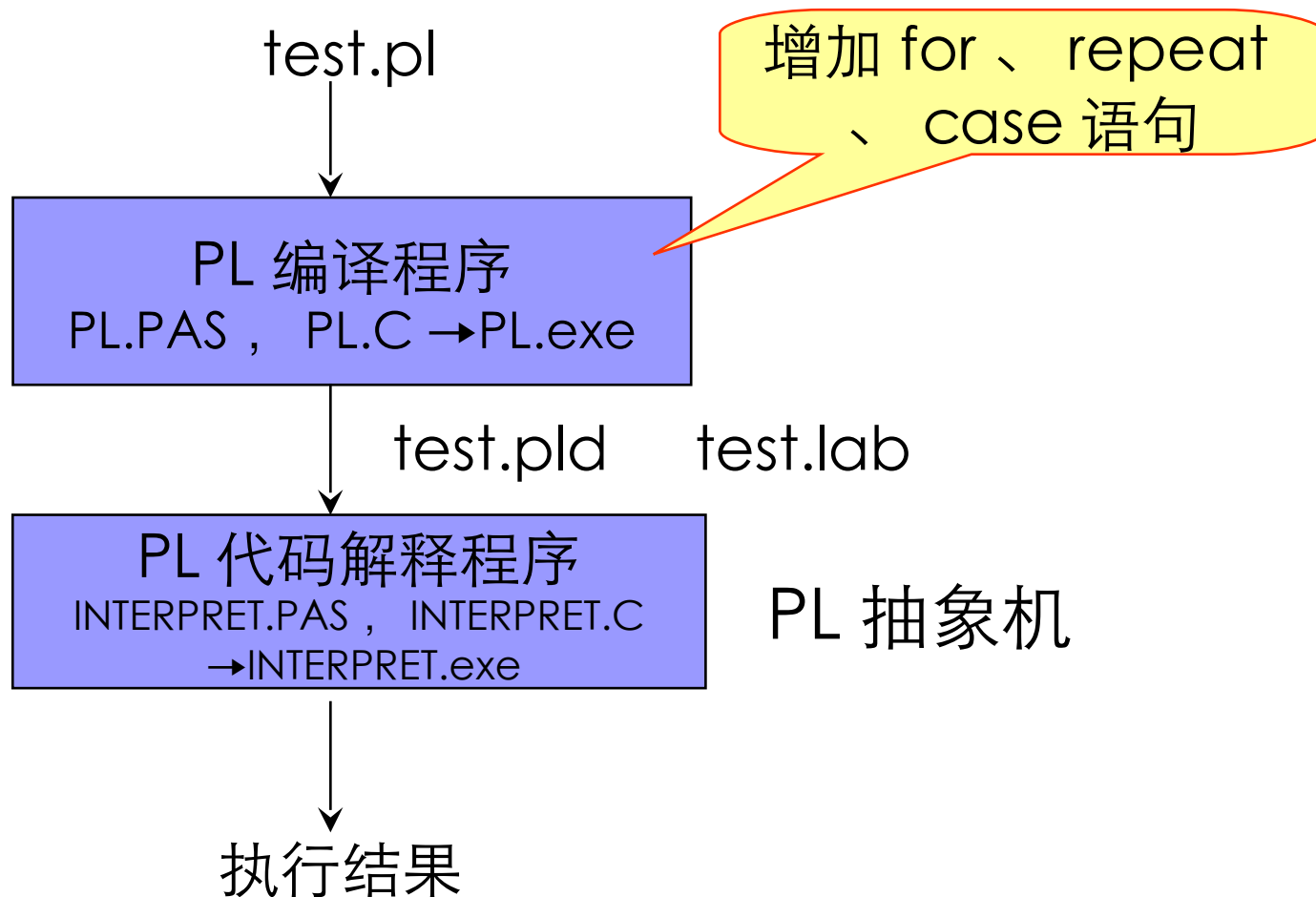
name	type	size	...

} n 项

# 杂凑查找 (HASH 技术)

- 杂凑函数  $H(\text{SYM})$  :  $0 \sim N-1$ 
  - $N$  : 符号表的项数
- 要求
  - 计算简单高效
  - 函数值分布均匀
- 填表快, 查找快

# PL 编译器实习框架



# 要求

1. 扩充语句，如增加 **for** 语句、**repeat** 语句、**case** 语句等
2. 增加**函数**的说明和使用
3. 扩充**数据类型**，如实型、记录类型等
4. 扩充其它语言成分

注意：

1. 每个同学必须完成第 1 部分的扩充；
2. 第 2、3、4 部分的扩充为选作；
3. 扩充的语言成分的多少是验收评分的重要标准。

# 上机实习检查要求

## ■ 要求提供的材料

- 实践报告
- 源程序（含完整的工程文件）
- 一个或多个说明扩充功能的 PL 语言示例程序，要求这些程序能够覆盖所有的扩充功能

## ■ 上机检查的流程

- 提交实践报告，简要说明完成的功能
- 编译并运行 PL 编译和解释程序，显示“说明扩充功能的 PL 语言示例程序”的执行结果
- 回答老师的问题



# 小结

- 符号表的组织与作用
- 整理和查找
- PL 编译器实习

# 作业

- 预习 《编译原理大作业实习》
  - 第三节 PL 语法分析
    - 三、表格处理