授课内容

第一章　编译程序概述
第二章　PL/0编译程序的实现
第三章　文法和语言
第四章　词法分析
第五章　自顶向下语法分析方法
第六章　自底向上优先分析方法
第七章　LR分析方法
第八章　语法制导翻译和中间代码生成
第九章　符号表
第一〇章　代码优化
第一一章　代码生成

编译程序概述

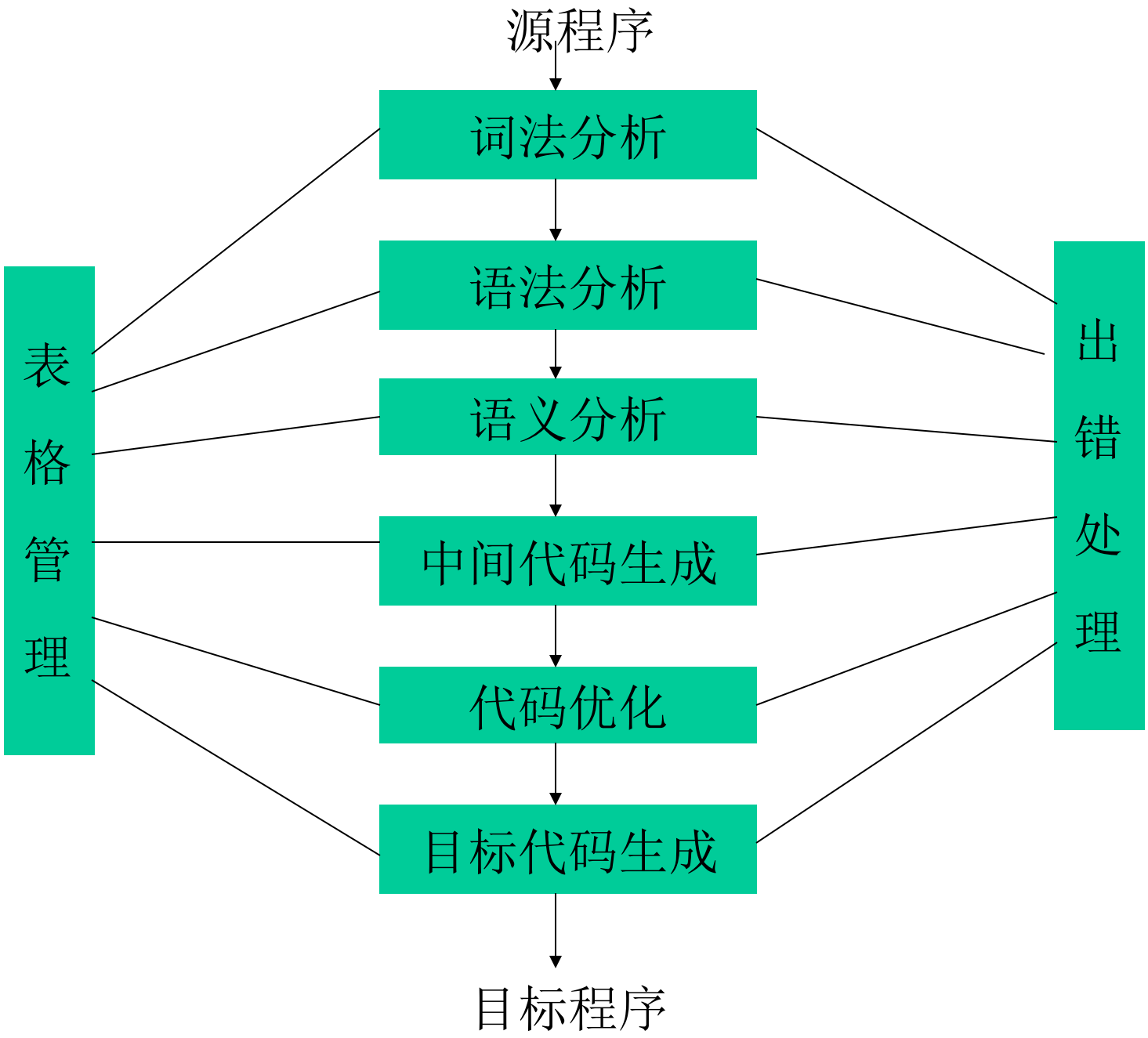
- 什么是编译程序/编译程序的作用是什么？
- 编译程序的逻辑过程是什么？

什么是编译程序

- 编译程序的功能：把高级语言程序翻译成等价的低级语言程序。
- 源程序——》编译程序——》目标程序
- 编译程序是现代计算机系统的基本组成部分。

编译过程概述

- 编译工作的基本过程是：
- 词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等6个阶段。
- 每个阶段都有表格管理和出错处理部分。


源程序
词法分析
语法分析
语义分析
中间代码生成
代码优化
目标代码生成
目标程序
表格管理
出错处理

授课内容

第一章　编译程序概述
第二章　PL/0编译程序的实现
第三章　文法和语言
第四章　词法分析
第五章　自顶向下语法分析方法
第六章　自底向上优先分析方法
第七章　LR分析方法
第八章　语法制导翻译和中间代码生成
第九章　符号表
第一〇章　代码优化
第一一章　代码生成

第2章　PL/0编译程序的实现

- 目的：以PL/0为实例,学习编译程序实现的基本步骤和相关技术，对高级语言编译程序的实现建立整体概念。

- PL/0语言是Pascal语言的一个子集，功能简单、结构清晰、可读性强，具有一般高级语言的必备部分
- PL/0编译程序是世界著名计算机科学家N.Wirth提出。
- PL/0语言编译程序采用以语法分析为核心、一遍扫描的编译方法，词法分析和代码生成作为独立的子程序供语法分析程序调用。

PL/0语言

- PL/0程序示例
- PL/0语法描述图
- PL/0语言文法的EBNF表示

PL/0程序示例

```
CONST A=10;
VAR   B,C;
PROCEDURE P;
    VAR D;
    PROCEDURE Q;
      VAR X;
      BEGIN
        READ(X);
        D:=X;
        WHILE X#0
        DO CALL P;
      END;
    BEGIN
      WRITE(D);
      CALL Q;
    END;
BEGIN
  CALL P;
END.
```

PL/0程序示例

```
CONST A=10;   (* 常量说明部分 *)
VAR   B,C;    (* 变量说明部分 *)
PROCEDURE P;     (* 过程说明部分 *)
    VAR D;
    PROCEDURE Q;
      VAR X;
      BEGIN
        READ(X);
        D:=X;
        WHILE X#0
        DO CALL P;
      END;
    BEGIN
      WRITE(D);
      CALL Q;
    END;
BEGIN
  CALL P;
END.
```

Q的过程体

p的过程体

主程序体

该程序执行的结果是什么？

PL/0语言

文法表示用语法图和EBNF描述。

扩充的巴科斯—瑙尔范式

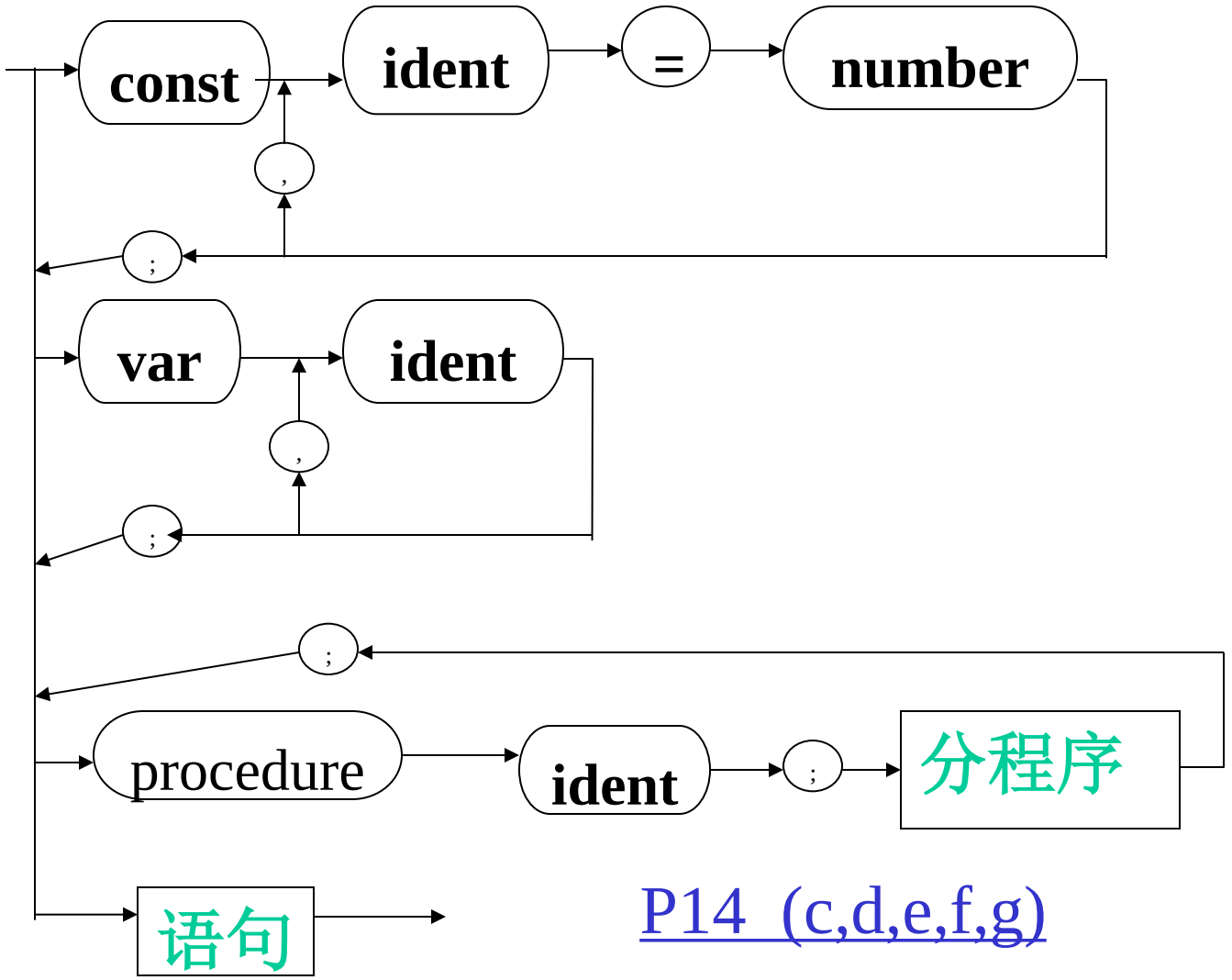
语法图：直观、易读

非终结符：长方形的中文

终结符：椭圆和圆圈中的英文

PL/0语言语法图描述

分程序



P14 (c,d,e,f,g)

指出以下程序代码哪些不合法

- Const a=1,b=2, c=3;
- Const d=1;
- Var f=6;
- Var g,h,t;
- Procedure M;
- begin
- g:=6;
- write(g);
- end;
- Procedure N;
- Write(5);
- call M.

指出以下程序代码哪些不合法

- Const a=1,b=2, c=3;
- Const d=1;
- Var f=6;
- Var g,h,t;
- Procedure M;
- begin
- g:=6;
- write(g);
- end;
- Procedure N;
- Write(5);
- call M.

PL/0语言

文法表示用EBNF描述。

扩充的巴科斯—瑙尔范式

1. BNF与EBNF的介绍：

❖BNF（BACKUS-NAUR FORM）是根据美国的John W. Backus与丹麦的Peter Naur来命名的，它从语法上描述程序设计语言的元语言。采用BNF就可说明哪些符号序列是对于某给定语言在语法上有效的程序

❖BNF引入的符号：

< >	用左右尖括号括起来的语法成分为非终结符
::=	定义为
|	或

❖EBNF引入的符号：

{ }	表示花括号内的语法成分可重复
[ ]	表示方括号内的语法成分为任选项
( )	表示圆括号内的成分优先

PL/0语言文法的EBNF表示

〈程序〉::=〈分程序〉.

〈分程序〉::= 〔〈常量说明部分〉〕〔〈变量说明部分〉〕〔〈过程说明部分〉〕〈语句〉

〈常量说明部分〉 :: =CONST〈常量定义〉{, 〈常量定义〉};

〈常量定义〉 :: =〈标识符〉=〈无符号整数〉

〈无符号整数〉 :: ＝〈数字〉{〈数字〉}

〈变量说明部分〉 :: =VAR〈标识符〉{, 〈标识符〉};

〈标识符〉 :: =〈字母〉{〈字母〉|〈数字〉}

P15

PL/0语言文法的EBNF表示

〈过程说明部分〉::=〈过程首部〉〈分程序〉{;〈过程说明部分〉};

〈过程首部〉::=PROCEDURE〈标识符〉;

〈语句〉::=〈赋值语句〉|〈条件语句〉|〈当型循环语句〉|〈过程调用语句〉|〈读语句〉|〈写语句〉|〈复合语句〉|〈空〉

〈赋值语句〉::=〈标识符〉:=〈表达式〉

〈复合语句〉::=BEGIN〈语句〉{;〈语句〉}END

〈条件〉::=〈表达式〉〈关系运算符〉〈表达式〉|ODD〈表达式〉

PL/0语言文法的EBNF表示

〈表达式〉::= [+|-]〈项〉{〈加法运算符〉〈项〉}

〈项〉::=〈因子〉{〈乘法运算符〉〈因子〉}

〈因子〉::=〈标识符〉|〈无符号整数〉|‘(’〈表达式〉‘)’

〈加法运算符〉::=+|-

〈乘法运算符〉::=*|/

〈关系运算符〉:: =#|〈|〈=|〉|〉=|=

PL/0语言文法的EBNF表示

〈条件语句〉::=IF〈条件〉THEN〈语句〉

〈过程调用语句〉::=CALL〈标识符〉

〈当型循环语句〉::= WHILE〈条件〉DO〈语句〉

〈读语句〉::=READ‘（’〈标识符〉{,〈标识符〉}‘）’

〈写语句〉::=WRITE‘（’〈表达式〉{,〈表达式〉}‘）’

〈字母〉::=a|b.....|x|y|z

〈数字〉::=0|1|2...|8|9

PL/0语言文法的EBNF表示

- 例：用EBNF描述<整数>的定义
- <整数>::=[+|-]<数字>{<数字>}
  <数字>::=0|1|2|3|4|5|6|7|8|9
-
- <整数>::=[+|-]<非零数字>{<数字>}
  <非零数字>::=1|2|3|4|5|6|7|8|9
  <数字>::=0|<非零数字>

- <整数>::=[+|-]<非零数字>{<数字>}|0
  <非零数字>::=1|2|3|4|5|6|7|8|9
  <数字>::=0|<非零数字>

PL/0语言是PASCAL语言的子集

同PASCAL

作用域规则（内层可引用包围它的外层定义的标识符）

过程可嵌套定义，可递归调用

子集

- 数据类型,只有整型(无符号整数)
- 数据结构 ,只有变量和常量
- 语句种类

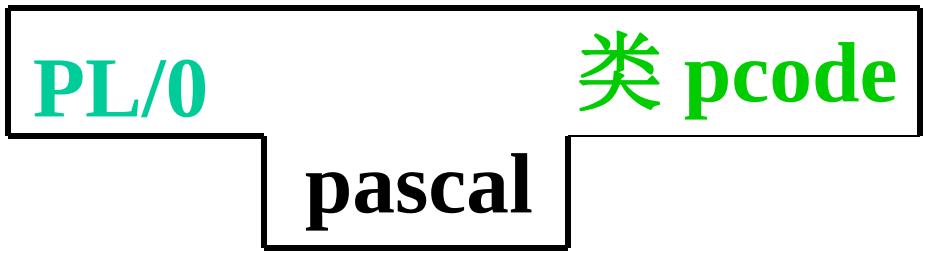
- 数字最多为14位
- 标识符的有效长度是10
- 过程最多可嵌套三层

PL/0编译程序

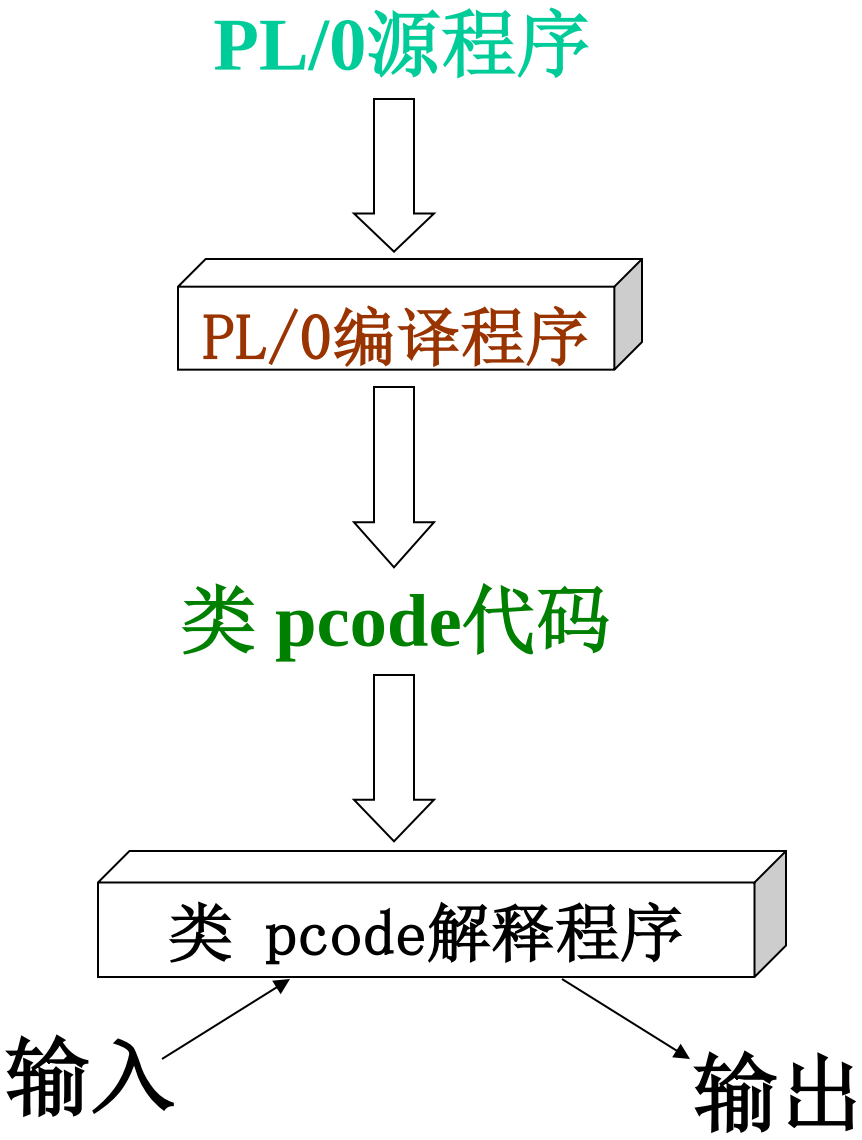
源语言(PL/0)

目标语言(类 pcode)

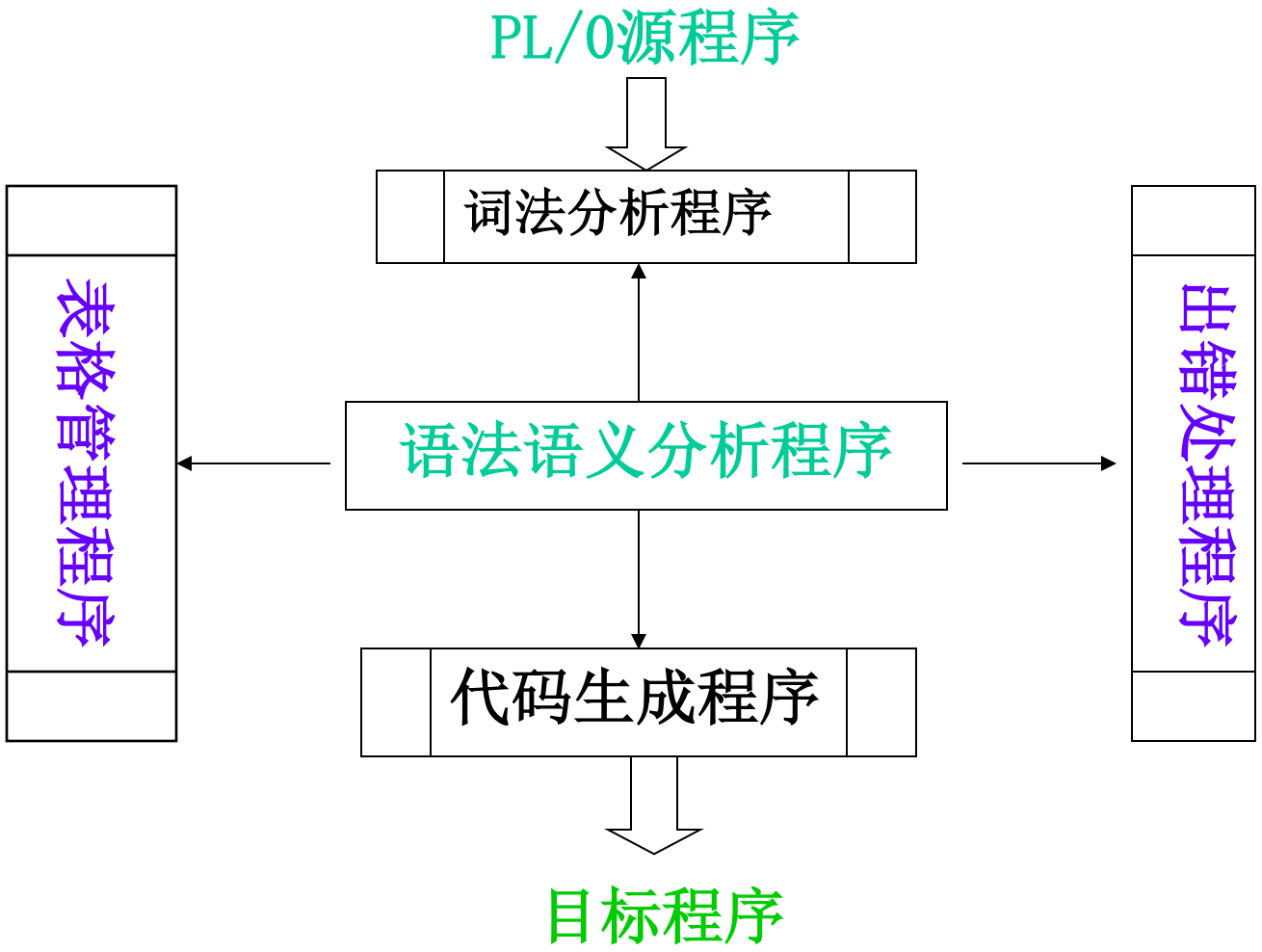
实现语言（pascal)



PL/0编译系统的结构框架



PL/0编译程序的结构



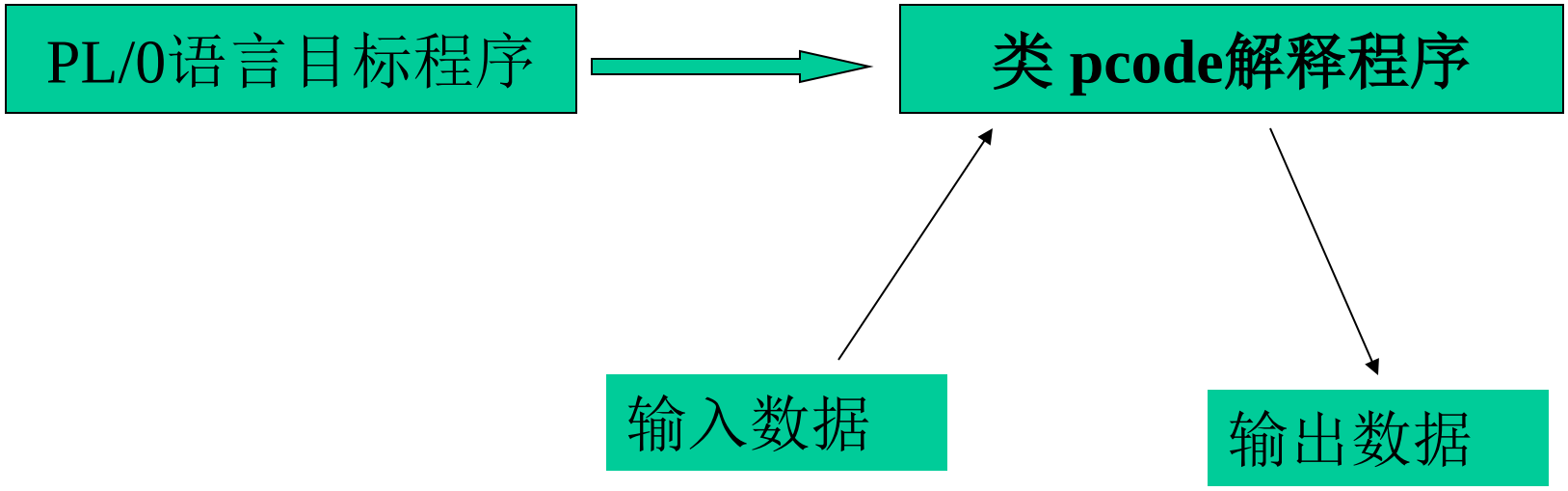
PL/0编译程序的总体设计

- 其编译过程采用一趟扫描方式
- 以语法、语义分析程序为核心

  词法分析程序和代码生成程序都作为一个过程，当语法分析需要读单词时就调用词法分析程序，而当语法、语义分析正确，需要生成相应的目标代码时，则调用代码生成程序。
- 表格管理程序实现变量，常量和过程标识符的信息的登录与查找。
- 出错处理程序，对词法和语法、语义分析遇到的错误给出在源程序中出错的位置和与错误 性质有关的编号，并进行错误恢复。

PL/0编译程序的总体设计

当源程序编译正确时，开始自动调用解释执行程序，对目标代码进行解释，并进行输入和输出。



PL/0编译程序的结构

- PL/0编译程序的组成：

  PL/0程序有18个过程或函数组成。（见表2.1）

- 过程、函数嵌套层次图（见图2.3）
- PL/0编译程序总流程图（见图2.4）

PL/0编译程序的词法分析

PL/0的词法分析程序Getsym是一个独立的过程

Getsym功能是 ：对单词进行分类分析

1、Getsym流程图（见图2.5）

2、流程图分析

3、Getch程序流程图（见图2.6）

PL/0编译程序的语法分析

PL/0的语法分析

1、方法：采用了自顶向下的递归子程序法

2、语法依据：文法的规则

3、任务：分析在词法分析基础上的单词符号结构是否符合给定的文法规则

4、分析步骤（粗）

开始→非终结符→调用相应处理程序→进入语法单元→ 沿语法图箭头方向分析→遇终结符→判断单词与图中匹配吗？ →匹配→单元翻译程序→下一个分析→程序结束符 ‘.’

5、PL/0语法调用关系图（见图2.7）

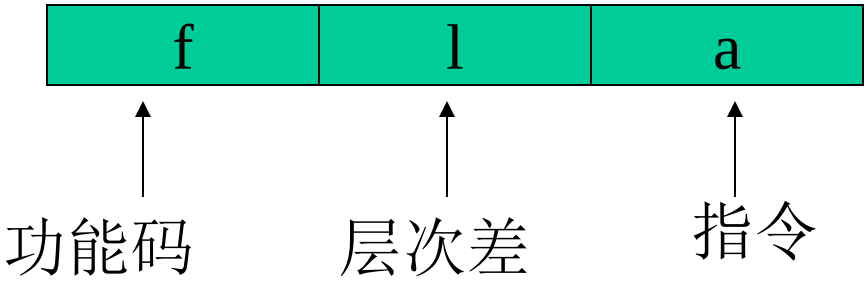
6、Block过程的流程图（见图2.8）

Block过程是处理说明部分和程序体部分

五、PL/0目标代码生成

1、PCODE代码

PCODE代码是PL/0所产生的目标代码。格式为：



变量、过程被调用的分程序与说明该变量、过程的分程序之间的层次差

2、源程序和目标程序清单

六、PL/0语法错误处理 （了解）

方法：
- 指出错误位置，并校正，如逗号、括号等
- 错误局限在语法当中。

七、PL/0存储分配

更详细的解释请参考教材

课后作业：阅读PL/0编译程序的源代码，找出语句处理、表达式、项、因子的语法分析程序片段，试画出各自类似图2.5和图2.8的流程图。