《编译原理》
(Compiling Principle)

- 主讲：王慧娇
- 办公室：金鸡岭3301-1
- 电话：13978321977
- QQ：248886622
- Email：whj7667@qq.com

- 答疑地点：5507
- 辅导时间：周三1,2节

课程简介

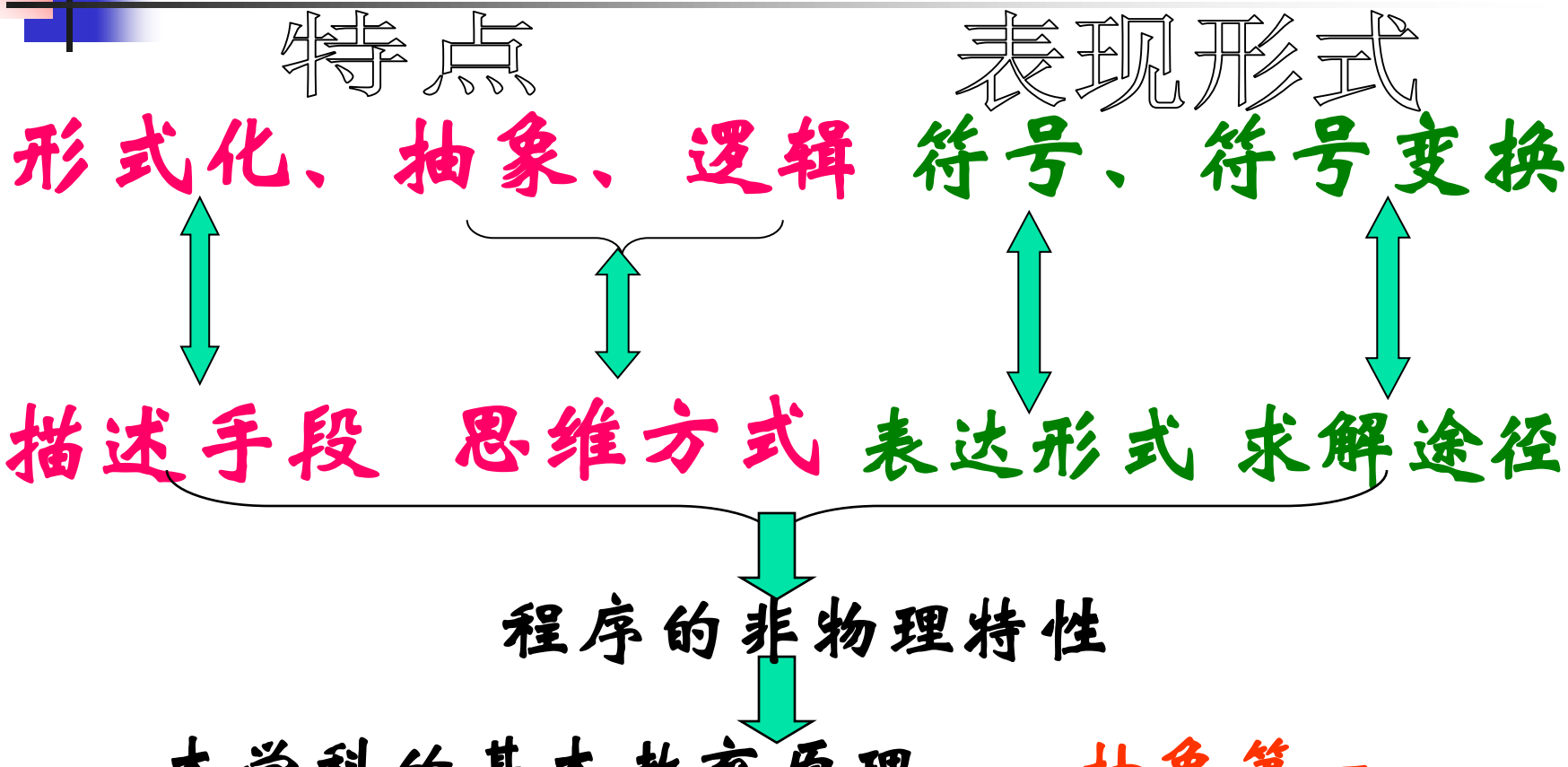
课程内容

- 介绍编译器构造的一般原理和基本实现方法
- 包括的理论知识：形式语言和自动机理论、语法制导的定义和属性文法等
- 强调形式描述技术和自动生成技术
- 强调对编译原理和技术的宏观理解，不把注意力分散到枝节算法，不偏向于任何源语言或目标机器

教学目的——计算学科的定义

对信息描述和变换算法的系统研究，主要包括它们的理论、分析、效率、实现和应用。计算学科的根本问题是什么能被（有效地）且如何自动化，讨论问题求解的“能行性”。

教学目的——学科基本特征


特点
表现形式
形式化、抽象、逻辑
符号、符号变换
描述手段
思维方式
表达形式
求解途径
程序的非物理特性
本学科的基本教育原理——抽象第一

教学目的——计算学科本科生专业能力构成

- “计算思维能力”——模型化、抽象思维能力、逻辑思维能力
- 算法设计与分析能力
- 程序设计与实现能力
- 计算机系统的认知、分析、设计和应用能力
- 编译原理的授课涉及上述四种能力的培养

教学目的——《编译原理》是一门非常好的课程

- Alfred V.Aho：编写编译器的原理和技术具有十分普遍的意义，以至于在每个计算机科学家的研究生涯中，本书中的原理和技术都会反复用到
- 涉及的是一个比较适当的抽象层面上的数据变换（既抽象，又实际）
- 一些具体的表示和变换算法
- “自顶向下”和“自底向上”的系统设计方法（思想、方法、实现全方位讨论）
- 一个相当规模的系统的设计（含总体结构）
- 结论：是计算机专业培养学生解决复杂工程问题的最恰当、有效的知识载体之一

教学要求

- 掌握编译程序总体结构
- 在系统级上认识算法、系统的设计
  - 具有把握系统的能力
- 学习有关的原理、实现方法和技术，了解计算学科的基本方法、思想
  - 掌握典型方法。“在每一个计算机科技工作者的职业生涯中，这些原理和技术都被反复用到。”
- 兼顾语言的描述方法、设计、应用——形式化
- 进一步培养“计算机思维能力”
  - 程序的非物理性质

学习方法

- 勤于思考
  - 博览、多思（学而不思则罔、思而不学则怠；书由厚到薄、由薄到厚）、常实践
  - 思考由怀疑和答案组成。怀疑是智慧的大门，知道得越多，就越会发问，而问题就越多。发问使人进步——学问
- 强化基础
  - 在独立思考之前，必须先有基础知识。所谓“获得基础知识”并不是形式上读过某门课程，而是将学过的东西完全弄懂——一定要上课
  - 理论与实践的结合能力

学习方法

- 应对困难
  - 不畏惧困难
  - 从教训到经验——亲身体验
  - 要实践（作业、实验），加深理解
- 学习是一个过程
  - 上课、读书、复习、做作业、讨论、做实验、自己编程序、上机调试排错.....是绝对必要的
- 辅导答疑
  - 教师是最宝贵的资源...
  - 自己要思考，以获得最大收获：习题、问题

希望...

- 1.掌握“编译原理”中的基本概念、基本理论、基本方法
- 2.在系统层面上再认识程序和算法
- 3.提升计算机问题的求解水平
- 4.增强系统能力
- 5.体验实现自动计算的乐趣

学时与参考教材

- 学时：48学时
- 教材：王生原等，《编译原理》，清华大学出版社
- 线上课程：国家精品课程，哈尔滨工业大学，陈鄞开设的《编译原理》，课程网址：https://www.icourse163.org/course/HIT-1002123007
- 参考教材：
  - 1、Alfred Aho．《编译原理》，赵建华等译，机械工业出版社，2003.8.(原版-邮电出版社)
  - 2、Kenneth C. Louden，《编译原理及实践》，冯博琴等译，机械工业出版社，2001.2
  - 3、蒋宗礼等，《编译原理》，高等教育出版社，2010.2.

学时与参考教材

- 4、何炎祥等，《编译原理》，华中理工大学出版社，2000.10.
- 5、陈意云，张昱，《编译原理习题精选与解析》，高等教育出版社，2014.9.
- 6、 Alfred Aho 等，《编译原理（本科教学版）》（第2版），机械工业出版社，2008.12
- 7、何炎祥等，《可信编译构造理论与关键技术》,科学出版社，2012.6
- ……

学时与参考教材


计 算 机 科 学 丛 书
第2版
编译领域里程碑式的经典教材——龙书，20年后再出新版！
这是一个绵延30年的故事，这是一部关于龙书的传奇……
编译原理
(美) Alfred V. Aho 哥伦比亚大学 Monica S. Lam 斯坦福大学 Ravi Sethi Avaya实验室 Jeffrey D. Ullman 斯坦福大学 著 赵建华 郑滔 戴新宇 南京大学 译
等待龙书二
花费了近10年的时间，
等待龙书三
花费了20年的时间，
那龙书四呢？
一部龙书的传奇还将继续下去……
Compilers
Principles, Techniques, & Tools
Second Edition
Alfred V. Aho
Monica S. Lam
Ravi Sethi
Jeffrey D. Ullman
1977年，Alfred V. Aho和Jeffrey D. Ullman合作出版了《Principles of Compiler Design》，封面是一位骑士和一只恐龙，那恐龙是绿色的，因此被称为龙书或绿龙书。
1986年，原来的两位作者加上Ravi Sethi，升级了前一本书，书名改为《Compilers: Principles, Techniques and Tools》，封面依然沿用骑士和恐龙，那恐龙是红色的，因此被称为龙书二或者红龙书。
又过了一个9年又一个9年，编译领域的巨无霸——龙书始终都没有升级。
终于在2006年底，龙书升级了，作者又增加了Monica S. Lam，名字与龙书二相同，封面依然沿用恐龙和武士的设计，这次的龙是紫色的，因此被称为龙书三或者紫龙书。
Compilers
Principles, Techniques and Tools
Second Edition
机械工业出版社
China Machine Press
计 算 机 科 学 丛 书
第2版
本科教学版
编译原理
(美) Alfred V. Aho 哥伦比亚大学 Monica S. Lam 斯坦福大学 Ravi Sethi Avaya实验室 Jeffrey D. Ullman 斯坦福大学 著 赵建华 郑滔 戴新宇 南京大学 译
Compilers
Principles, Techniques, & Tools
Second Edition
Alfred V. Aho
Monica S. Lam
Ravi Sethi
Jeffrey D. Ullman
Compilers
Principles, Techniques and Tools
Second Edition
机械工业出版社
China Machine Press

主要内容

- 编译系统及其设计概述(总体结构、设计方法——2)
- 语言与文法(文法、推导、归约、分类、语法树——6)
- 词法分析(词法分析、正规式与正规文法、DFA状态转换图——10)
- 语法分析(自顶向下：LL(1)、递归子程序；自底向上：算符优先、LR类分析法——16)
- 语义分析(属性文法、各种语句的语法制导翻译——10)
- 运行环境(存储分配、符号表管理——4)

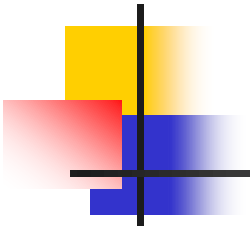
成绩评定

- 考试必备条件
  - 按照要求完成指定的习题
  - 按照要求完成指定的测试
  - 严格按照考勤要求的比例
- 成绩
  - 平时30%
  - 期末70%

成绩评定

平时成绩

评价项目	评价子项目	评价要求
平时成绩	作业	按正确程度打分
	小测	对某个具体算法相关知识点进行考核，按正确程度打分
	阶段测验	对基本概念、原理或问题解决方案进行考核，按正确程度打分



第1章 绪论

- 1.1 编译程序概述
- 1.2 编译程序的组织
- 1.3 编译程序的生成
- 1.4 编译技术的应用

1.1 编译程序概述

- 1.1.1 程序设计语言
- 机器语言(Machine Language)
  - 0、1代码（1000 1110 1101 1000）
- 汇编语言(Assemble Language)
  - 0、1代码与指令助记符：更接近于计算机硬件指令系统的工作（MOV DS，AX）
- 高级语言(High Level Language)
  - 定义数据、描述算法（程序）
  - 如：C、FORTRAN、PASCAL、C++、JAVA、SQL(数据定义、数据操作)

1.1.2 翻译系统

- 请同学们翻译以下英文：
  - Location-based geographic routing is also attractive in wireless sensor networks due to its efficiency and scalability, and it is more energy-efficient for data forwarding on the cluster head backbone compared to traditional hop based methods.
  - Geographic routing algorithms have been studied in the context of wireless networks. Frey and Stojmenovic provide a good review of geographic and energy-aware routing algorithms for wireless sensor networks.

1.1.2 翻译系统

请同学们“翻译”以下“英文”：

```
main()
{ int s,a,b;
  scanf( “%d” ,&s);
   while( s>0 )
    { a=s%10;
      printf( “%d,” , a);
      s=s/10;
    }
    printf( “%f\n” , m/30);
}
```

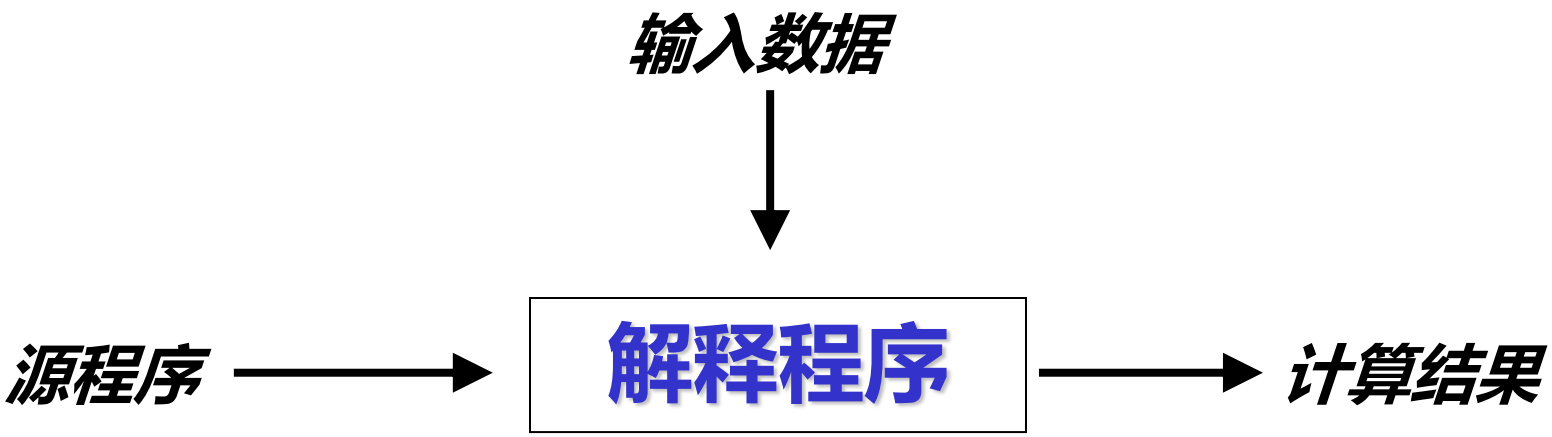
1.1.2 翻译系统

- 翻译程序(Translator)

将某一种语言描述的程序(源程序——Source Program)翻译成等价的另一种语言描述的程序(目标程序——Object Program)的程序。

1.1.2 翻译系统

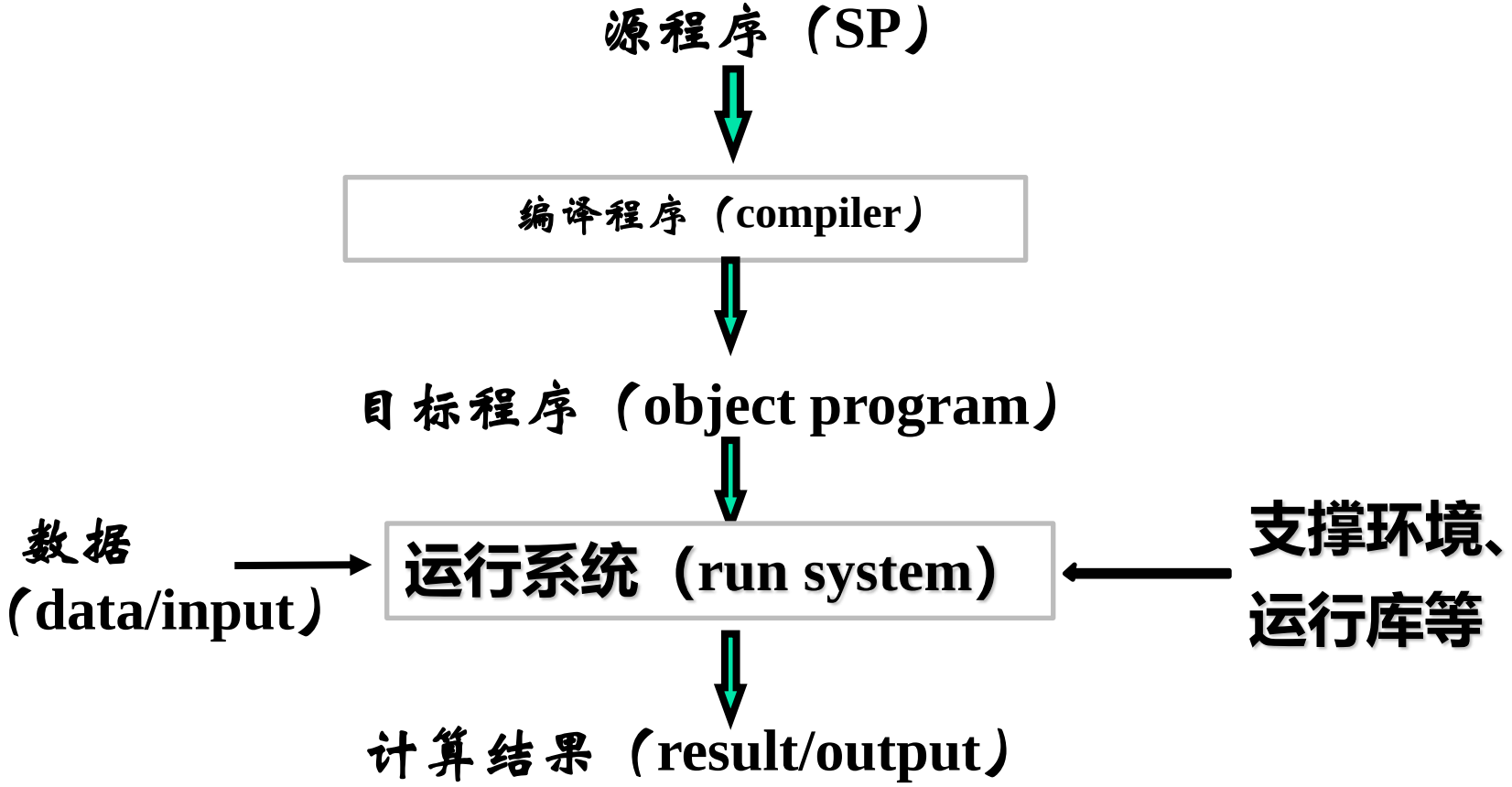
- 解释程序(Interpreter)（教材P7）
  - 口译与笔译（单句提交与整篇提交）



1.1.2 翻译系统

- 编译程序(Compiler)
  - 高级语言程序（源）→汇编/机器语言程序（目标）

1.1.2 编译系统



编译系统(Compiling System)

编译系统=编译程序+运行系统

1.1.2 翻译系统

- 其它：
  - 汇编程序(Assembler)、交叉汇编程序(Cross Assembler)、反汇编程序（Disassembler)
  - 交叉编译程序（Cross Compiler)
  - 可变目标编译程序（Retargetable Compiler)
  - 并行编译程序（Parallelizing Compiler）
  - ……

1.2 编译程序结构与编译过程

1.2.1 编译过程

习惯上是将编译过程划分为5个基本阶段：

词法分析程序：状态图，手工编写程序

语法分析程序：递归子程序法，编写程序

语义分析和代码生成：语法制导的翻译（属性翻译文法）

完成了一个翻译过程

1.2 编译过程

- 一个微型PASCAL语言的定义
  - 1）PROGRAM语句；
  - 2）说明语句；
  - 3）BEGIN-END语句；
  - 4）赋值语句；

程序1-1 一个PASCAL源程序source

```
PROGRAM  source;
{This little source program is used to
     illustrate compiling procedure }
VAR x,y,z:integer;
     a:integer;
BEGIN
{ This program has only 4 statement }
x:=23+5;
z:=x DIV -3;
y:=z+18*3;
a:=x+(y-2) DIV 4;
END.
```

1.2.1 编译过程
1.词法分析程序

- 词法分析器 (Lexical Analyzer)又叫做扫描器(Scanner)，完成词法分析
- 功能：词法分析器从左到右扫描源程序（字符串），并将其转换成单词(记号—Token)串；同时查词法错误，进行标识符登记——符号表管理。
- 输入：字符串
- 输出：(类别码，属性值)——序对
  - 属性值——token的机内表示

1.2.1 编译过程

1. 词法分析程序

程序1-1 一个PASCAL源程序source

```
PROGRAM  source;
{This little source program is used to
     illustrate compiling procedure }
VAR x,y,z:integer;
     a:integer;
BEGIN
{ This program has only 4 statement }
x:=23+5;
z:=x DIV -3;
y:=z+18*3;
a:=x+(y-2) DIV 4;
END.
```

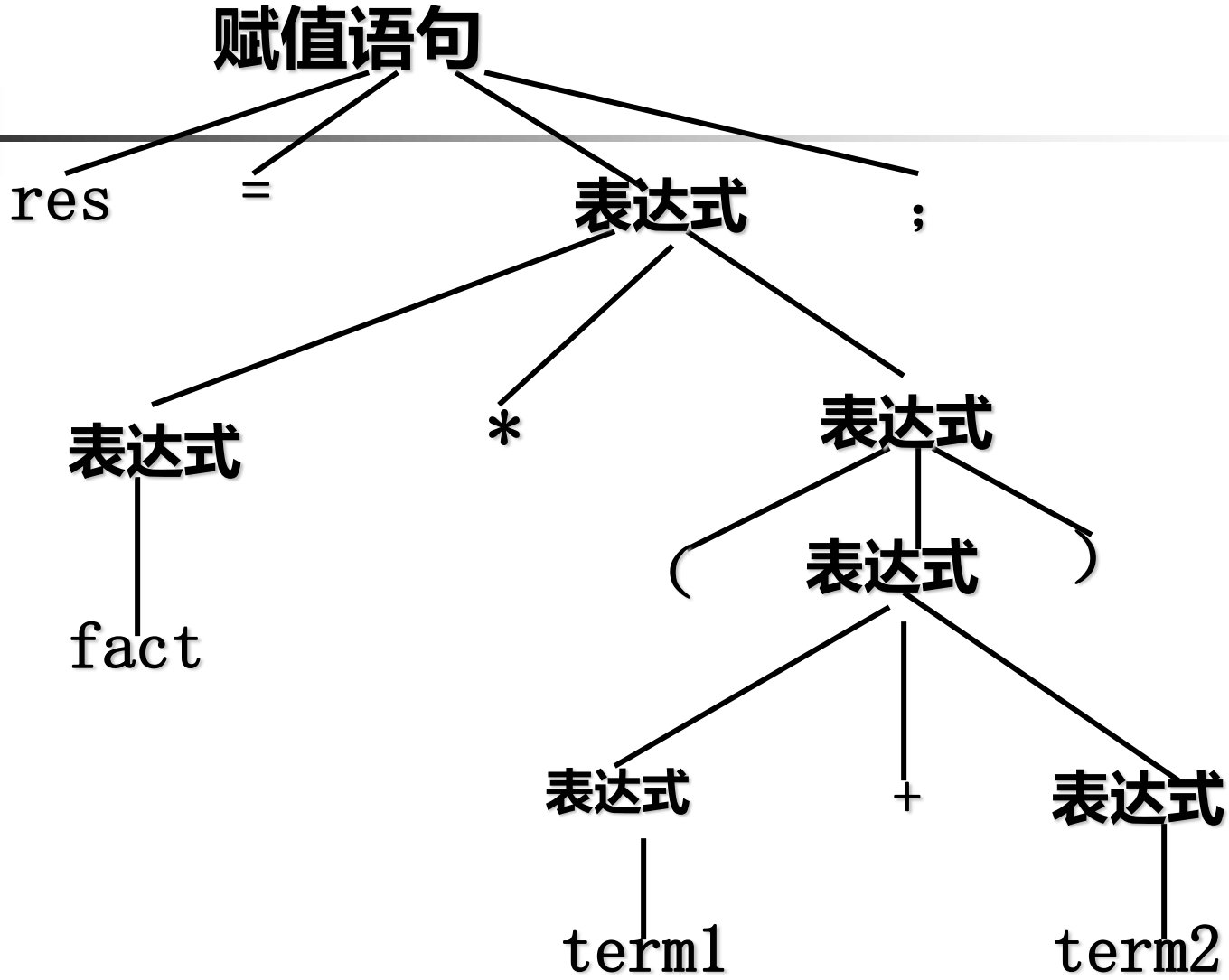
```
# PROGRAM # source
# ; # VAR # x # , # y # , #
z # : # integer # ; # a # :
# integer # ; # BEGIN # x
# := # 23 # + # 5 # ; # z
# := # x # DIV # - # 3 # ;
# y # := # z # + # 18 # *
# 3 # ; # a # := # x # + #
( # y # - # 2 # ) # DIV # 4
# ; # END # . #
```

1.2.1 编译过程
2. 语法分析

- 语法分析器(Syntax Analyzer，又叫Parser)完成语法分析
- 功能：Parser实现“组词成句”，构造分析树，指出语法错误，指导翻译
- 输入：Token序列
- 输出：语法成分

2. 语法分析

res=fact*(term1+term2);



1.2.1 编译过程
3. 语义分析

- 功能：分析由语法分析器给出的语法单位的语义
  - 获取标识符的属性：类型、作用域等
  - 语义检查：运算的合法性、取值范围等
  - 子程序的静态绑定：代码的相对地址
  - 变量的静态绑定：数据的相对地址

4. 中间代码生成

- 中间代码(intermediate Code)
- 例:$id_1+id_2*id_3$

后缀表示(逆波兰Reverse Polish Notation)

$id_1id_2id_3 * +$

前缀表示(波兰Polish Notation)

$+ id_1*id_2id_3$

四元组表示

(三地址码)

1 $(*, id_1, id_2, T_1)$

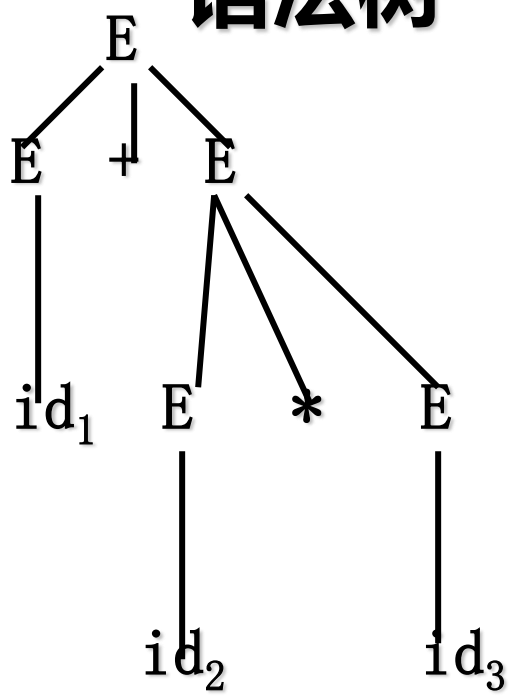
2 $(+, id_3, T_1, T_2)$

三元组表示

1 $(*, id_2, id_3)$

2 $(+, id_1, (1))$

语法树



波兰表示问题——Lukasiewicz(武卡谢维奇)

1929/1951年发明

- 中缀表示(Infix notation):(a+①b)*(-c+②d)+③e/f
- 波兰表示（Polish / Prefix / Parenthesis-free / Lukasiewicz notation）——也就是前缀表示
  - +③*+①a b+②@c d/ef 运算顺序从右向左
- 逆波兰表示(Reverse Polish / Suffix / Postfix notation) ——也就是后缀表示
  - a b +①c d +②*ef/+ ③ 运算顺序从左向右

4. 中间代码生成

- 中间代码的特点
  - 简单规范
  - 机器无关
  - 易于优化与转换

- 三地址码的另一种表示形式

$T_1=id_2*id_3$

$T_2=id_1*T_1$

4. 中间代码生成

程序1-1 一个PASCAL源程序source

```
PROGRAM  source;
{This little source program is used to
     illustrate compiling procedure }
VAR x,y,z:integer;
     a:integer;
BEGIN
{ This program has only 4 statement }
x:=23+5;
z:=x DIV -3;
y:=z+18*3;
a:=x+(y-2) DIV 4;
END.
```

(prologue,'source')
(add,'23','5',T)
(store,T, ,'x')
(div,'x','-3',T)
(store,T, ,'z')
(mult, '18','3',T)
(add,'z',T,T)
(store,T, ,'y')
(sub,'y','2',T)
(div,T,'4',T)
(add,'x',T,T)
(store,T, ,'a')
(epilogue)

5. 代码优化

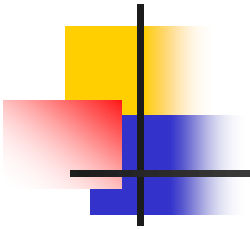
- 对中间代码的优化处理:对代码进行等价变换以求提高执行效率——提高运行速度和节省存储空间
  - 与机器无关的优化
  - 与机器有关的优化

5. 代码优化

- 与机器无关的优化
- 局部优化
  - 常量合并：常数运算在编译期间完成，如8+9*4
  - 公共子表达式的提取:基本块内
- 循环优化
  - 强度削减
  - 代码外提

5.代码优化
与机器有关的优化

- 寄存器的利用
  - 将常用量放入寄存器，以减少访问内存的次数
- 体系结构
  - MIMD、SIMD、SPMD、向量机、流水机
- 存储策略
  - 根据算法访存的要求安排：Cache、并行存储体系——减少访问冲突
- 任务划分
  - 按运行的算法即体系结构，划分子任务(MPMD)

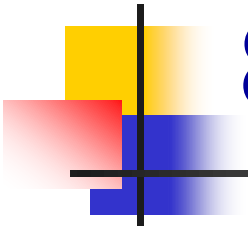


6. 目标代码生成

- 将中间代码转换成目标机上的机器指令代码或汇编代码

7. 表格管理

- 管理各种符号表（常数、标号、变量、过程、结构……），查、填（登记、查找）源程序中出现的符号和编译程序生成的符号，为编译的各个阶段提供信息。
  - 辅助语法检查、语义检查
  - 完成静态绑定、管理编译过程
- Hash表、链表等各种查、填表技术



8. 错误处理

- 进行各种错误的检查、报告、纠正，以及相应的续编译处理（如：错误的定位与局部化）
  - 词法：拼写……
  - 语法：语句结构、表达式结构……
  - 语义：类型不匹配……
  - ……

1.2 编译程序结构与编译过程
1.2.2 编译程序结构

典型的编译程序具有8个逻辑部分(请参阅P6图1.10)

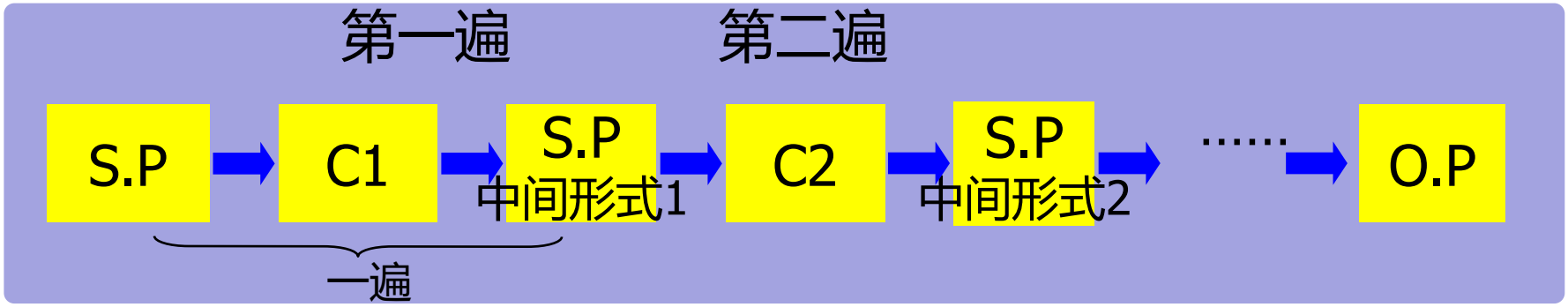
1.2.3 编译程序的组织

一、模块分类

- 分析：词法分析、语法分析、语义分析
- 综合：中间代码生成、代码优化、目标代码生成
- 辅助：符号表管理、出错处理
- 8项功能对应8个模块

二、 遍（PASS）

遍：对源程序（包括源程序中间形式）从头到尾扫描一次，并做有关的加工处理，生成新的源程序中间形式或目标程序，通常称之为一遍。



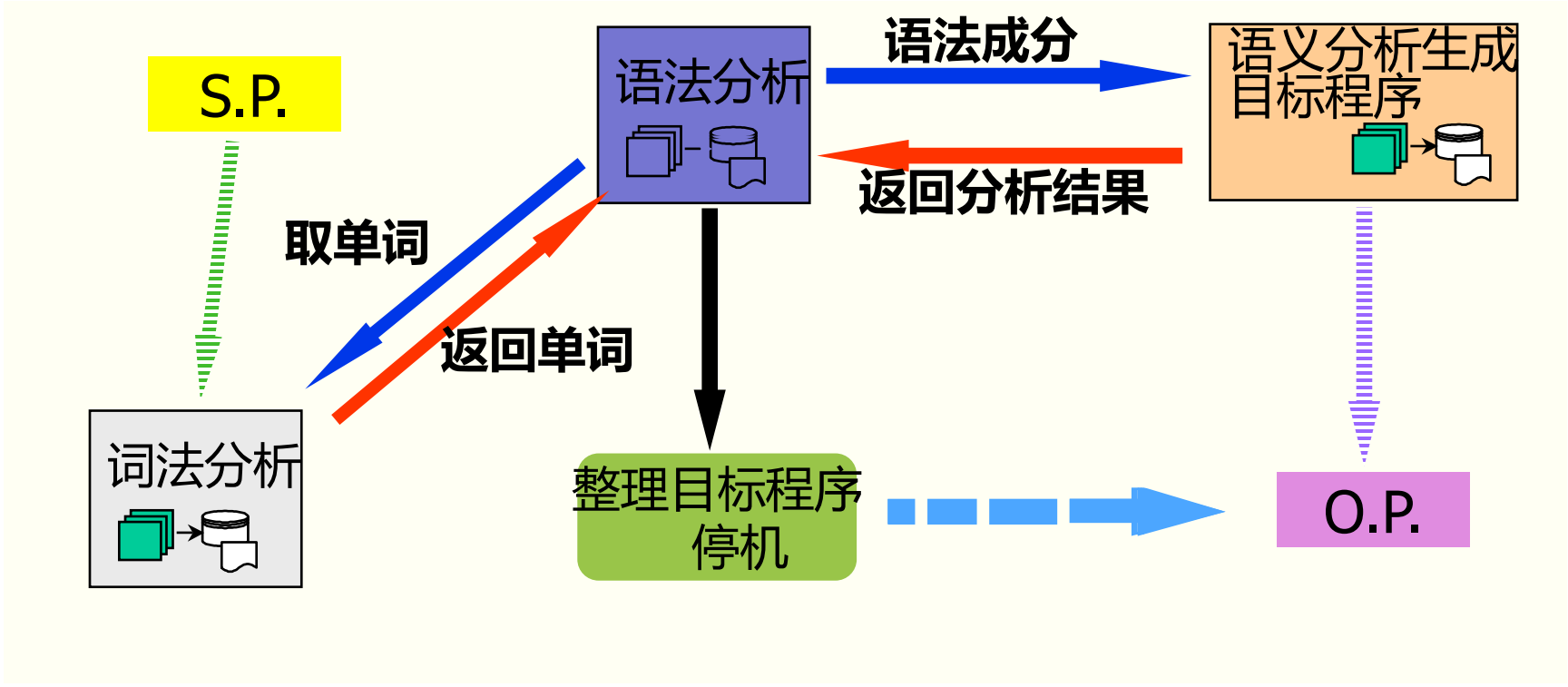
★ **要注意遍与基本阶段的区别**

五个基本阶段：是将源程序翻译为目标程序在逻辑上要完成的工作。

遍：是指完成上述5个基本阶段的工作，要经过几次扫描处理。

1.2.3 编译程序的组织

一遍扫描的示意图



1.2.3 编译程序的组织

- 一遍扫描与多遍扫描
- 多遍扫描的优点
  - 编译程序的结构清晰
  - 有利于进行细致和充分的代码优化
  - 易于采用覆盖技术，提高内存利用率

1.2.3 编译程序的组织

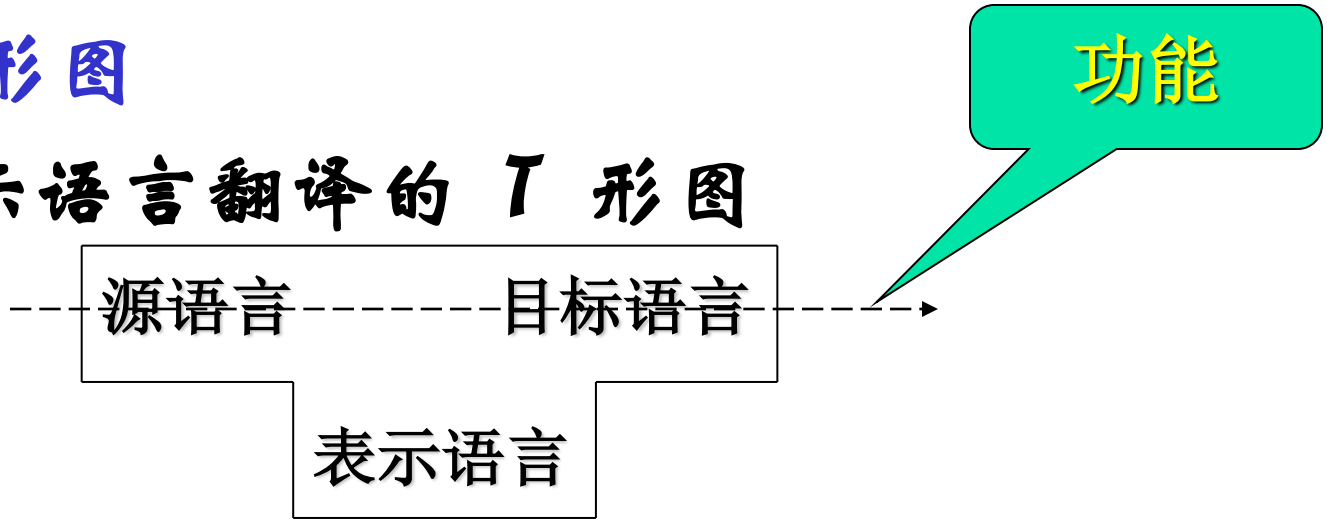
- 编译的前端与后端
- 前端
  - 与源语言有关、与目标机无关的部分
  - 词法分析、语法分析、语义分析与中间代码生成、与机器无关的代码优化
- 后端
  - 与目标机有关的部分
  - 与机器有关的代码优化、目标代码生成

1.3 编译程序的生成

- 设计目标
  - 目标程序小，执行速度快
  - 编译程序小，执行速度快
  - 诊断能力强，可靠性高
  - 可移植性，可扩充性
- 如何实现编译器？（用机器语言、高级语言编写？从头到尾编写？）

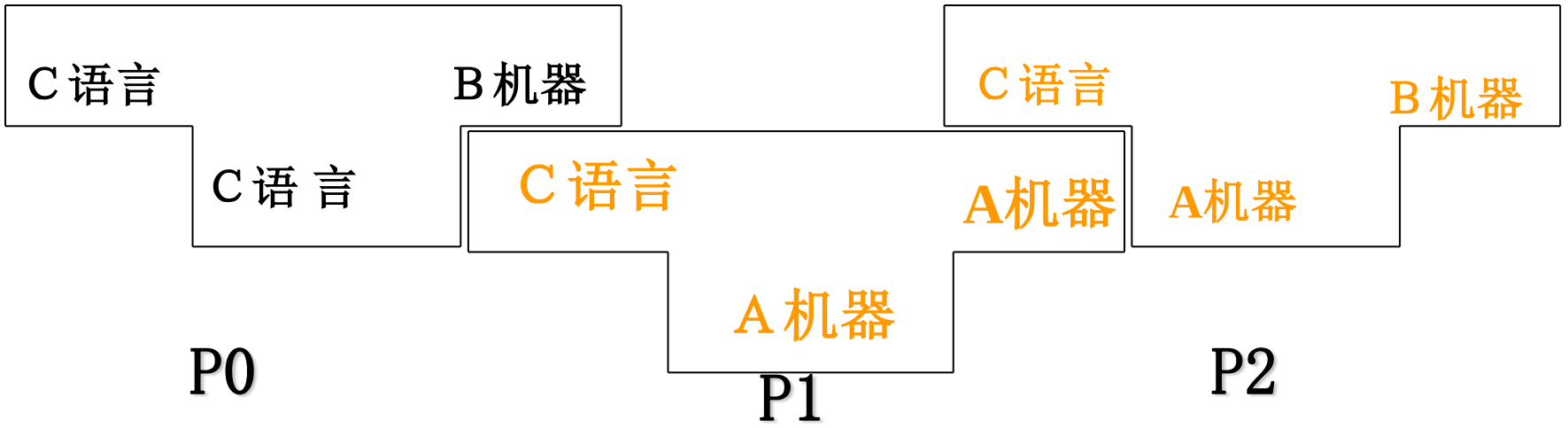
1.3 编译程序的生成

- T 形图
- 表示语言翻译的 T 形图



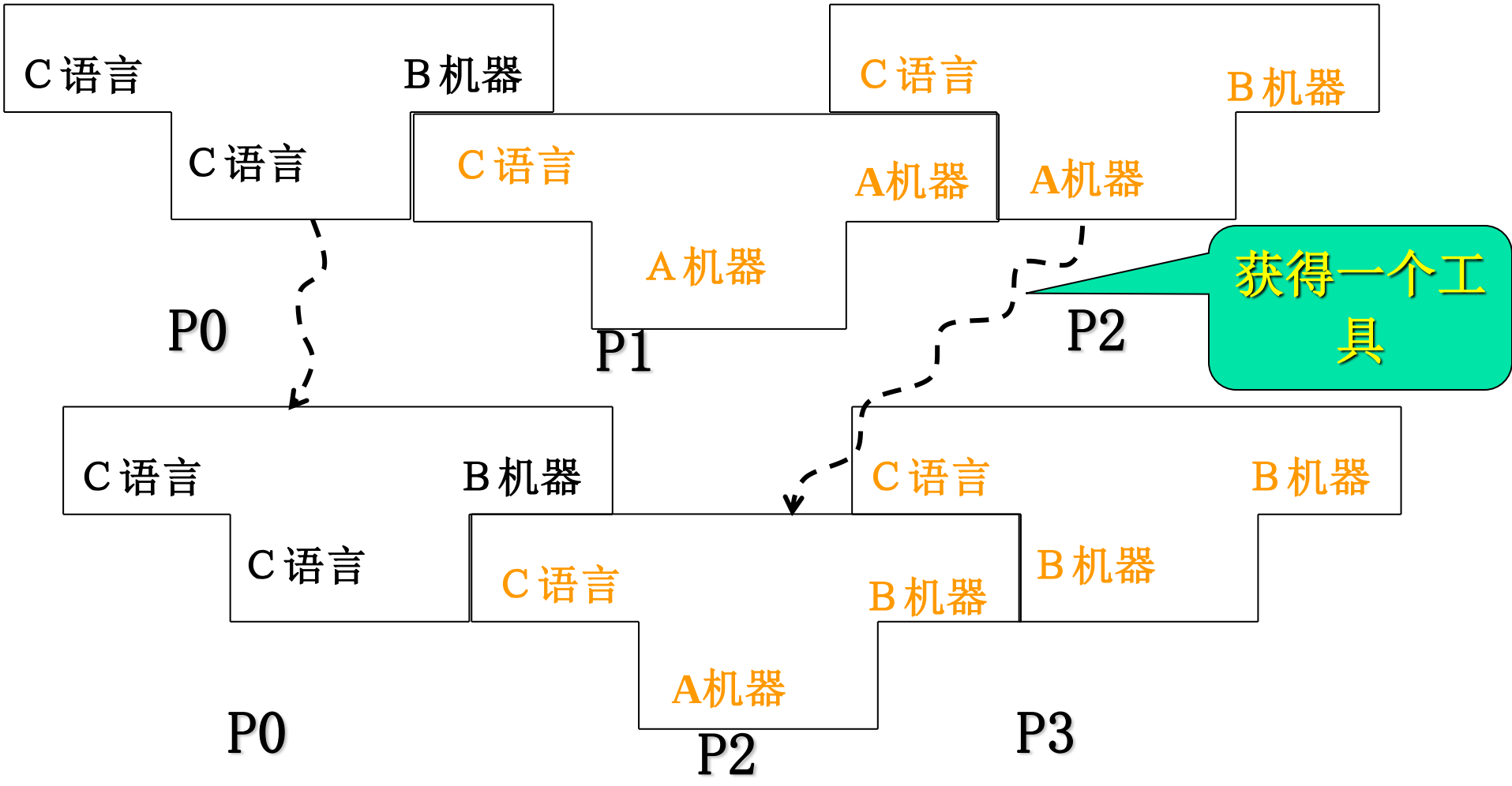
1）交叉编译(Cross Compiling)/移植

- 问题一：A机上有一个C语言编译器，是否可利用此编译器实现B机上的C语言编译器?
  - 条件：A 机有 C 语言的编译程序（P1)
  - 目的：实现 B 机的 C 语言的编译(P3)



1. (人)用C语言编制B机的C编译程序P0(C→B)

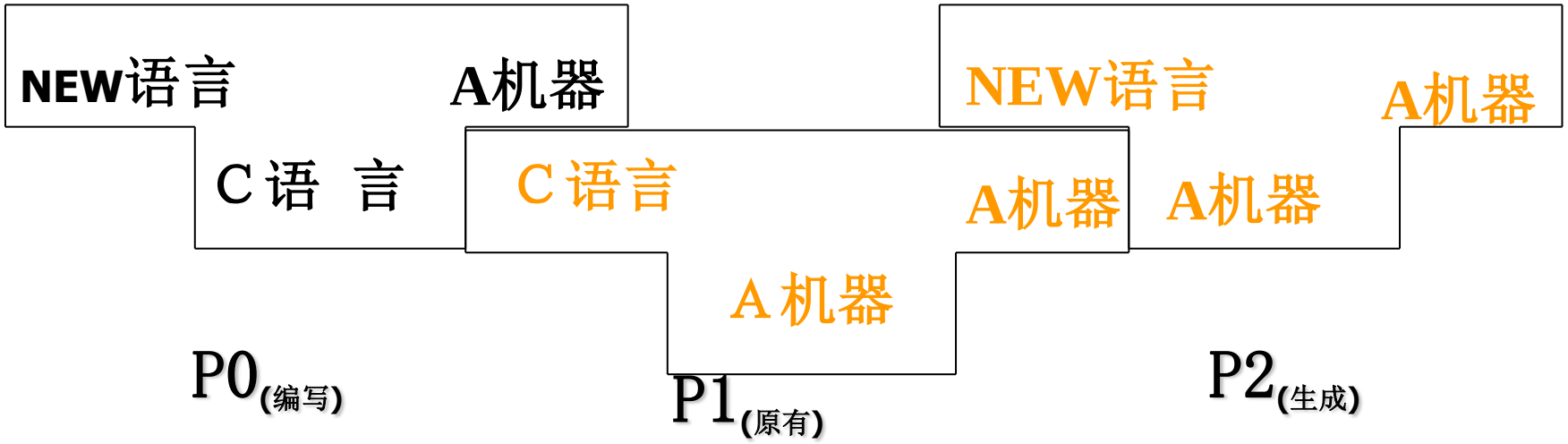
2. (A机的C编译P1)编译P0，得到在A机上可运行的P2(C→B)



3. (用A机的P2)编译P0，得到在B机上可运行的P3(C→B)

2) 本机编译器利用

- 问题二： A机上有一个C语言编译器，现要实现一个新语言NEW的编译器？能利用交叉编译技术么？
- 用C编写NEW的编译，并用C编译器编译它



3) 编译程序的自展技术

- 问题三：直接在一个机器上实现C语言编译器，还有别的技术么?
- 解决:
  - 用汇编语言实现一个C子集的编译程序(P0—人)
  - 用汇编程序处理该程序,得到P2(P2:可直接运行)
  - 用C子集编制C语言的编译程序(P3—人)
  - 用P2编译P3，得到P4

4）利用编译程序自动生成器

词法分析器的自动生成程序

输入：

词法（正规表达式）

识别动作（C程序段）

输出：

yylex( ) 函数

语法分析器的自动生成程序

输入：

语法规则（产生式）

语义动作（C程序段）

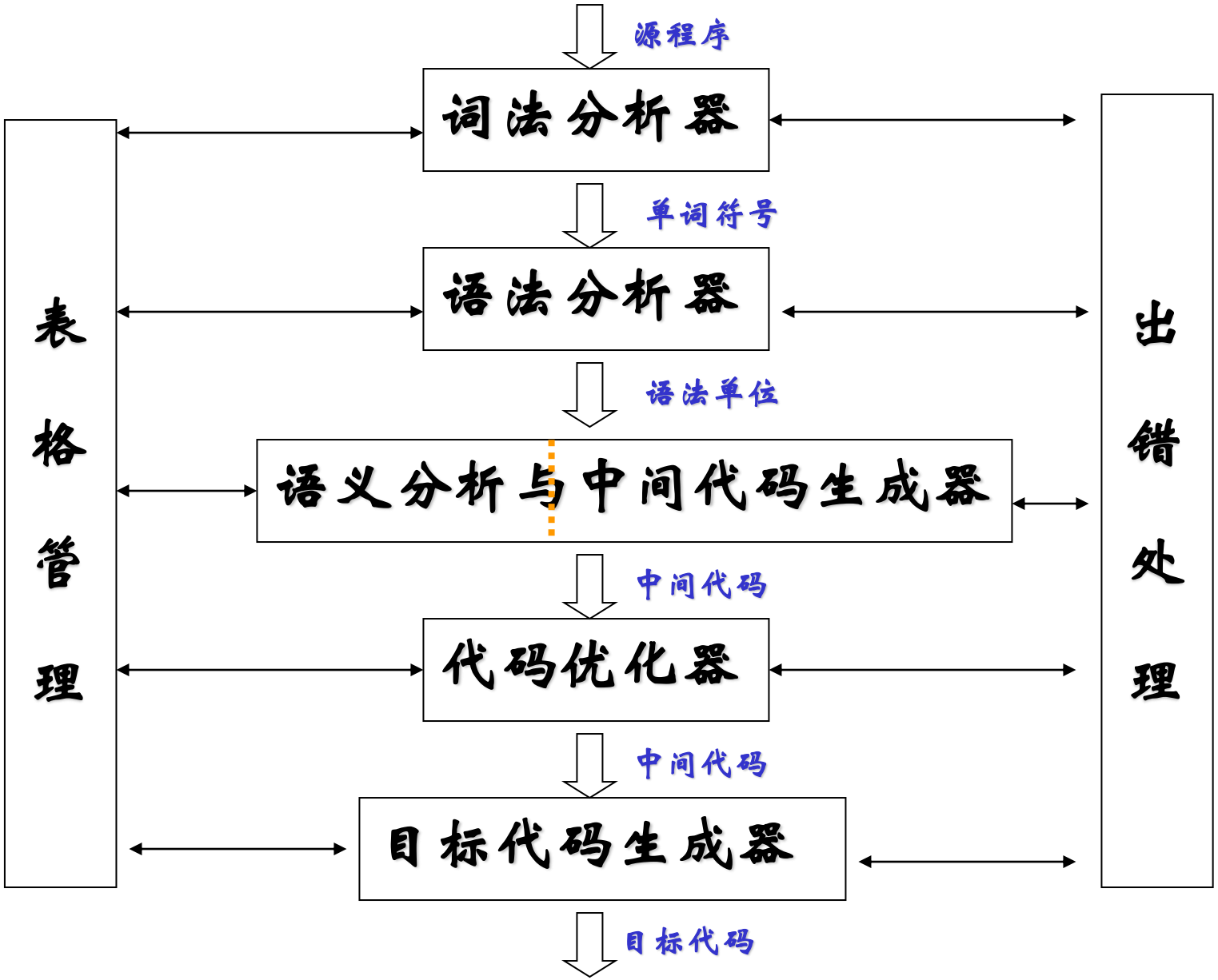
输出：

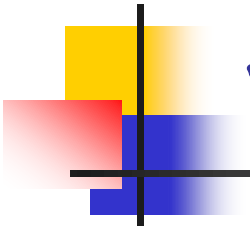
yyparse( ) 函数

1.4 编译器技术的应用

- 高级语言的实现
- 针对计算机体系结构的优化
- 新计算机体系结构的设计
- 程序翻译
- 提高软件开发效率的工具

编译程序总体结构


源程序
词法分析器
单词符号
语法分析器
语法单位
语义分析与中间代码生成器
中间代码
代码优化器
中间代码
目标代码生成器
目标代码
表格管理
出错处理



课后任务

- 请查阅Pascal语言的语法编写2个程序
  - （1）程序代码行数10行左右（包含简单的变量定义，算术运算）
  - （2）程序代码行数30行左右，包含三种基本结构（顺序、选择和循环）
- 阅读教材1.4节
- 完成课后习题1.1, 1.2, 1.4