国家级精品课

教育部-微软精品课

编译技术

编译原理及编译程序构造

北京航空航天大学计算机学院

课程目的

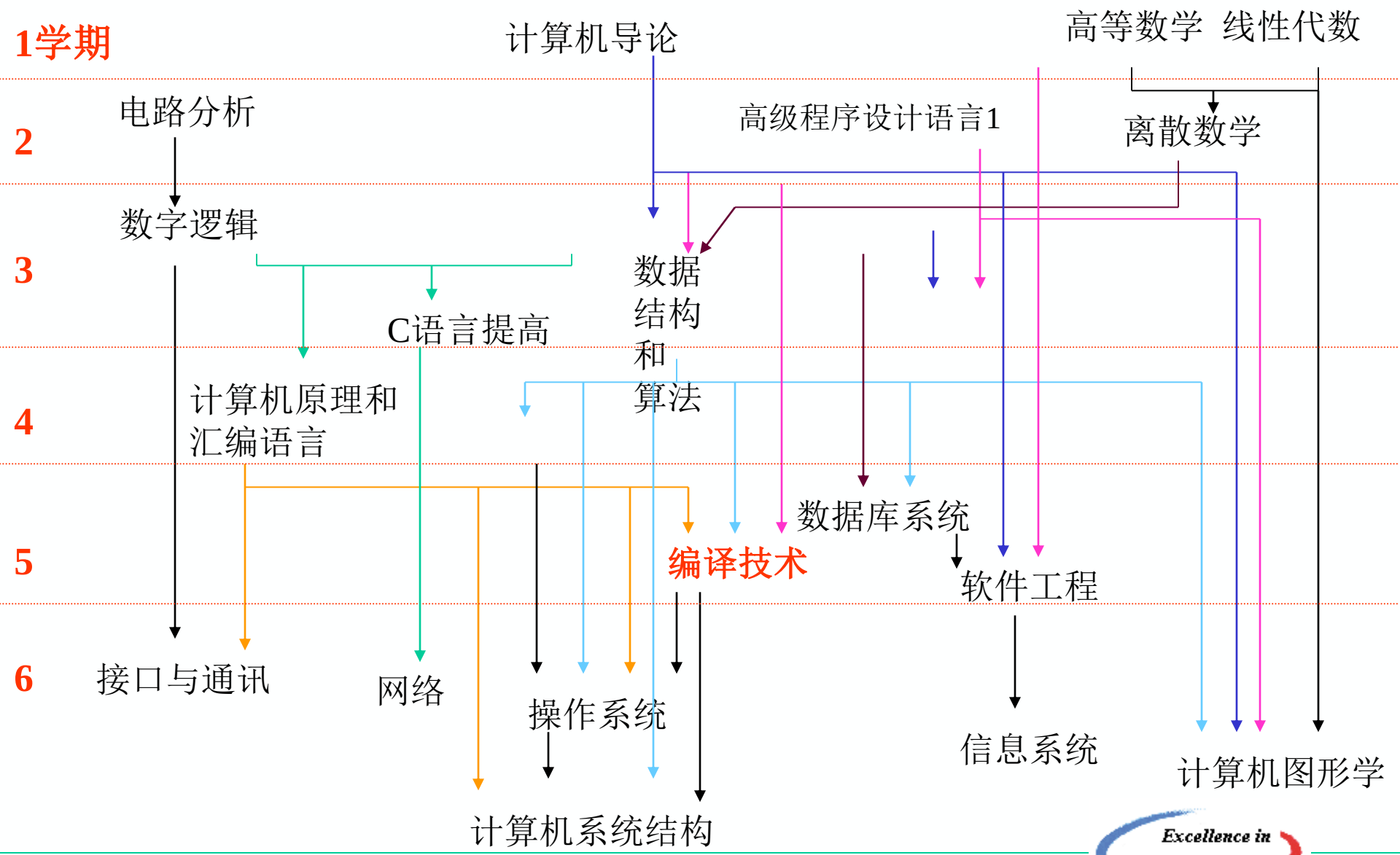
目的:

- 掌握编译的基本理论、常用的编译技术，了解编译过程及编译系统的构造（结构和机理）。
- 能运用所学技术解决实际问题，能独立编写一个小型编译系统。（课程设计大作业，可选不同难易系数）

课程定位——计算机学院核心课程

分类	课程名称	课程定位	备注
计算机基础	计算机导论	入门	
	算法和数据结构	基础	
	高级语言程序设计（1，2）	必备工具	
计算机理论（离散数学1,2,3）	数理逻辑	计算机数学	
	集合论和图论		
	组合数学		
计算机硬件类课程	数子电路和数字逻辑	硬件基础课程	含实验
	计算机原理和汇编语言	部件原理	含实验
	计算机接口与通讯	部件间通讯	含实验
	计算机体系结构	体系结构	含实验
	计算机网络		
计算机软件类课程	编译技术	系统软件层	含课程设计
	操作系统		含课程设计
	数据库系统原理		含课程设计
	软件工程		
	信息系统分析与设计	应用类	
	计算机图形学（多媒体技术）	应用类	

课程间的拓扑关系


1学期
2
3
4
5
6
计算机导论
高等数学 线性代数
电路分析
高级程序设计语言1
离散数学
数字逻辑
数据
结构
和
算法
C语言提高
计算机原理和
汇编语言
数据库系统
编译技术
软件工程
接口与通讯
网络
操作系统
信息系统
计算机图形学
计算机系统结构

- **教材和参考书**
  - 张莉，杨海燕，史晓华等《编译原理及编译程序构造》，清华出版社，2012
  - 龙书：Alfred V. Aho, Monica S.Lam, Ravi Sethi, Compilers—Principles, Techniques, and Tools. 机械出版社
  - 鲸书：Steven S. Muchnick, Advanced Compiler Design and Implementation
  - 虎书：A. W. Apple, J. Palsberg ，Modern Compiler Implementation in Java
  - Kenneth C. Louden 著，《编译原理及实践》，机械工业出版社
  - 陈火旺，刘春林，谭庆平等 编著，《程序设计语言编译原理》，国防工业出版社出版
  - 吕映芝，张素琴等，《编译原理》，清华大学出版社

特点

- 经典课程：
  - 国内外大部分学校都开设
  - 历史悠久
  - 有经典教材
  - 重视实践教学，尤其是国内外重点大学
  - 强调教学过程

编译技术

- 国外大学：名称不一样：
  - **Compiler Design**： Carnegie Mellon
  - **Programming Languages and Compilers**: University of Illinois–Urbana-Champaign, University of California–Berkeley,
  - **Compilers and Interpreters** :Yale
  - **Compiler Design and Implementation** :Harvard
  - **Compiling Techniques**: Princeton
  - **Compilers**: Stanford
- 国内：编译原理、编译技术
- 内容差异不大。

教材

- 1.**龙书(Dragon book)**：书名是Compilers: Principles,Techniques,and Tools；作者是：Alfred V.Aho,Ravi Sethi,Jeffrey D.Ullman ；国内所有的编译原理教材基本都是参考了本书，重点是编译的前端技术。

- 2.**鲸书(Whale book)**：书名是：Advanced Compiler Design and Implementation；作者是：Steven S.Muchnick ；也就是高级编译原理。

- 3.**虎书(Tiger book)**：书名是：Modern Compiler Implementation in Java/C++/ML, Second Edition；作者是：Andrew W.Appel,with Jens Palsberg。这本书是3本书中最薄的一本，也是最最牛的一本！

评分标准分析比较

- **CMU**：作业80%，期中考试 20%
- **UIUC:**
  - 8 道题目，20%
  - 2 次期中考试，各占25%
  - 1 次期末考试，占30%
- **Berkeley**：
  - 编程作业：7个编程作业，30%
  - 考试：两次平时考试和一次期末考试，分别占10%，10%，30%
  - 问题集（Problem sets）：提交“学习问题”，15%
  - 参加讨论：根据出勤、讨论课和协作情况给分，5%
- **Priceton**：
  - 不完成高级项目 (total = 100%)
    - 标准编程作业 (1-10): 50%
    - 期中和期末考试 50%
  - 完成高级项目 (total = 100%)
    - 高级项目: N% (0 < N < 50; N 与难度有关.)
    - 标准编程作业(1-5): (50-3N/4)%
    - 期中和期末考试: (50-N/4)%

课程定位的理解

- **系统性**：编译是一个完整的系统，也是学生接触到的第一个系统
- **过程完整性**：编译不仅讲解了编译技术，其实质是讲解了模型从一种语言表达形式到另一种语言表达形式的等价转化方法，应该保证过程的完整性
- **实践性**：理论和实践相结合。无论是研究性大学、非研究性大学都应该注重实践。不同的在于具体要求的不同。

教学要求

- 理论学习
  - 理解高级程序语言的工作原理
  - 学习不同语言表示的程序之间的等价转化方法
  - 编译程序的构造和工作原理
  - 编译相关技术和常用优化技术
- 能力培养
  - 理解系统的概念，完整设计一个不同难度的编译系统

课程要求

★课时:48学时

★ 分为两部分:(分别计分)

- 理论基础（3学分）:课堂教学，按时交作业。
  - 作业10分(补交或晚交6分）；
  - 3-6次随堂考试，共计30分；（不通知、不补）
  - 期末闭卷考试，60分
  - 主动回答问题，每次奖励0.5分，5分封顶(考前公布)
- 实践部分(1.5学分）：上机实践(50机时)

要　　求：

1. **提前预习，上课认真听讲；**
   课后及时复习，独立认真完成作业。
2. **每周一交作业。**

网　　址：

http://www.sei.buaa.edu.cn/compile/
http://judge.buaa.edu.cn(115.25.138.223)

第一章 概论

（介绍名词术语、了解编译系统的结构和编译过程）

- 编译的起源：程序设计语言的发展
- 基本概念
- 编译过程和编译程序构造
- 编译技术的应用

1.1 程序设计语言的发展

机器语言（机器指令） → 汇编语言 → 面向用户的语言 → 面向问题的语言

C7 06 0000 0002　　MOV x, 2

x = 2

低级语言

高级语言

- 低级语言（Low level Language)
  - 字位码、机器语言、汇编语言
    - 特点：与特定的机器有关，功效高，但使用复杂、繁琐、费时、易出错
- 高级语言
  - -- Fortran、Pascal、C 语言等
    - 特点：不依赖具体机器，移植性好、对用户要求低、易使用、易维护等。

用高级语言编制的程序，计算机不能立即执行，必须通过一个“翻译程序”加工，转化为与其等价的机器语言程序，机器才能执行。

这种翻译程序，称之为“编译程序”。

1.2 基本概念

- **源程序**

  用汇编语言或高级语言编写的程序称为源程序。

- **目标程序**

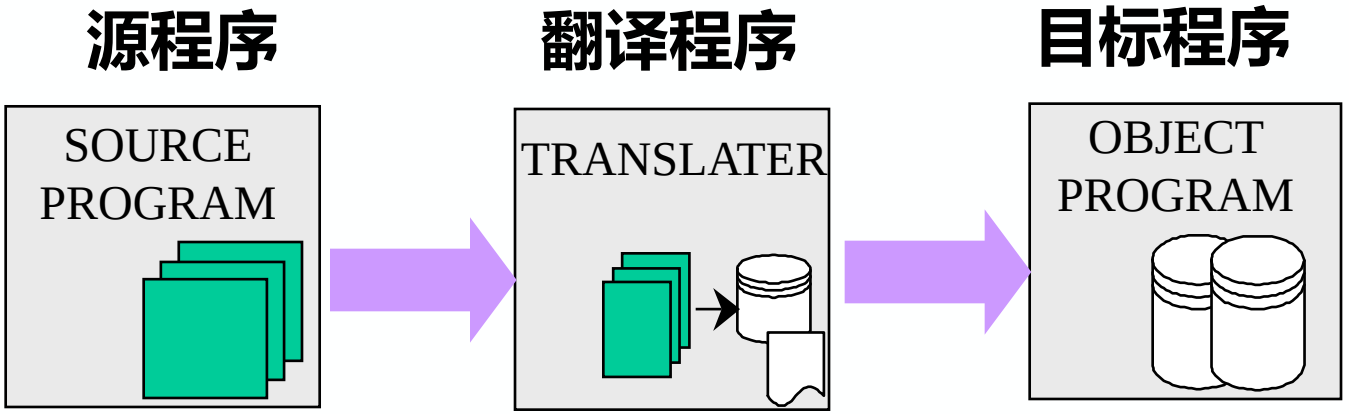
  用目标语言所表示的程序。

  目标语言：可以是介于源语言和机器语言之间的“中间语言”，可以是某种机器的机器语言，也可以是某机器的汇编语言。

- **翻译程序**

  将源程序转换为目标程序的程序称为翻译程序。它是指各种语言的翻译器，包括汇编程序和编译程序，是汇编程序、编译程序以及各种变换程序的总称。

源程序、翻译程序、目标程序 **三者关系：**



即源程序是翻译程序的输入，目标程序是翻译程序的输出

源程序	翻译程序	目标程序
汇编语言	汇编程序	机器语言
高级语言	编译程序	目标程序

- **汇编程序**

若源程序用汇编语言书写，经过翻译程序得到用机器语言表示的程序，这时的翻译程序就称之为汇编程序，这种翻译过程称为“汇编”（Assemble）

- **编译程序**

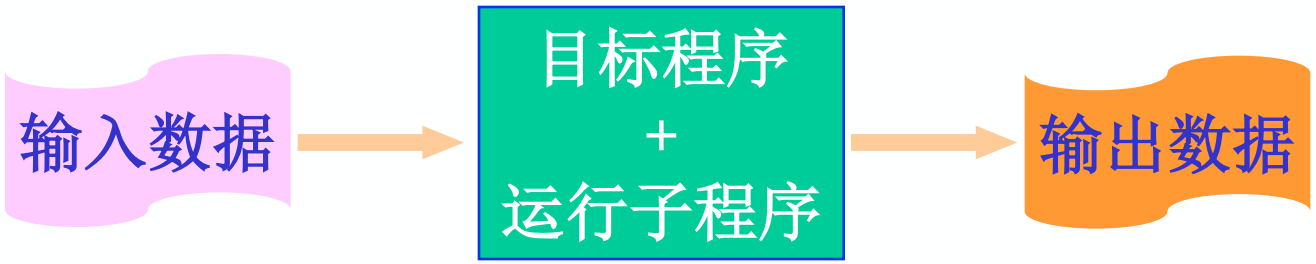
若源程序是用高级语言书写，经加工后得到目标程序，这种翻译过程称“编译”（Compile)

汇编程序与**编译程序**都是**翻译程序**，主要区别是加工对象的不同。由于汇编语言格式简单，常与机器语言之间有一一对应的关系，汇编程序所要做的翻译工作比编译程序简单得多。

源程序的编译和运行

- 编译或汇编阶段

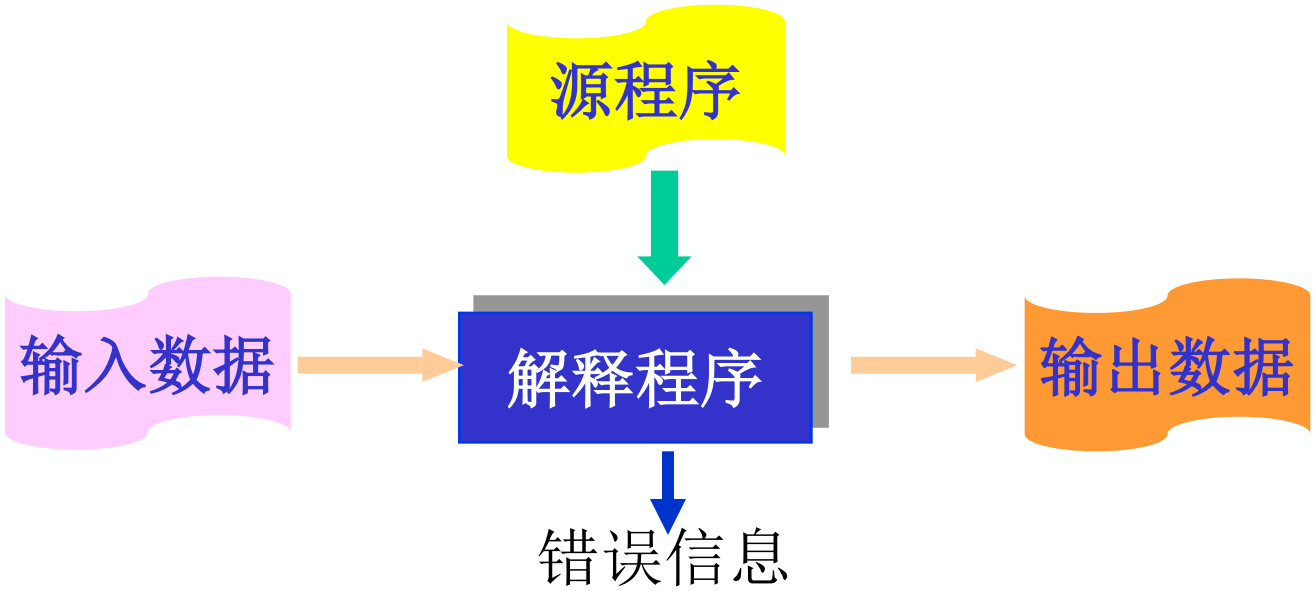
- 运行阶段



- **解释程序（Interpreter）**

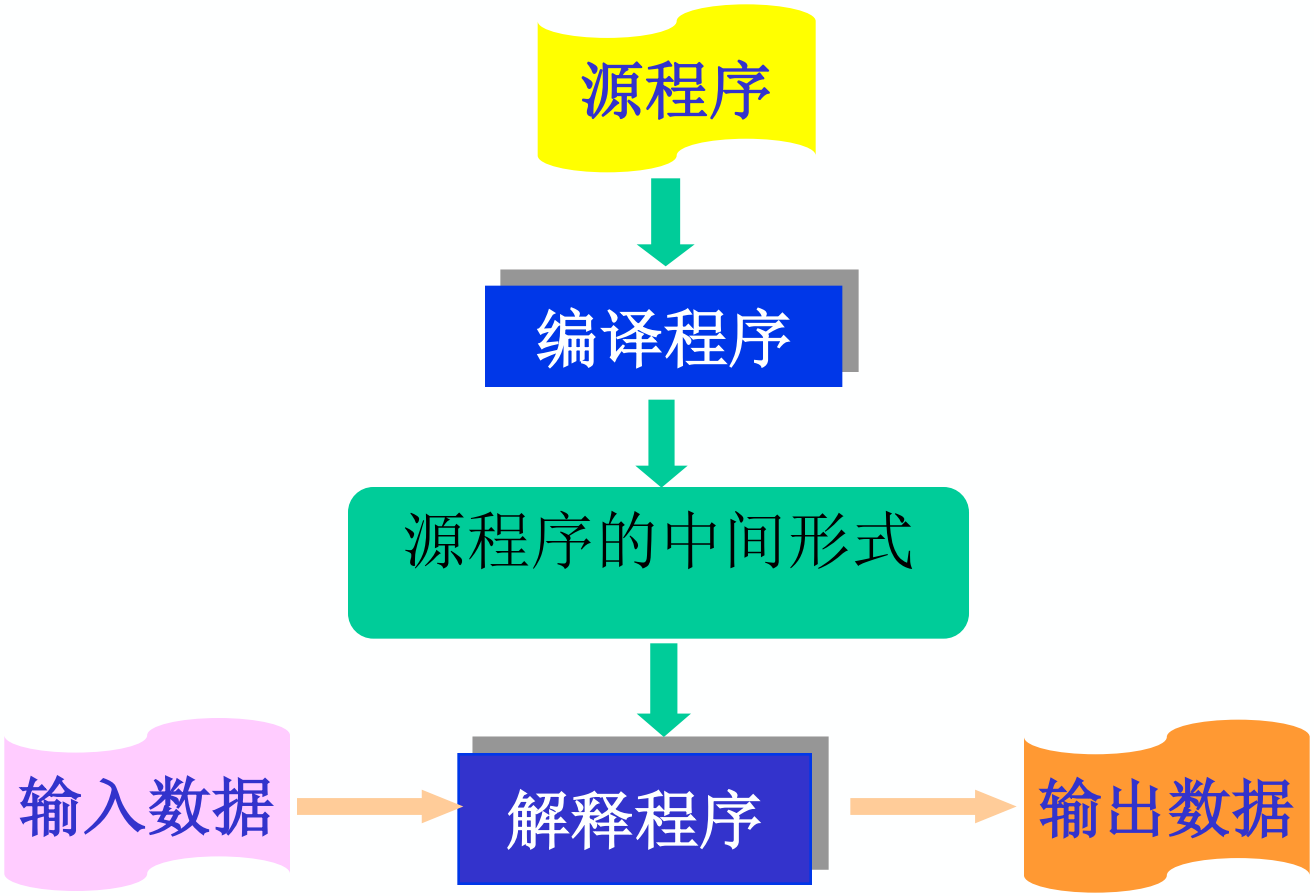
  对源程序进行解释执行的程序。

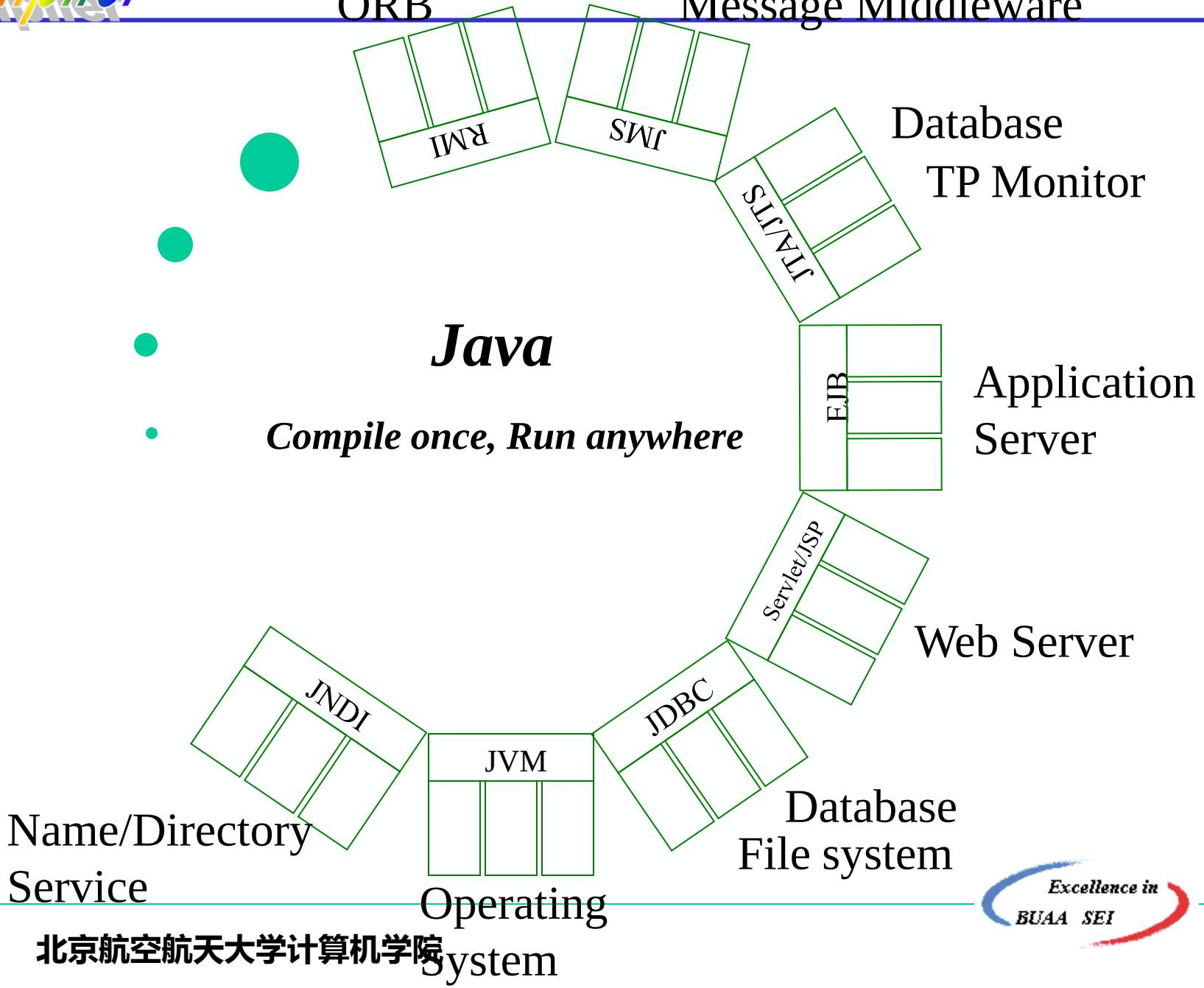
- **工作过程**



- **特点、与编译程序比较**

“编译-解释执行”系统




ORB
Message Middleware
RMI
JMS
Database
TP Monitor
JTA/JTS
Java
Compile once, Run anywhere
EJB
Application
Server
Servlet/JSP
Web Server
JNDI
JVM
JDBC
Database
File system
Name/Directory
Service
Operating
System

1.3.1 编译过程

编译过程是指将高级语言程序翻译为等价的目标程序的过程。

习惯上是将编译过程划分为5个基本阶段：

一、词法分析

任务：分析和识别单词。

源程序是由字符序列构成的，词法分析扫描源程序(字符串),根据语言的词法规则分析并识别单词，并以某种编码形式输出。

- 单词：是语言的基本语法单位，一般语言有四大类单词

<1>语言定义的关键字或保留字（如BEGIN、END、IF）

<2>标识符

<3>常数

<4>分界符（运算符） （如+、-、*、/、;、（、） .....）

对于如下的字符串,词法分析程序将分析和识别出9个单词：

X1	:=	(	2.0	+	0.8	)	*	C1
1	2	3	4	5	6	7	8	9

⟹ 也称为线性分析。

二、语法分析

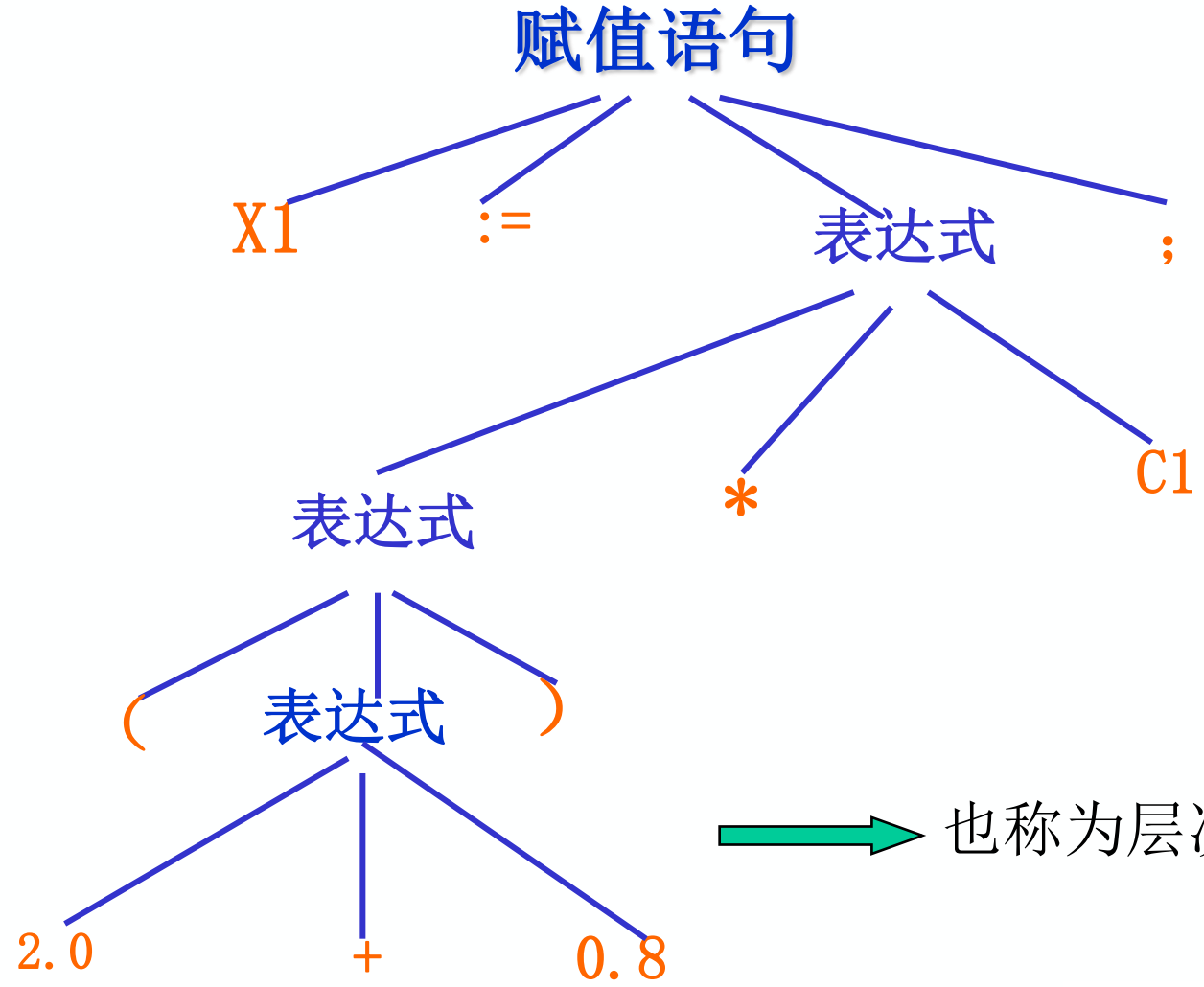
任务：根据语法规则（即语言的文法），分析并识别出各种语法成分，如表达式、各种说明、各种语句、过程、函数等，并进行语法正确性检查。

X1:= (2.0 + 0.8) * C1

赋值语句的文法：

<赋值语句>→<变量><赋值操作符><表达式>
<变量>→<简单标识符>
<赋值操作符>→:=
<表达式>→ ……

X1:= (2.0 + 0.8) * C1;



也称为层次分析。

三、语义分析、生成中间代码

任务：对识别出的各种语法成分进行语义分析，
并产生相应的中间代码。

- 中间代码：一种介于源语言和目标语言之间的中间语言形式
- 生成中间代码的目的：
  〈1〉便于做优化处理；
  〈2〉便于编译程序的移植。
- 中间代码的形式：编译程序设计者可以自己设计，常用的有四元式、三元式、逆波兰表示等。

例：X1:= （ 2.0 + 0.8 ）* C1

- 由**语法分析**识别出为赋值语句，语义分析首先要分析语义上的正确性，例如要检查表达式中和赋值号两边的类型是否一致。

- 根据赋值语句的语义，生成中间代码。即用一种语言形式来代替另一种语言形式，这是翻译的关键步骤。（**翻译的实质：**语义的等价性）

★ 四元式（三地址指令）

X1:=（2.0 + 0.8）* C1

	运算符	左运算对象	右运算对象	结果
(1)	+	2.0	0.8	T1
(2)	*	T1	C1	T2
(3)	:=	X1	T2	

其中T1和T2为编译程序引入的工作单元

四元式的语义为：
2.0 + 0.8 → T1
T1 * C1 → T2
T2 → X1

这样所生成的四元式与原来的赋值语句在语言的形式上不同，但语义上等价。

四、代码优化

任务：目的是为了得到高质量的目标程序。

例如：前面的四元式中第一个四元式是计算常量表达式值，该值在编译时就可以算出并存放在工作单元中，不必生成目标指令来计算，这样四元式可优化为：

编译时： 2.0 + 0.8 → T1

(1) * T1 C1 T2

(2) := X1 T2

优化前的四元式：

(1)	+	2.0	0.8	T1
(2)	*	T1	C1	T2
(3)	:=	X1	T2	

五、生成目标程序

由中间代码很容易生成目标程序（地址指令序列）。这部分工作与机器关系密切，所以要根据机器进行。在做这部分工作时（要注意充分利用累加器），也可以进行优化处理。 X1:=（2.0 + 0.8）* C1

```
LOAD   2.0
ADD    0.8
STO    T1
LOAD   T1
MUL    C1
STO    T2
LOAD   T2
STO    X1
```

作利用累加器的优化

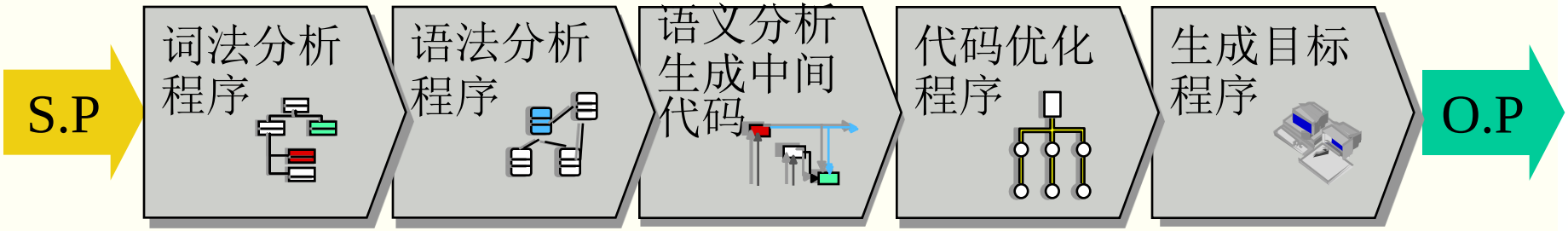
```
LOAD   2.0
ADD    0.8
MUL    C1
STO    X1
```

注意：在翻译成目标程序的过程中，要切记保持语义的等价性。

1.3.2 编译程序构造

一、 编译程序的逻辑结构

按逻辑功能不同，可将编译过程划分为五个基本阶段，与此相对应，我们将实现整个编译过程的编译程序划分为五个逻辑阶段（即五个逻辑子过程）。



在上列五个阶段中都要做两件事：

（1）建表和查表；（2）出错处理；

所以编译程序中都要包括符号表管理和出错处理两部分

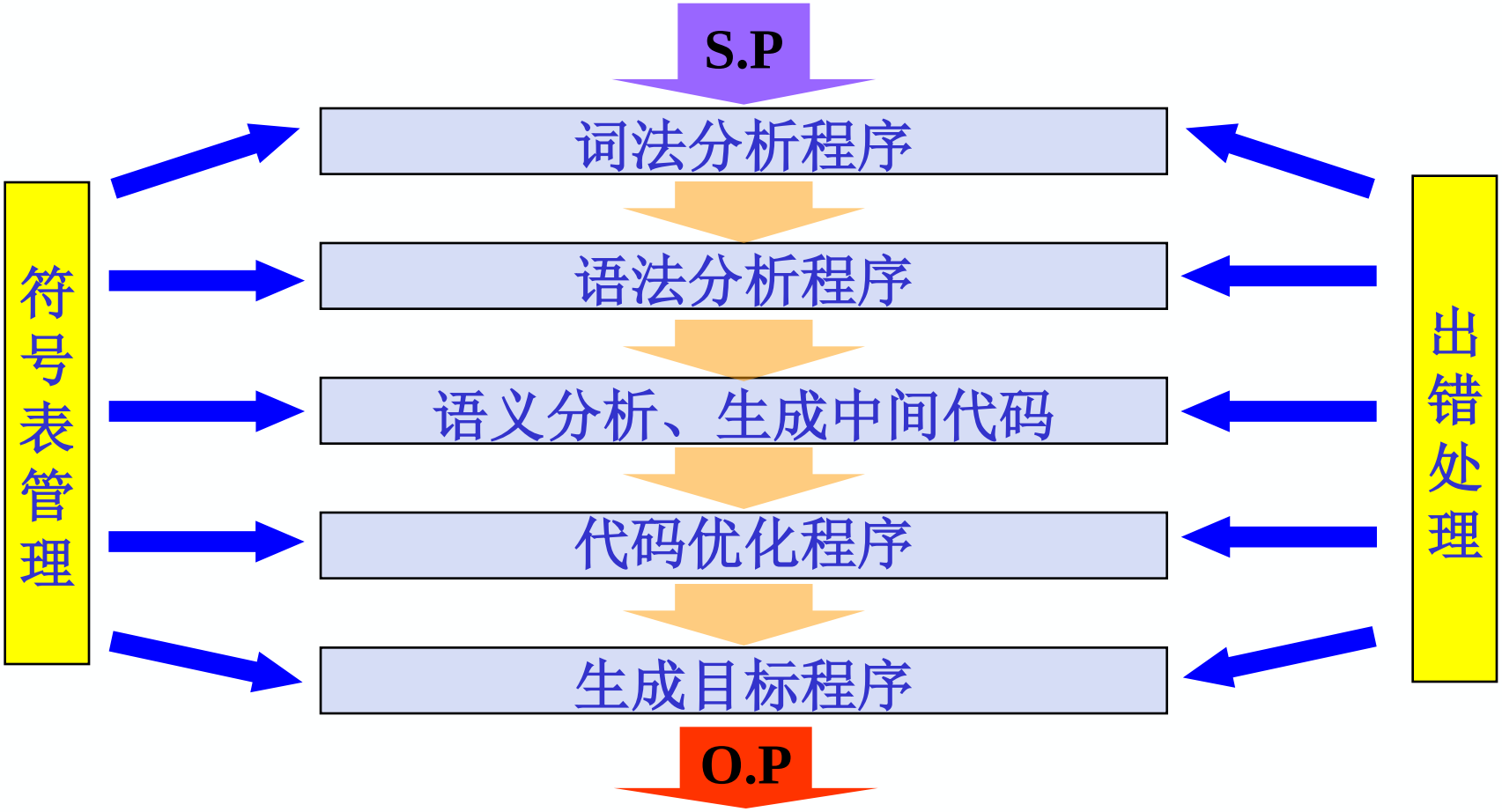
★ 符号表管理

在整个编译过程中始终都要贯穿着建表（填表）和查表的工作。即要及时地把源程序中的信息和编译过程中所产生的信息登记在表格中，而在随后的编译过程中同时又要不断地查找这些表格中的信息。

★ 出错处理

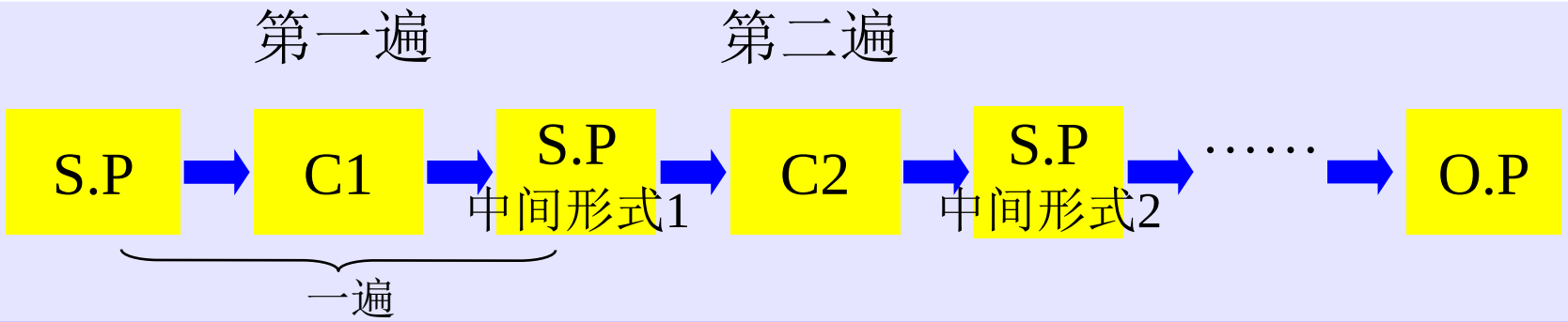
规模较大的源程序难免有多种错误，编译程序必须要有出错处理的功能。即能诊察出错误，并能报告用户错误的性质和位置，以便用户修改源程序。出错处理能力的大小是衡量编译程序质量好坏的一个重要指标。

典型的编译程序具有7个逻辑部分



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。



上一遍的结果是下一遍的输入，最后一遍生成目标程序。

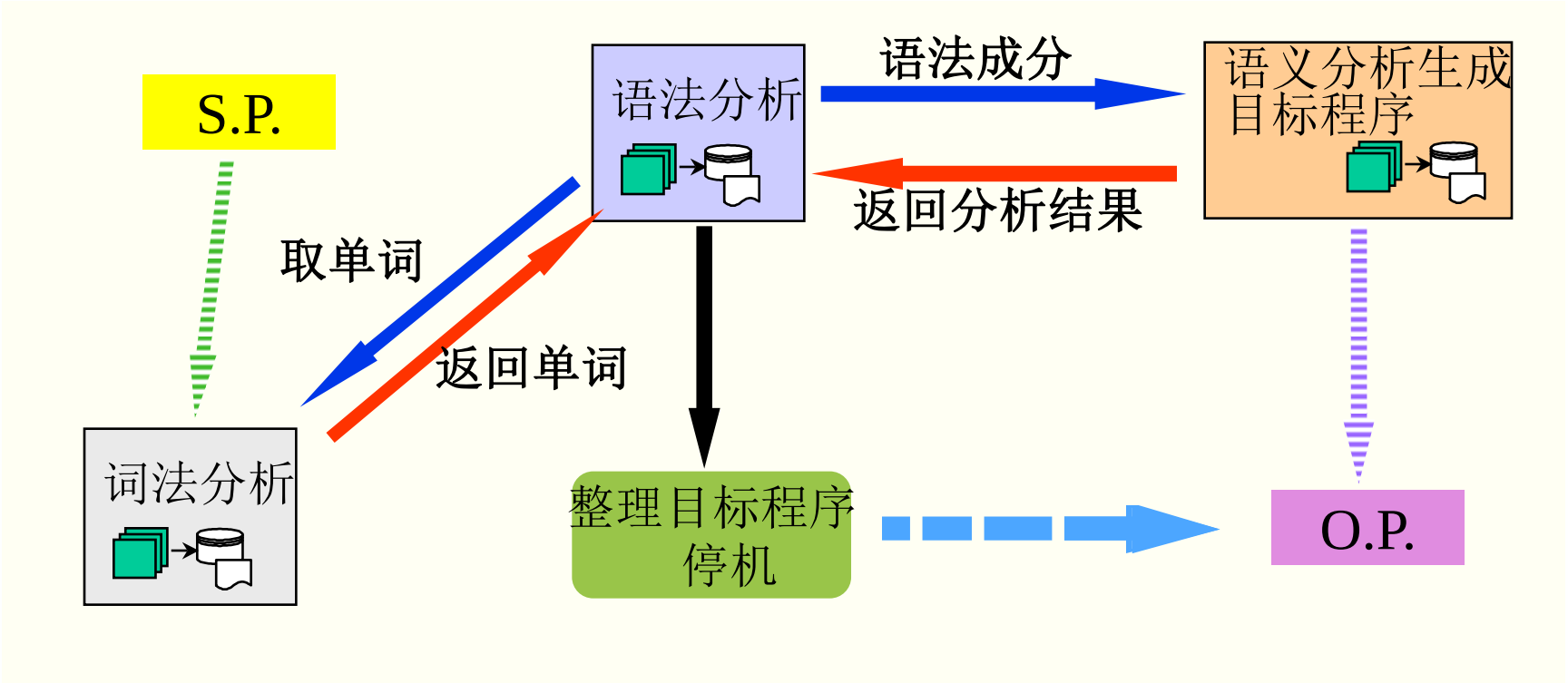
★要注意遍与基本阶段的区别

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。

一遍扫描即可完成整个编译工作的称为一遍扫描编译程序

其结构为：



三、前端和后端

根据编译程序各部分功能，将编译程序分成前端和后端。

前端：通常将与源程序有关的编译部分称为前端。

词法分析、语法分析、语义分析、中间代码生成、代码优化 --------分析部分

特点：与源语言有关

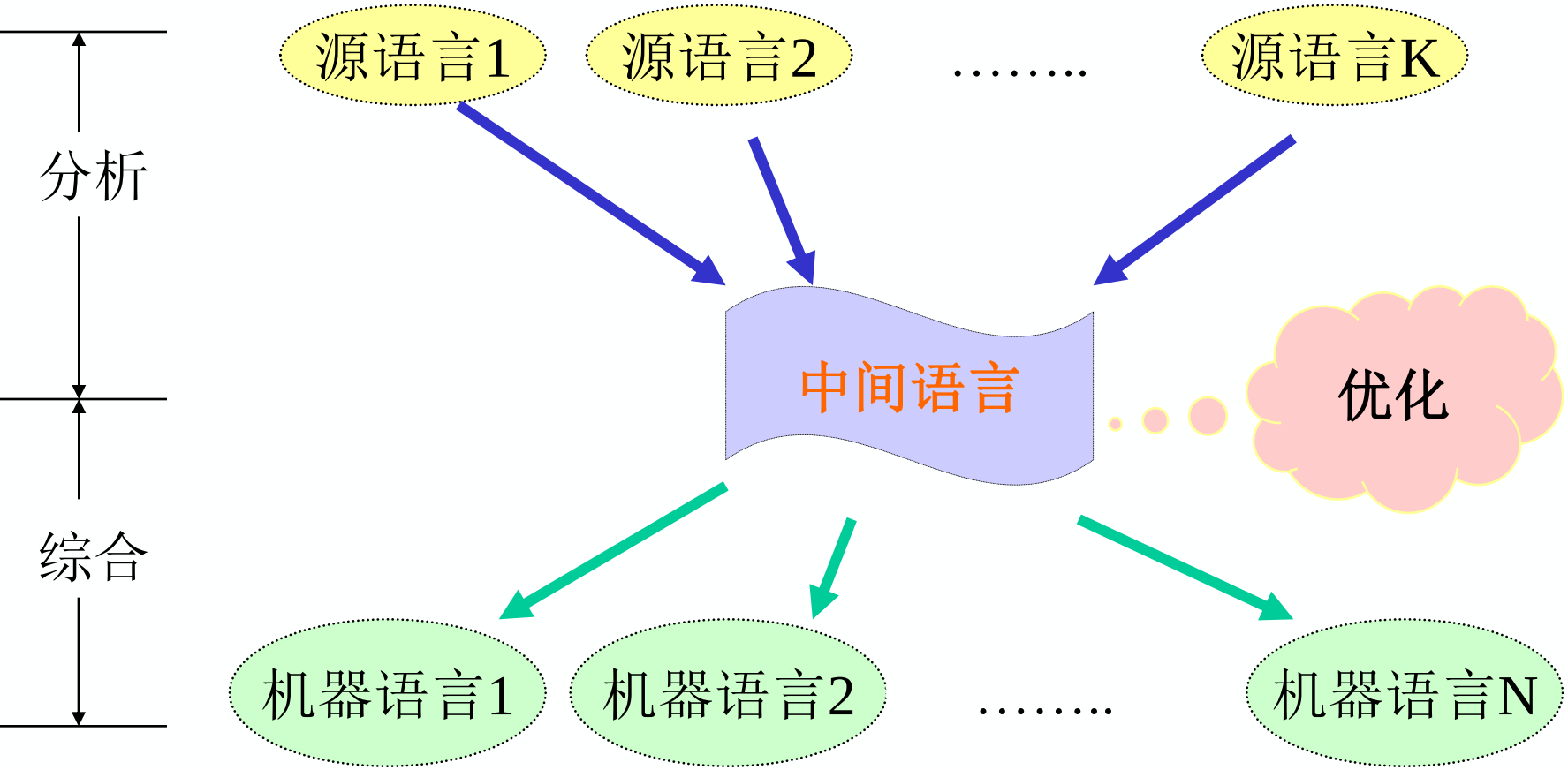
后端：与目标机有关的部分称为后端。

目标程序生成（与目标机有关的优化）

--------综合部分

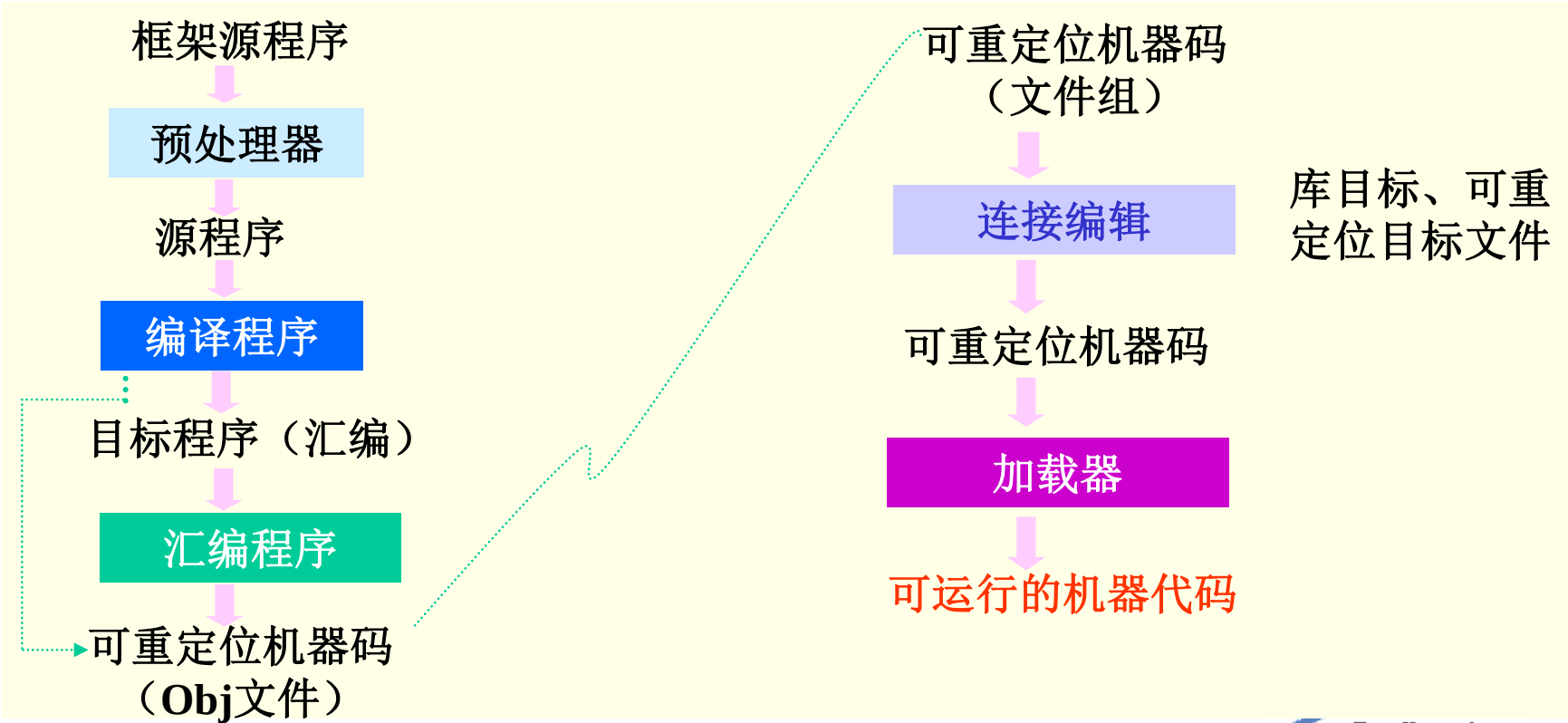
特点：与目标机有关

中间表示的重要性



四、 编译程序的前后处理器

源程序：多文件、宏定义和宏调用，包含文件
目标程序：一般为汇编程序或可重定位的机器代码



1.4 编译技术的应用

- 语法制导的结构化编辑器
- 程序格式化工具
- 软件测试工具
- 程序理解工具
- 高级语言的翻译工具
- 等等。

作业： P15： 1,2,3

课程内容的组织、教学安排

第0章：课程要求
第一章：概论
1.1 编译的起源：程序设计语言的发展
1.2 基本概念
1.3 编译过程和编译程序构造
1.4 译技术的应用
第二章：文法和语言的概念和表示
2.1 预备知识
2.2 文法的非形式讨论
2.3 文法和语言的形式定义
2.4 语法树与二义性文法
2.5 句子的分析
2.6 有关文法的实用限制
2.7 文法的其它表示法
2.8 文法和语言分类
第三章：词法分析
3.1 词法分析程序的功能
3.2 词法分析程序的设计与实现
3.3 词法分析程序的自动生成
第四章：语法分析
4.1 概述：语法分析的功能；基本任务；语法分析方法分为两大类：自顶向下分析法和自底向上分析法
4.2 顶向下分析法
（1）自顶向下分析的一般过程
（2）递归下降分析法
（3）LL1分析法
4.3自底向上分析法
（1）自底向上分析的一般过程
（2）算符优先分析法
（3）SLR1分析法

第五章：符号表管理
5.1 概述
5.2 符号表的组织和内容
5.3 非分程序结构语言的符号表组织
5.4 分程序结构语言的符号表组织
第六章：运行时的存储组织及管理
6.1 概述
6.2 静态存储分配
6.3 动态存储分配
第七章：源程序的中间形式
7.1 波兰表示
7.2 N-元表示
7.3 抽象机代码
第八章：错误处理
8.1 概述
8.2 错误分类
8.3 错误的诊察和报告
8.4 错误处理技术
第九章：语法制导翻译技术
9.1 翻译文法和语法制导翻译
9.2 属性翻译文法
9.3 自顶向下语法制导翻译
第十章：语义分析和代码生成
原理和几个典型实例：声明语句；表达式语句；赋值语句；控制语句；过程调用和返回
第十一章：代码优化
介绍常用的代码优化方法：和当前的研究热点。
第十二章：目标代码生成
介绍面向目标体系结构的代码生成和优化技术。
第十三章：编译程序生成方法和工具
介绍编译程序的书写工具、自展、移植、YACC、LEX

过程完整：至少翻译过程完整

习惯上是将编译过程划分为5个基本阶段：

词法分析

语法分析

语义分析、生成中间代码

代码优化

生成目标程序

词法分析程序：状态图，手工编写程序

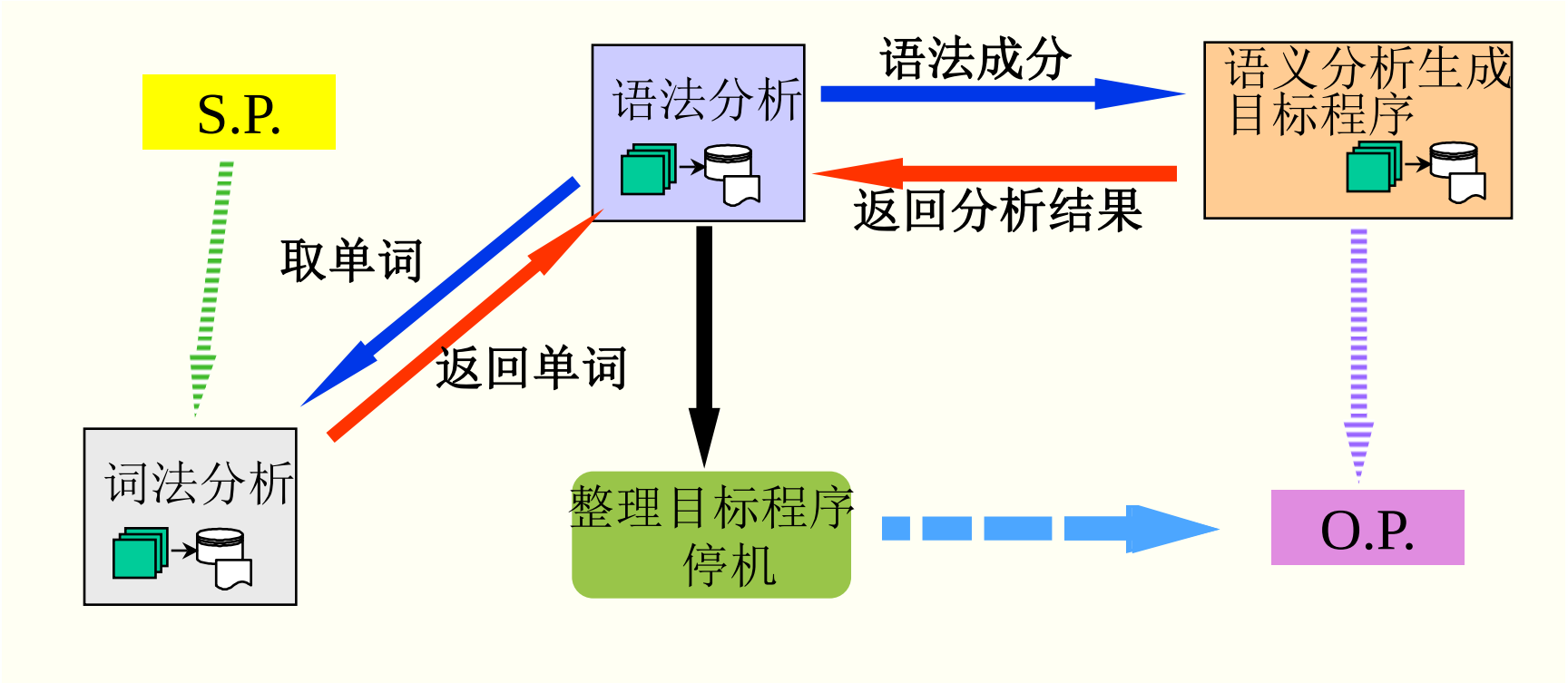
语法分析程序：递归子程序法，编写程序

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

一遍扫描即可完成整个编译工作的称为一遍扫描编译程序

其结构为：



过程完整基础上：提高

习惯上是将编译过程划分为5个基本阶段：

词法分析

语法分析

语义分析、生成中间代码

代码优化

生成目标程序

词法分析程序：状态图，手工编写程序

词法分析程序的自动生成：有穷自动机，LEX

语法分析程序：递归子程序法，编写程序

自顶向下分析法：

自底向上分析法

语义分析和代码生成：语法制导的翻译（属性翻译文法）

第四章　语法分析

语法分析方法

- 自顶向下分析法
  - (1) 递归子程序法
  - (2) LL(1)分析法（算法和自动生成）
- 自底向上分析法
  - (1)算符优先分析法（算法和自动生成）
  - (2)LR分析法（算法和自动生成--YACC）

过程完整基础上：提高

习惯上是将编译过程划分为5个基本阶段：

词法分析

语法分析

语义分析、生成中间代码

代码优化

生成目标程序

词法分析程序：状态图，手工编写程序

词法分析程序的自动生成：有穷自动机，LEX

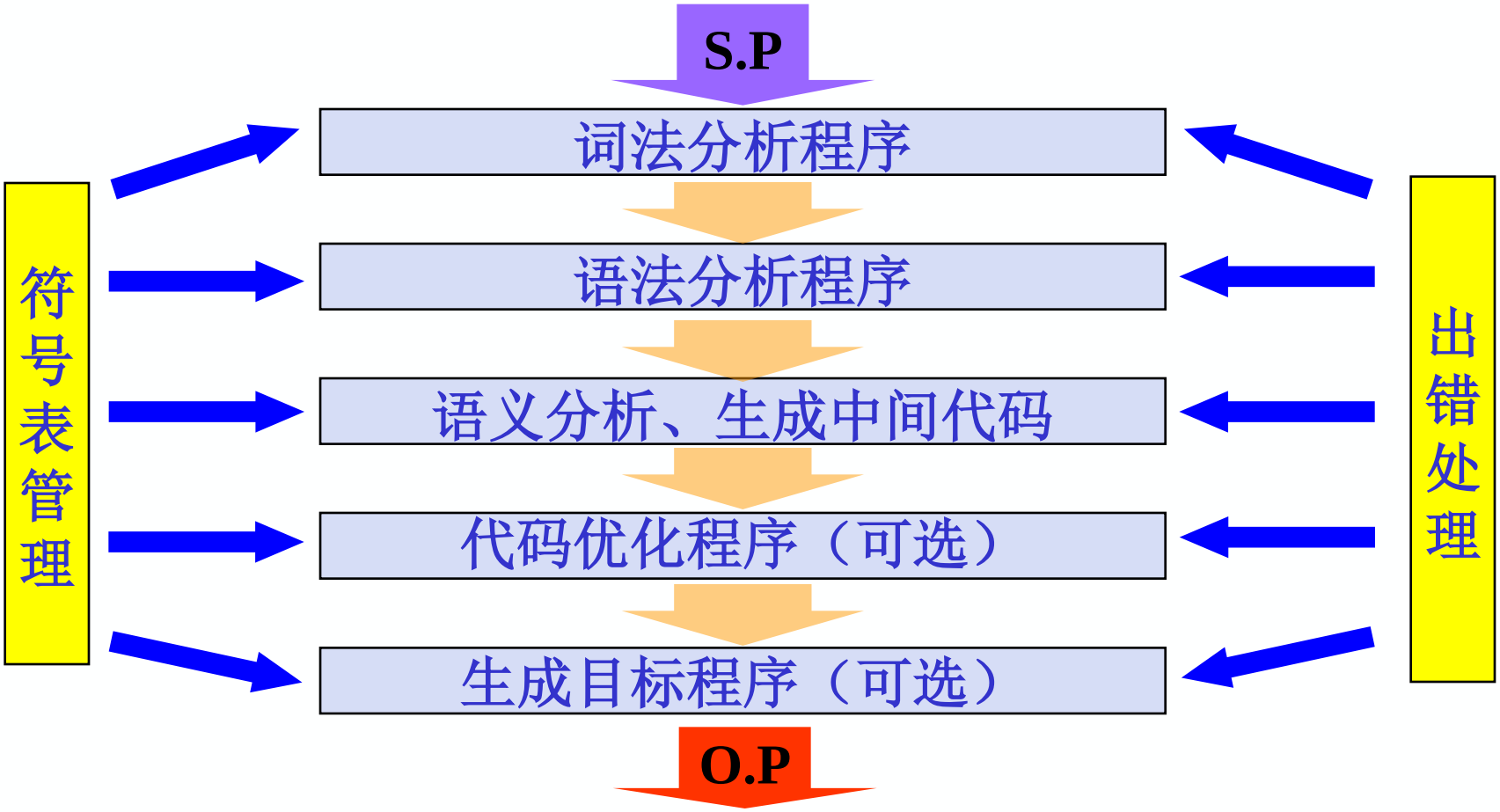
语法分析程序：递归子程序法，编写程序

自顶向下分析法：

自底向上分析法

语义分析和代码生成：语法制导的翻译（属性翻译文法）

系统完整


S.P
词法分析程序
语法分析程序
语义分析、生成中间代码
代码优化程序（可选）
生成目标程序（可选）
O.P
符号表管理
出错处理

教学安排

词法分析

语法分析

语义分析、生成中间代码

代码优化

生成目标程序

词法分析程序：状态图，手工编写程序

编写词法分析程序

语法分析程序：递归子程序法，编写程序

编写语法分析程序

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

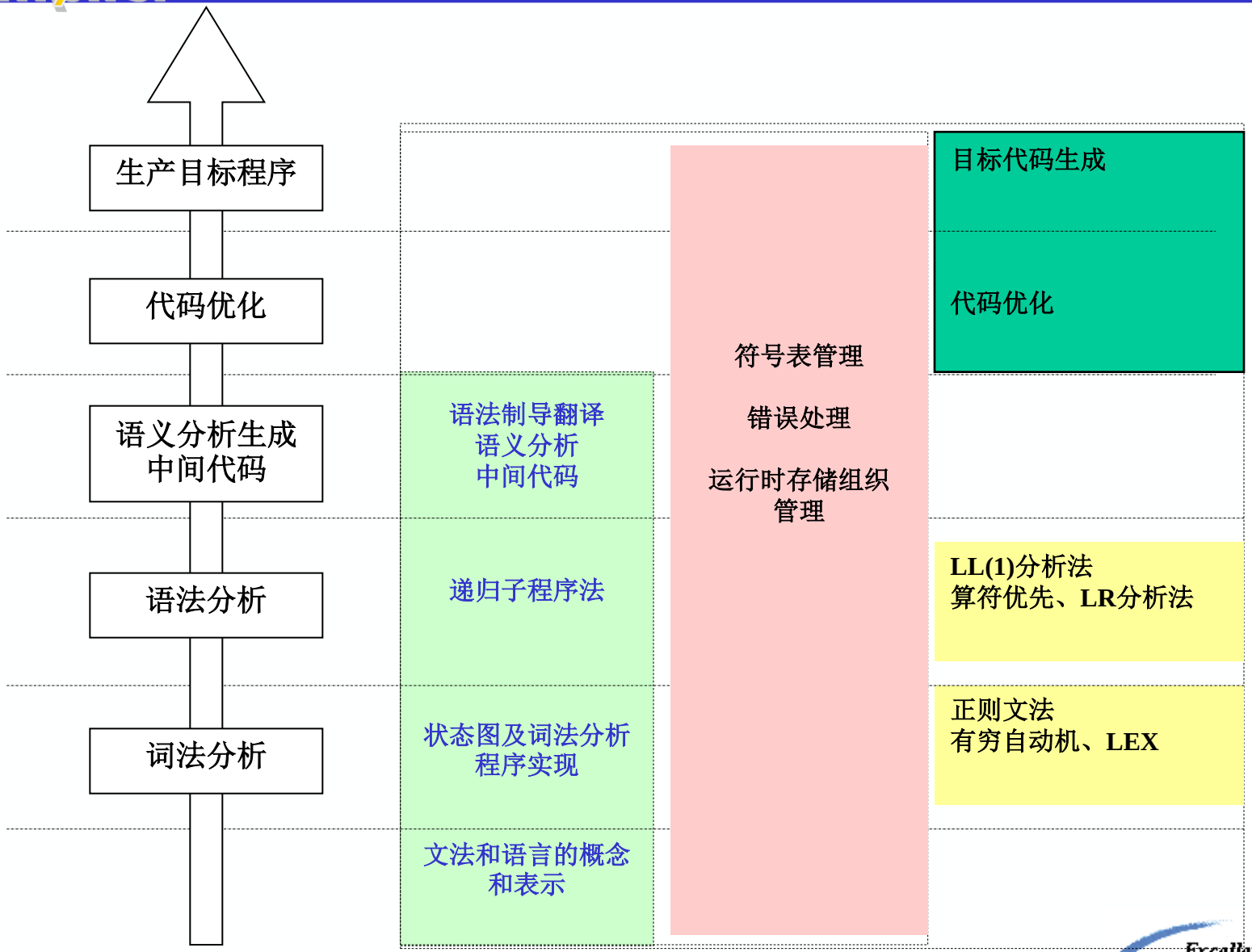


图1 编译过程和知识点组织

课程内容的组织、教学安排

第0章：课程要求
第一章：概论
1.1 编译的起源：程序设计语言的发展
1.2 基本概念
1.3 编译过程和编译程序构造
1.4 译技术的应用
第二章：文法和语言的概念和表示
2.1 预备知识
2.2 文法的非形式讨论
2.3 文法和语言的形式定义
2.4 语法树与二义性文法
2.5 句子的分析
2.6 有关文法的实用限制
2.7 文法的其它表示法
2.8 文法和语言分类
第三章：词法分析
3.1 词法分析程序的功能
3.2 词法分析程序的设计与实现
3.3 词法分析程序的自动生成
第四章：语法分析
4.1 概述：语法分析的功能；基本任务；语法分析方法分为两大类：自顶向下分析法和自底向上分析法
4.2 顶向下分析法
（1）自顶向下分析的一般过程
（2）递归下降分析法
（3）LL1分析法
4.3自底向上分析法
（1）自底向上分析的一般过程
（2）算符优先分析法
（3）SLR1分析法

第五章：符号表管理
5.1 概述
5.2 符号表的组织和内容
5.3 非分程序结构语言的符号表组织
5.4 分程序结构语言的符号表组织
第六章：运行时的存储组织及管理
6.1 概述
6.2 静态存储分配
6.3 动态存储分配
第七章：源程序的中间形式
7.1 波兰表示
7.2 N-元表示
7.3 抽象机代码
第八章：错误处理
8.1 概述
8.2 错误分类
8.3 错误的诊察和报告
8.4 错误处理技术
第九章：语法制导翻译技术
9.1 翻译文法和语法制导翻译
9.2 属性翻译文法
9.3 自顶向下语法制导翻译
第十章：语义分析和代码生成
原理和几个典型实例：声明语句；表达式语句；赋值语句；控制语句；过程调用和返回
第十一章：代码优化
介绍常用的代码优化方法：和当前的研究热点。
第十二章：目标代码生成
介绍面向目标体系结构的代码生成和优化技术。
第十三章：编译程序生成方法和工具
介绍编译程序的书写工具、自展、移植、YACC、LEX

回顾：

编译的起源

概念：源程序 目标程序 翻译程序

汇编程序 编译程序

编译过程：5个基本阶段

编译程序：7个逻辑部分

遍 前端 后端 前后处理器