

编译原理

王 挺

国防科学技术大学计算机学院

Email: tingwang@nudt.edu.cn

编译原理

引论

引论

- ▶ 什么是编译程序
- ▶ 为什么要学习编译原理
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

编译原理

引论——什么是编译程序

引论

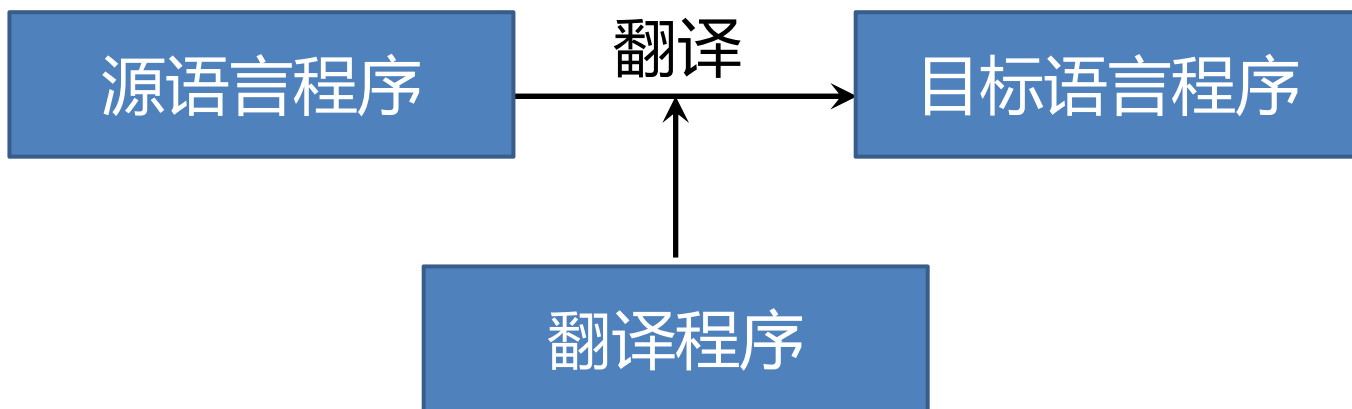
- ▶ 什么是编译程序
- ▶ 为什么要学习编译原理
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

课程内容

- ▶ 介绍程序设计语言编译程序构造的基本原理和基本实现技术
- ▶ 什么是编译程序？

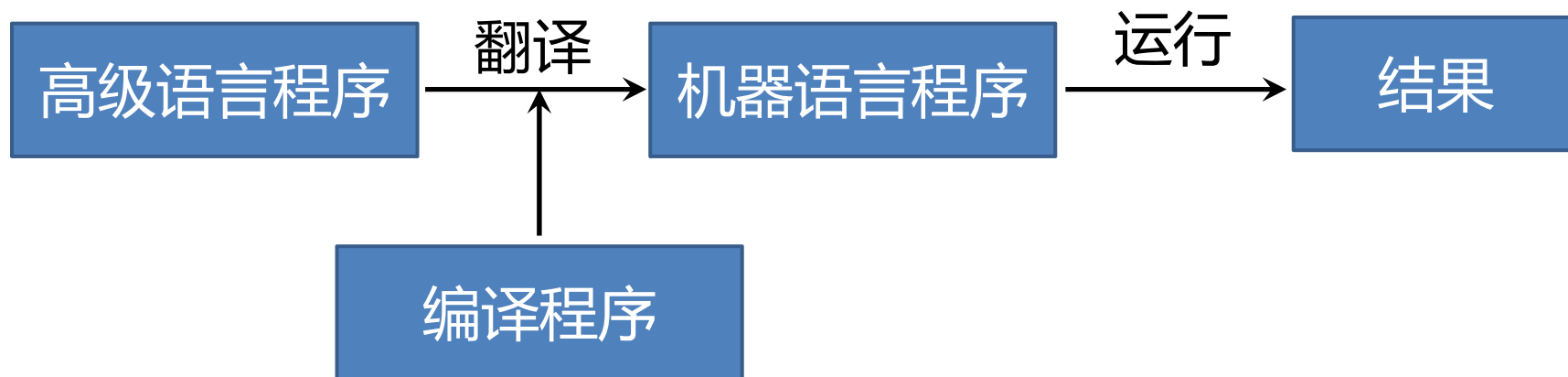
翻译程序(Translator)

- 把某一种语言程序(称为源语言程序)等价地转换成另一种语言程序(称为目标语言程序)的程序



编译程序(Compiler)

- 把某一种高级语言程序等价地转换成另一种低级语言程序(如汇编语言或机器语言程序)的程序



诊断编译程序(Diagnostic Compiler)

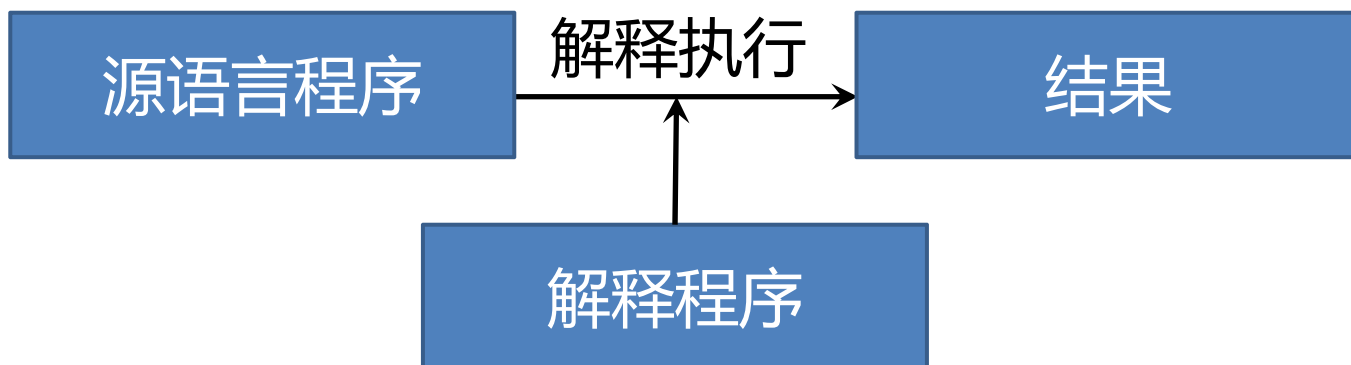
优化编译程序(Optimizing Compiler)

交叉编译程序(Cross Compiler)

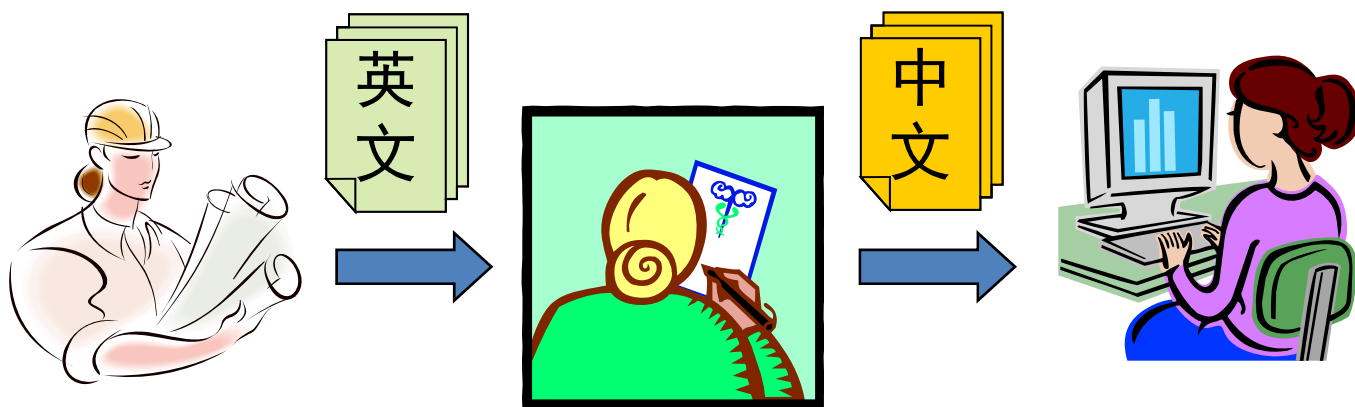
可变目标编译程序(Retargetable Compiler)

解释程序(Interpreter)

- 把源语言写的源程序作为输入，但不产生目标程序，而是边解释边执行源程序



测试：编译 vs. 解释



A.编译 B.解释

测试：编译 vs. 解释



A.编译

B.解释

编译原理

引论——计算思维

引论

- ▶ 什么是编译程序
- ▶ 为什么要学习编译原理
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

学习目的

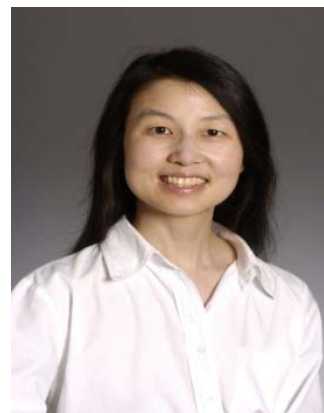
- ▶ 从计算机科学与技术中学什么?
 - ▶ 理解计算系统
 - ▶ 设计计算系统
 - ▶ 训练计算思维(Computational Thinking)

计算思维

► 计算思维基本概念

► Jeannette M. Wing, Computational Thinking, Communications of ACM, Vol.49, No.3, 2006, pp.33-35.

► 被认为是近十年来产生的最具有基础性、长期性的学术思想，成为21世纪计算机科学研究和教育的热点



Viewpoint | Jeannette M. Wing

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking is the thinking process of solving problems by using the concepts and methods of computer science. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

Computational thinking is a process of thinking that is not just about computers, but about the world. It is a process of thinking that is not just about computers, but about the world.

计算思维

- ▶ 计算思维是什么[J. Wing, 2006]
 - ▶ **计算思维**是运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为，它包括了一系列广泛的计算机科学的思维方法
 - ▶ 计算思维和阅读、写作和算术一样，是21世纪每个人的基本技能，而不仅仅属于计算机科学家
 - ▶ 计算思维在生物、物理、化学、经济学、统计学等其他学科中的影响已经显现

计算思维

- ▶ 包括一系列广泛的计算机科学的思维方法
 - ▶ 抽象
 - ▶ 自动化
 - ▶ 问题分解
 - ▶ 递归
 - ▶ 权衡
 - ▶ 保护、冗余、容错、纠错和恢复
 - ▶ 利用启发式推理来寻求解答
 - ▶ 在不确定情况下的规划、学习和调度
 - ▶ ...

编译原理

引论——编译与计算思维

关于编译理论与技术

► 编译理论与技术

- 计算机科学与技术中理论和实践相结合的最好典范

► ACM 图灵奖

- 授予在计算机技术领域作出突出贡献的科学家
- 程序设计语言、编译相关的获奖者是最多的

Analysis of Algorithms **Artificial Intelligence** Combinatorial Algorithms
Compilers
Computational Complexity Computer Architecture
Computer Hardware
Cryptography Data Structures Databases Education
Error Correcting Codes Finite Automata Graphics
Interactive Computing Internet Communications List Processing Numerical Analysis
Numerical Methods Object Oriented Programming Operating Systems Personal Computing
Program Verification Programming
Programming Languages Proof Construction Software
Theory Software Engineering
Verification of Hardware and Software Models Computer Systems Machine Learning
Parallel Computation

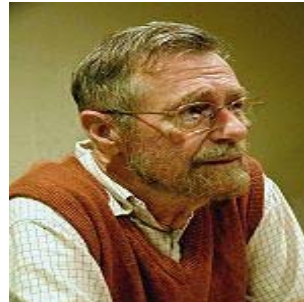


<http://amturing.acm.org/bysubject.cfm>

ACM图灵奖



Alan J. Perlis



Edsger W. Dijkstra



John W. Backus



Kenneth E. Iverson

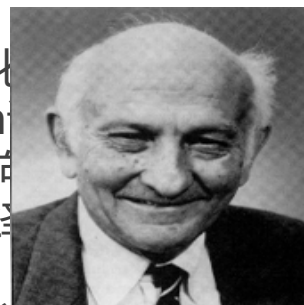
- ▶ Alan J. Perlis (1966) -- ALGOL
- ▶ Edsger Wybe Dijkstra (1972) -- ALGOL
- ▶ Michael O. Rabin & Dana S. Scott (1976) -- 非确定自动机
- ▶ John W. Backus (1977) -- FORTRAN
- ▶ Kenneth Eugene Iverson (1979) -- APL程序语言
- ▶ Niklaus Wirth (1984) -- PASCAL
- ▶ John Cocke (1987) -- RISC & 编译优化
- ▶ O. Dahl, K.Nygaard (2001) -- Simula
- ▶ Alan Kay(2003) -- SmallTalk语言和面向对象
- ▶ Peter Naur(2005) -- ALGOL60以及编译器
- ▶ Frances E. Allen(2006)-- 优化编译器
- ▶ Barbara Liskov(2008)-- 编程语言和系统设计的实践与理论



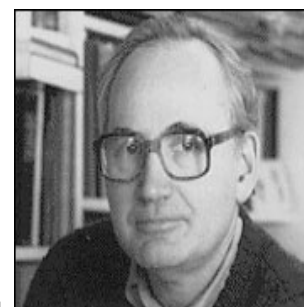
K.Nygaard



O. Dahl



John Cocke



Dana S. Scott



Michael O. Rabin



Niklaus Wirth



Donald E. Knuth



Barbara Liskov



Frances E. Allen



Peter Naur



Alan Kay

关于编译理论与技术

- ▶ 编译理论与技术

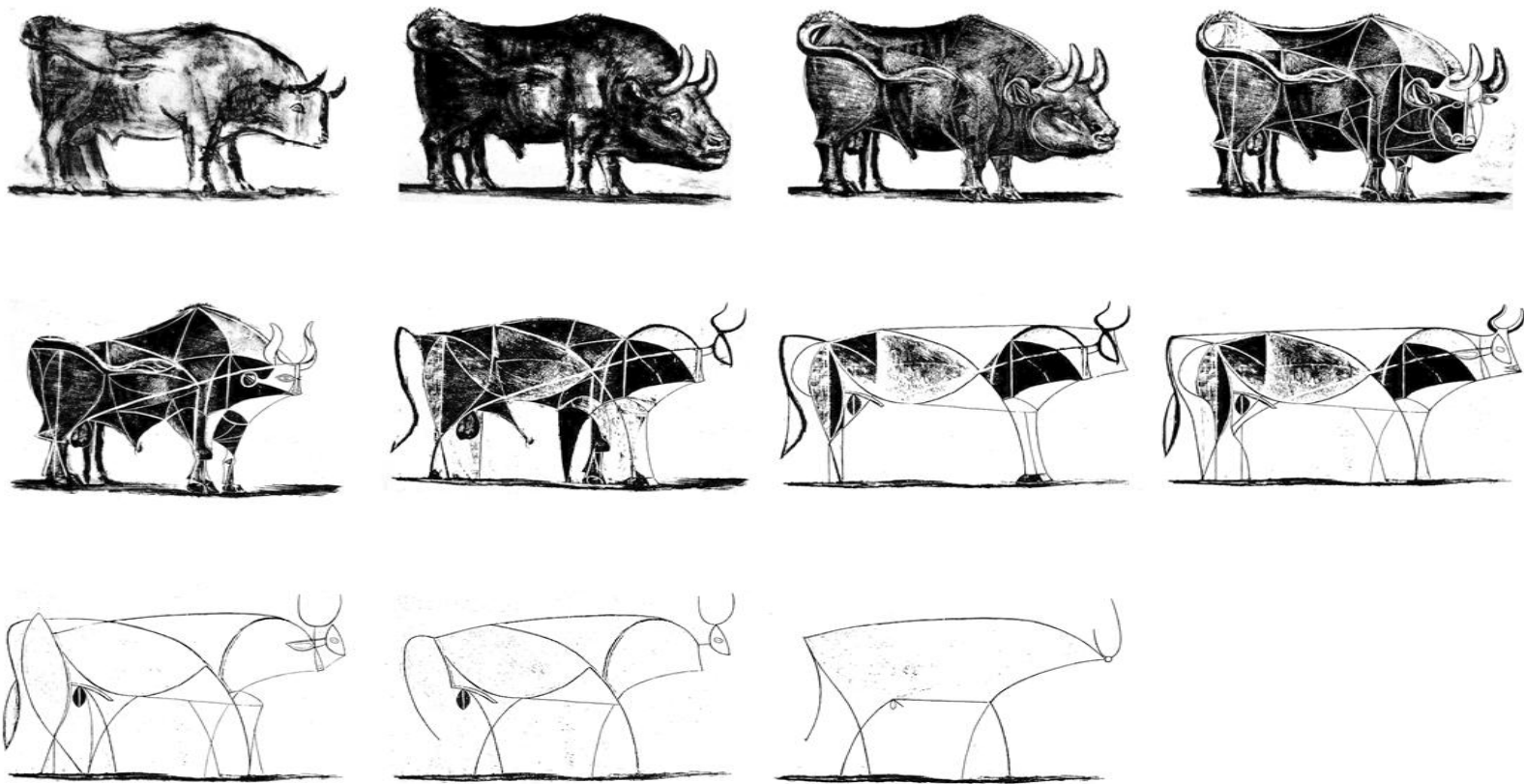
- ▶ 计算机科学与技术中理论和实践相结合的最好典范

- ▶ 体现了很多典型的计算思维方法

计算思维 vs. 编译

▶ 抽象(Abstraction)

- ▶ 忽略一个主题中与当前问题(或目标)无关的那些方面, 以便更充分地注意与当前问题(或目标)有关的方面
- ▶ 从众多的事物中抽取出共同的、本质性的特征, 舍弃其非本质的特征
- ▶ 是一种从个体把握一般、从现象把握本质的认知过程和思维方法

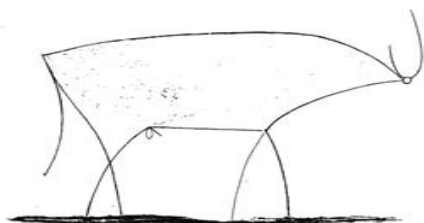


毕加索《牛》

毕加索认为，画家的职责不是借助具体形象反映现实,而是以抽象的形象表达科学的真实。

抽象的例子

► 从抽象的观点看



是对



的抽象

质点

是对

物理对象

的抽象

化学分子式

是对

化合物

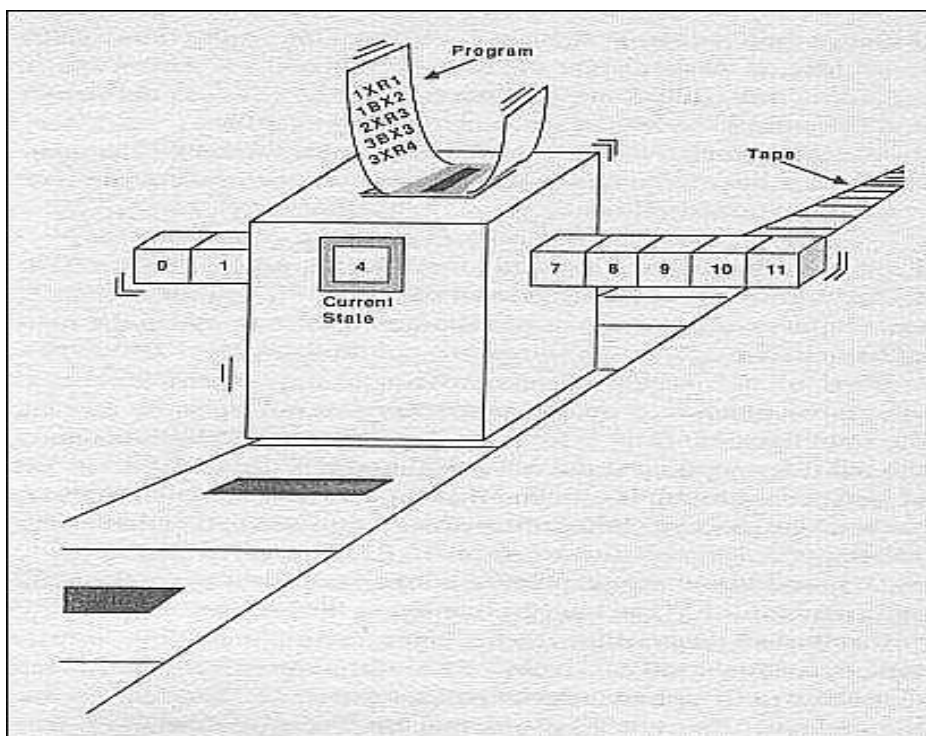
的抽象

计算机科学中的抽象?

计算思维 vs. 编译

► 抽象(Abstraction)

► 图灵机

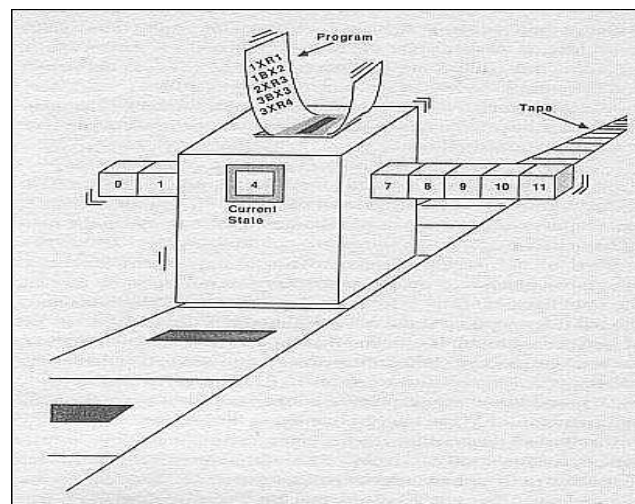


计算思维 vs. 编译

► 抽象(Abstraction)

► 图灵机

- 一条无限长的纸带
- 一个读写头
- 一个状态寄存器
- 一套控制读写头工作的规则



计算思维 vs. 编译

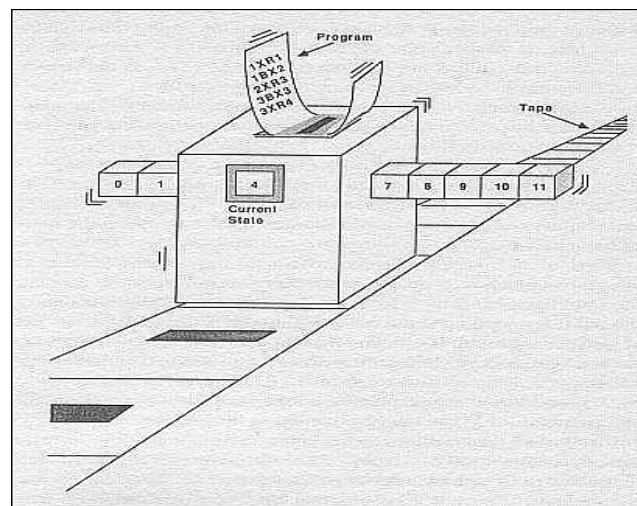
► 抽象(Abstraction)

► 图灵机

► 邱奇-图灵论题(The Church-Turing thesis)

► 所有计算或算法都可以由一台图灵机来执行

► 可计算=图灵可计算



计算思维 vs. 编译

- ▶ 编译原理中的"抽象"

- ▶ 有限自动机

- ▶ 形式文法

- ▶ ...

计算思维 vs. 编译

► 自动化(Automation)

- 将抽象思维的结果在计算机上实现，是一个将计算思维成果物化的过程，也是将理论成果应用于技术的实践
- 自动化的思维方法不仅体现在编译程序本身的工作机制上，更体现在了编译程序的生成工具的研究和设计上

计算思维 vs. 编译

▶ 编译原理中的“自动化”

- ▶ 有限自动机
- ▶ 预测分析程序
- ▶ 算符优先分析
- ▶ LR分析
- ▶ ...

控制程序

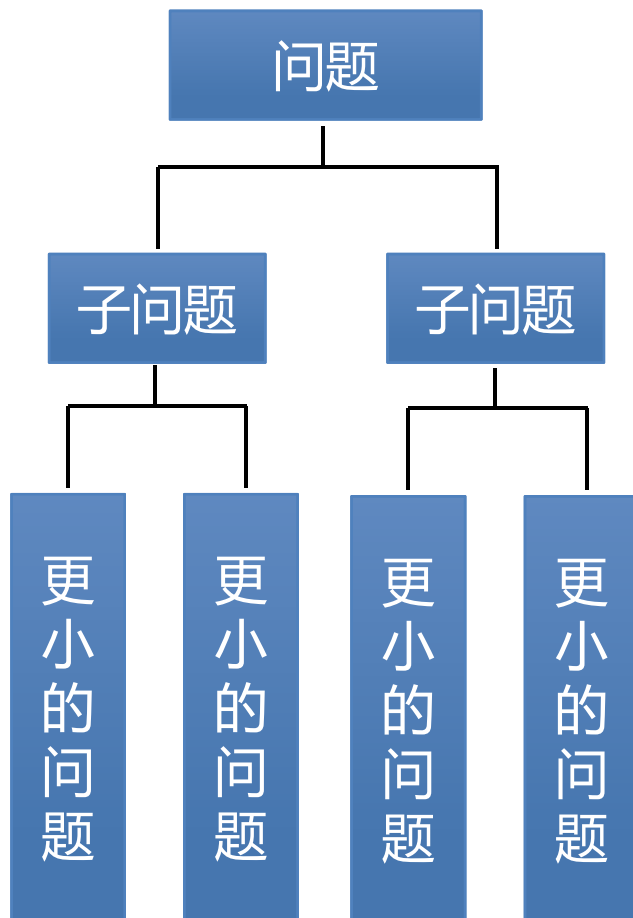
分析表

计算思维 vs. 编译

► 分解(Decomposition)

► 将大规模的复杂问题分解成若干个较小规模的、更简单的问题加以解决

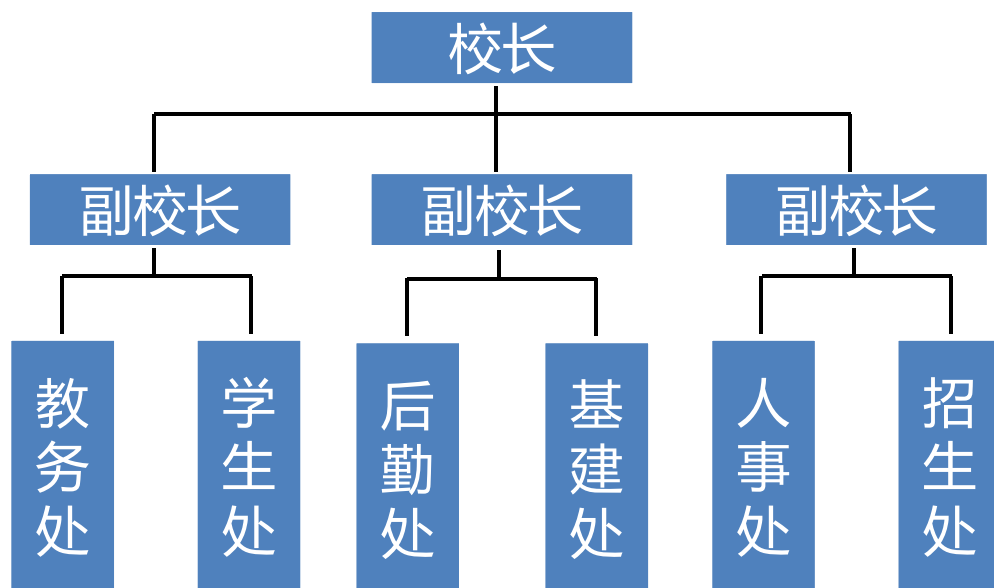
- 对问题本身的明确描述，并对问题解法作出全局性决策
- 把问题分解成相对独立的子问题
- 再以同样的方式对每个子问题进一步精确化，直到获得对问题的明确的解答



计算思维 vs. 编译

► 分解(Decomposition)

► 层次化管理



计算思维 vs. 编译

▶ 编译原理中的"问题分解"

- ▶ 为什么编译程序引入中间语言?
- ▶ 为什么编译分成多个阶段?
- ▶ 为什么分析过程分成多遍?
- ▶ ...

计算思维 vs. 编译

▶ 递归(Recursion)

- ▶ 问题的解决依赖于类似问题的解决，只不过后者的复杂程度或规模较原来的问题更小
- ▶ 一旦将问题的复杂程度和规模化简到足够小时，问题的解法其实非常简单

计算思维 vs. 编译

- ▶ 编译原理中的"递归"
 - ▶ 递归下降分析
 - ▶ 基于树遍历的属性计算
 - ▶ 语法制导翻译
 - ▶ ...

计算思维 vs. 编译

▶ 权衡(折衷, Tradeoff)

- ▶ 理论可实现 vs. 实际可实现

- ▶ 理论研究重在探寻问题求解的方法，对于理论成果的研究运用又需要在能力和运用中作出权衡

计算思维 vs. 编译

- ▶ 编译原理中的"权衡"

- ▶ 用上下文无关文法来描述和处理高级程序设计语言
- ▶ 优化措施的选择
- ▶ ...

计算思维 vs. 编译

- ▶ 计算思维包括一系列计算机科学的思维方法

- ▶ 抽象

- ▶ 自动化

- ▶ 问题分解

- ▶ 递归

- ▶ 权衡

- ▶ ...

学习编译原理
训练计算思维
享受计算之美

学习编译原理的意义

- ▶ 学习编译程序构造原理，技术
 - ▶ 提高对计算机系统总体认识
 - ▶ 感悟计算思维
 - ▶ 更好地理解“计算”
- ▶ 更好地理解高级语言
- ▶ 运用编译原理和方法构造实用工具
 - ▶ 用“计算”的眼光看世界
 - ▶ 用计算解决实际问题

编译原理

引论——编译原理和方法的应用

编译原理和方法的应用

► Html/XML 分析

HTML 语言教程：目录	
新的特色 *** HTML 标记(Tag)的索引(Index)	
页面(Page)	字体(Font) 文字布局(Text Style) 图象(Image) 表单(Form)
	表格(Table) 表格进阶(Advanced) 多窗口页面(Frames)
	会移动的文字(Marquee) 多媒体页面(Alternative Inline Elements)
详细目录	
• 页面	文件结构 语言字符集信息 背景色彩和文字色彩 页面空白 链接 开新窗口 标尺线
• 字体	标题 字号 物理字体 逻辑字体

编译原理和方法的应用

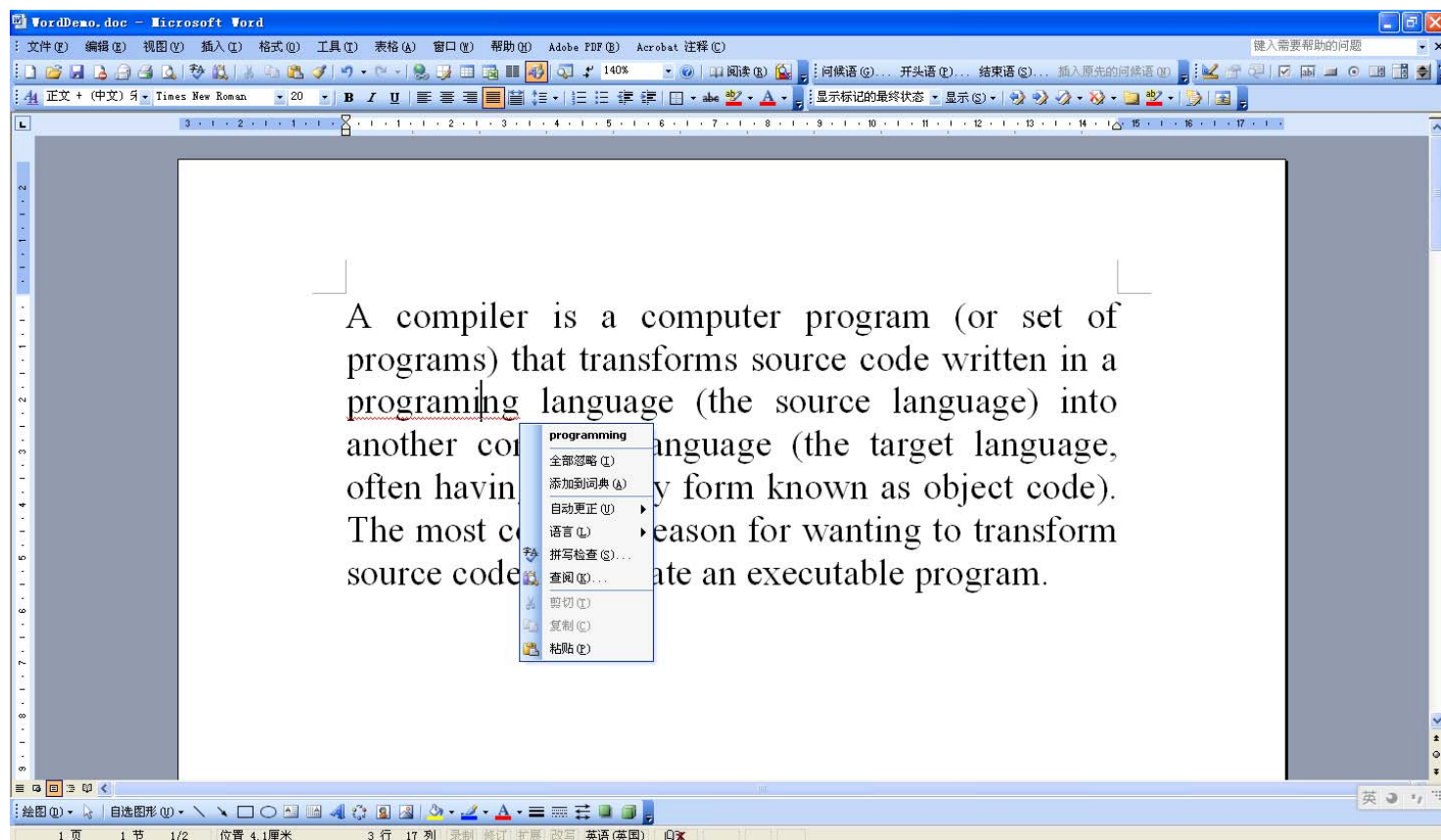
► Html/XML 分析

```
<HTML><HEAD><TITLE>HTML 语言教程</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<META content="MSHTML 6.00.6000.16705" name=GENERATOR></HEAD>
<BODY text=#000000 bgColor=#ffffff>
<H1 align=center><FONT size=+3>HTML 语言教程: 目录</FONT></H1>
<P>
<HR>

<P></P>
<CENTER>
<P><A href="http://www.gzsums.edu.cn/webclass/html/new.html">新的特色</A> *** <A |
href="http://www.gzsums.edu.cn/webclass/html/tag-index.html">HTML
标记 (Tag) 的索引 (Index) </A> </P></CENTER>
<CENTER><A
href="http://www.gzsums.edu.cn/webclass/html/page.html">页面 (Page) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/font.html">字体 (Font) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/text_style.html">文字布局 (Text
Style) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/image.html">图象 (Image) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/form.html">表单 (Form) </A>
<P><A href="http://www.gzsums.edu.cn/webclass/html/table.html">表格 (Table) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/table02.html">表格进阶 (Advanced) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/frame.html">多窗口页面 (Frames) </A>
<P><A
href="http://www.gzsums.edu.cn/webclass/html/marquee.html">会移动的文字 (Marquee) </A>      <A
href="http://www.gzsums.edu.cn/webclass/html/inline.html">多媒体页面 (Alternative
Inline Elements) </A> </CENTER>
<P>
```

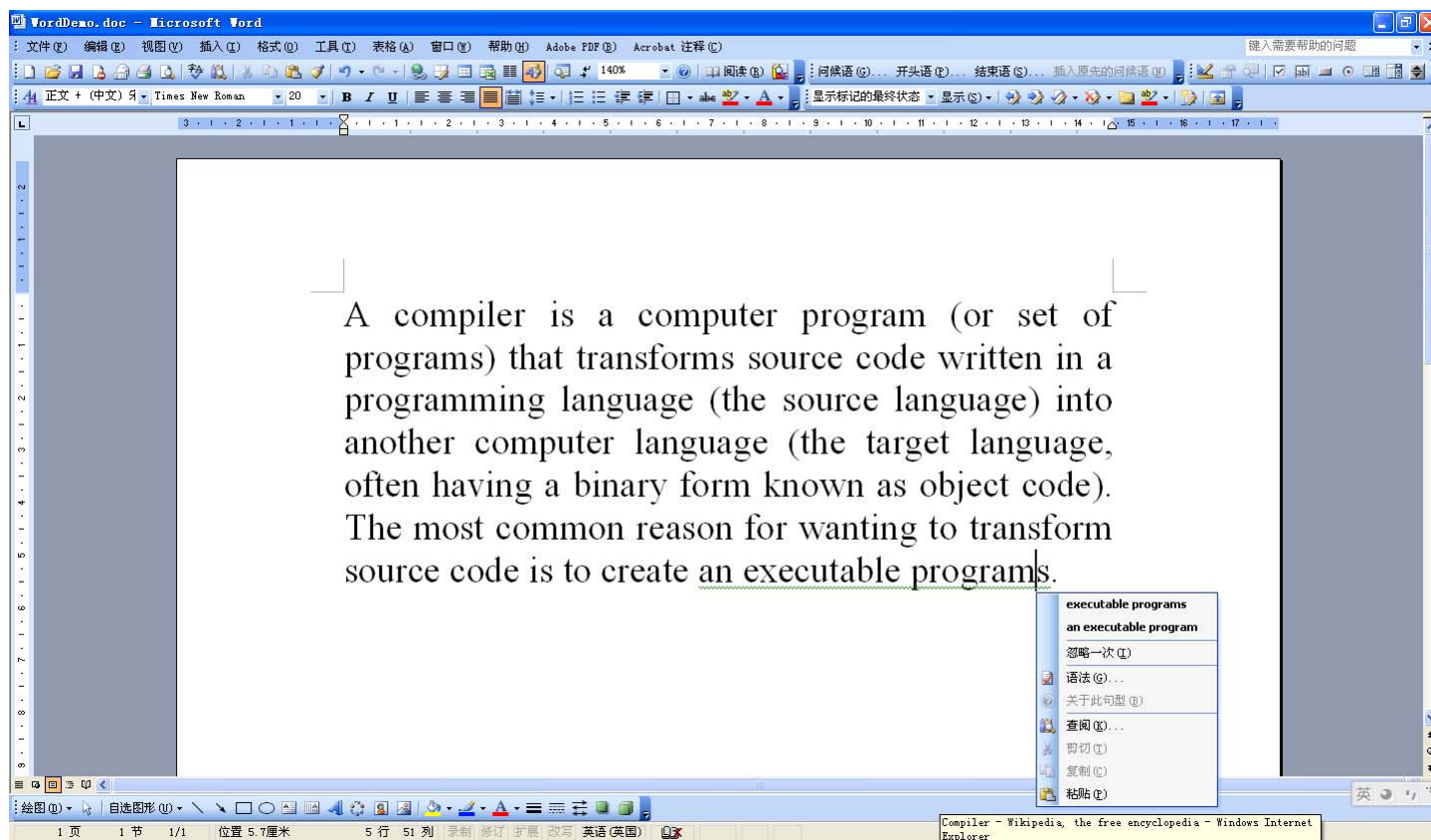
编译原理和方法的应用

► 语言处理工具



编译原理和方法的应用

► 语言处理工具



编译原理和方法的应用

► 语言处理工具



编译原理和方法的应用

► 搜索引擎

中国奥运金牌数是多少? x

← → C https://www.baidu.com/s?word=中国奥运金牌数是多少?&ttn=98979085_hao_pg 百度一下 百度首

Baidu 百度 中国奥运金牌数是多少?

网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»

百度为您找到相关结果约2,840,000个 搜索工具

问 中国奥运金牌数是多少? :

- [历届奥运会中国金牌数_百度知道](#)
第二十三届洛杉矶奥运会(1984年)中国15金第二十四届汉城奥运会(1988年)中国5金第二十五届巴塞罗那奥运会(1992年)中国16金第二十六届亚特兰大奥运会(...)
来自百度知道 | 1个回答 | 2006-03-09
- [2012奥运会中国金牌数奖牌总数及排名是多少?_百度知道](#)
中国共38金27银23铜,居第二. 2012年伦敦奥运会奖牌榜 排名 国家/地区 金牌 银牌 铜牌总数 1 美国 46 29 29 104 2 中国 38 27 23 88 3 英国 29 17 19 65 4 俄...
来自百度知道 | 6个回答 | 2012-09-08
- [雅典奥运会中国金牌数是多少?_百度知道](#)
第28届2004年雅典奥运会 获金牌32枚,累计112枚;排名第2 中国奥运第81金:射击女子10米气步枪(杜丽) 中国奥运第82金:射击男子10米气手枪(王义夫) 中国奥运第83金...
来自百度知道 | 1个回答 | 2014-09-30

[2008中国奥运会金牌总数是多少_百度知道](#)

相关人物 展开 ▼

 孙杨	 柳金	 许海峰
-------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------

相关运动员 展开 ▼

 刘翔	 廖辉	 杨威
---------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------

其他人还搜 展开 ▼

		
---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------

测试

- ▶ 请列举在哪些计算机技术中应用了语言的翻译或变换
 - ▶ 用户接口：Shell命令解释器， ...
 - ▶ 查询语言：SQL, XQuery, ...
 - ▶ 网络协议：HTTP, SOAP, ...
 - ▶ ...

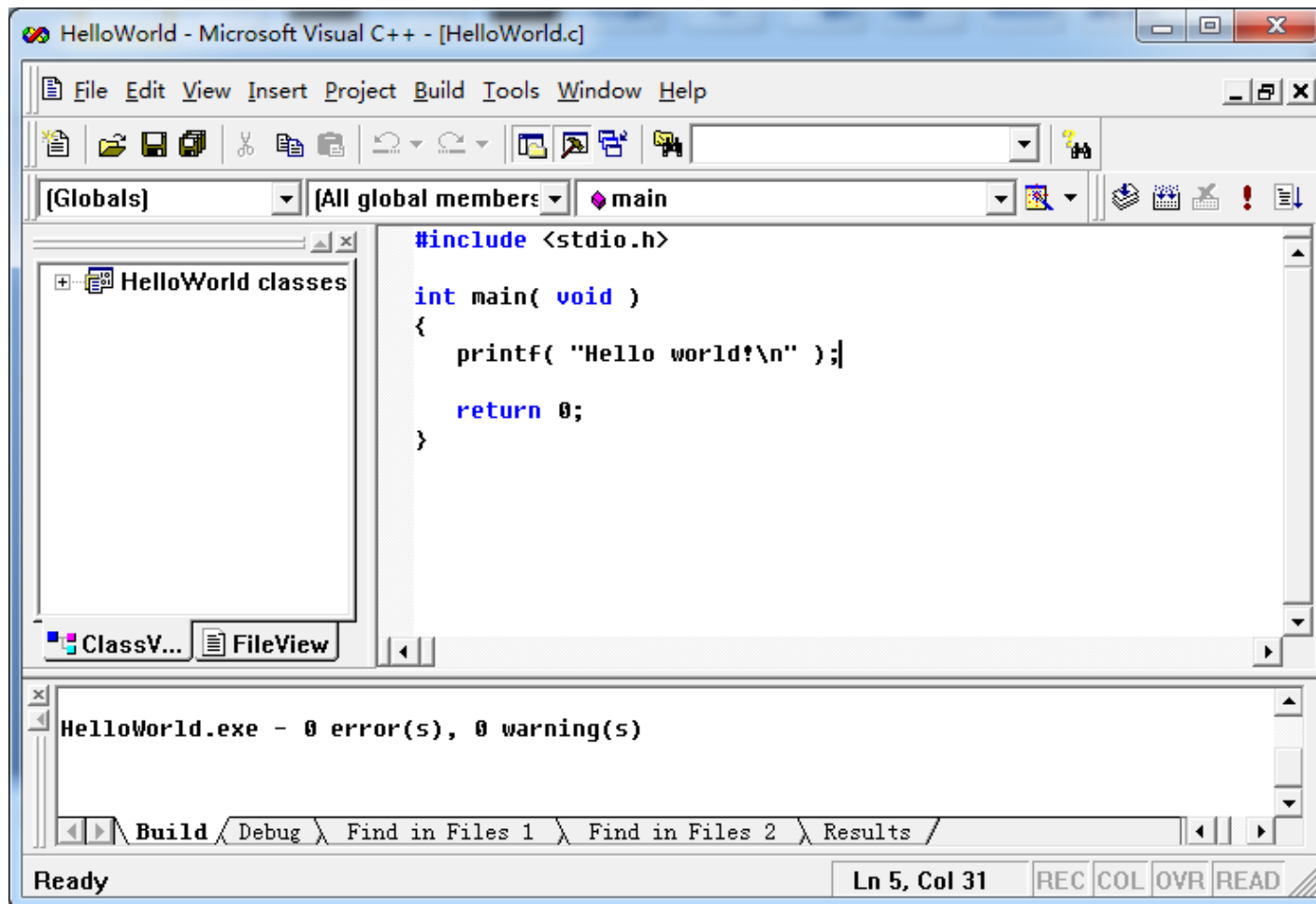
编译原理

引论——编译过程

引论

- ▶ 什么是编译程序
- ▶ 为什么要学习编译原理
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

Hello World! 的背后



编译过程

- ▶ 编译程序是怎样把高级语言(如C++)翻译成低级语言(如机器指令)的?

编译过程

The compiler can translate a program from source language to target language.



编译程序能够把一个程序从源语言翻译成目标语言。

► 把英文翻译为中文

► 识别出句子中的一个单词

词法分析

► 分析句子的语法结构

语法分析

► 根据句子的含义进行初步翻译

中间代码产生

► 对译文进行修饰

优化

► 写出最后的译文

目标代码产生

编译程序工作的五个阶段

编译过程——词法分析

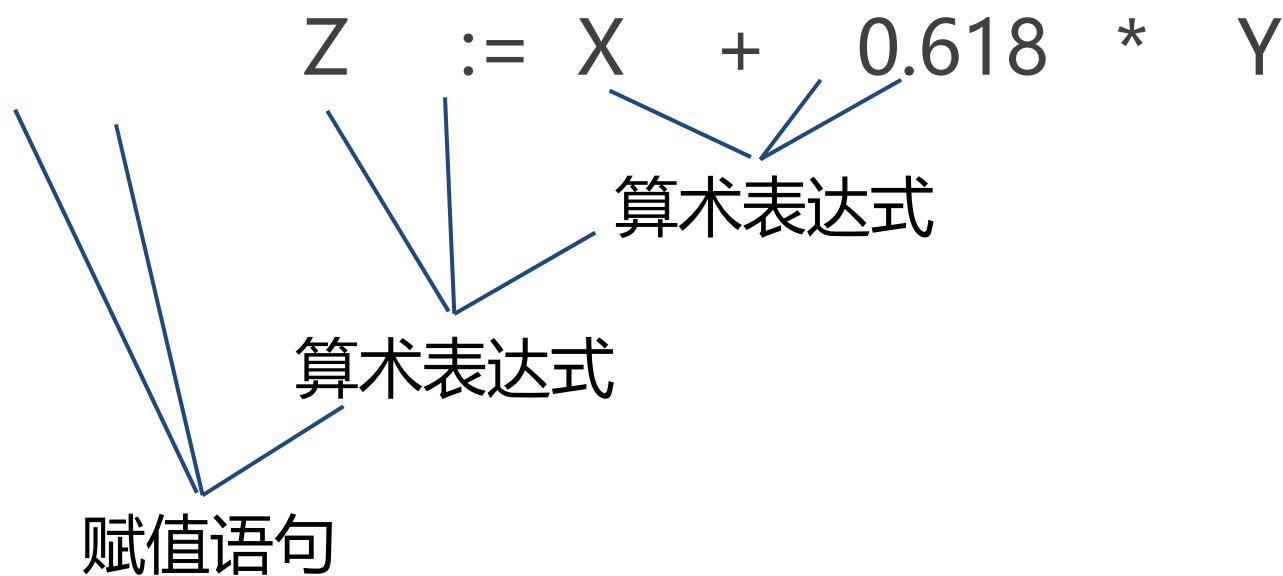
- ▶ 任务: 输入源程序, 对构成源程序的字符串进行扫描和分解, 识别出单词符号
- ▶ 依循的原则: 构词规则
- ▶ 描述工具: 有限自动机

for i := 1 to 100 do

基本字 标识符 赋值号 整常数 基本字 整常数 基本字

编译过程——语法分析

- ▶ 任务：在词法分析的基础上，根据语法规则把单词符号串分解成各类语法单位(语法范畴)
- ▶ 依循的原则：语法规则
- ▶ 描述工具：上下文无关文法



编译过程——中间代码产生

- ▶ 任务：对各类语法单位按语言的语义进行初步翻译
- ▶ 依循的原则：语义规则
- ▶ 描述工具：属性文法
- ▶ 中间代码：三元式，四元式，树，...
- ▶ $Z := X + 0.618 * Y$ 翻译成四元式为

序号	OPR	OPN1	OPN2	RESULT	注释
(1)	*	0.618	Y	T1	$T1 := 0.618 * Y$
(2)	+	X	T1	T2	$T2 := X + T1$
(3)	:=	T2		Z	$Z := T2$

编译过程——优化

- ▶ 任务：对前阶段产生的中间代码进行加工变换，以期在最后阶段产生更高效的目标代码
- ▶ 依循的原则：程序的等价变换规则

```
FOR K:=1 TO 100 DO  
BEGIN  
    X:=I+1;  
    M := I + 10 * K;  
    N := J + 10 * K;  
END
```

编译过程：优化

```
FOR K:=1 TO 100 DO  
BEGIN  
    X:=I+1;  
    M := I + 10 * K;  
    N := J + 10 * K;  
END
```

序号	OPR	OPN1	OPN2	RESULT	注释
(1)	:=	1		K	K:=1
(2)	j<	100	K	(10)	if (100<K) goto (10)
(3)	+	I	1	X	X:=I+1
(4)	*	10	K	T1	T1=10*K
(5)	+	I	T1	M	M:=I+T1
(6)	*	10	K	T2	T2=10*K
(7)	+	J	T2	N	N:=J+T2
(8)	+	K	1	K	K:=K+1
(9)	J				goto (2)
(10)					

400次加法
200次乘法

编译过程： 优化后的等价代码

```
FOR K:=1 TO 100 DO  
BEGIN  
    X:=I+1;  
    M := I + 10 * K;  
    N := J + 10 * K;  
END
```

序号	OPR	OPN1	OPN2	RESULT	注释
(1)	+	I	1	X	X:=I+1
(2)	:=	I		M	M:=I
(3)	:=	J		N	N:=J
(4)	:=	1		K	K:=1
(5)	j<	100	K	(10)	if (100<K) goto (10)
(6)	+	M	10	M	M:=M+10
(7)	+	N	10	N	N:=N+10
(8)	+			K	K:=K+1
(9)				(5)	goto (5)
(10)					

301次加法

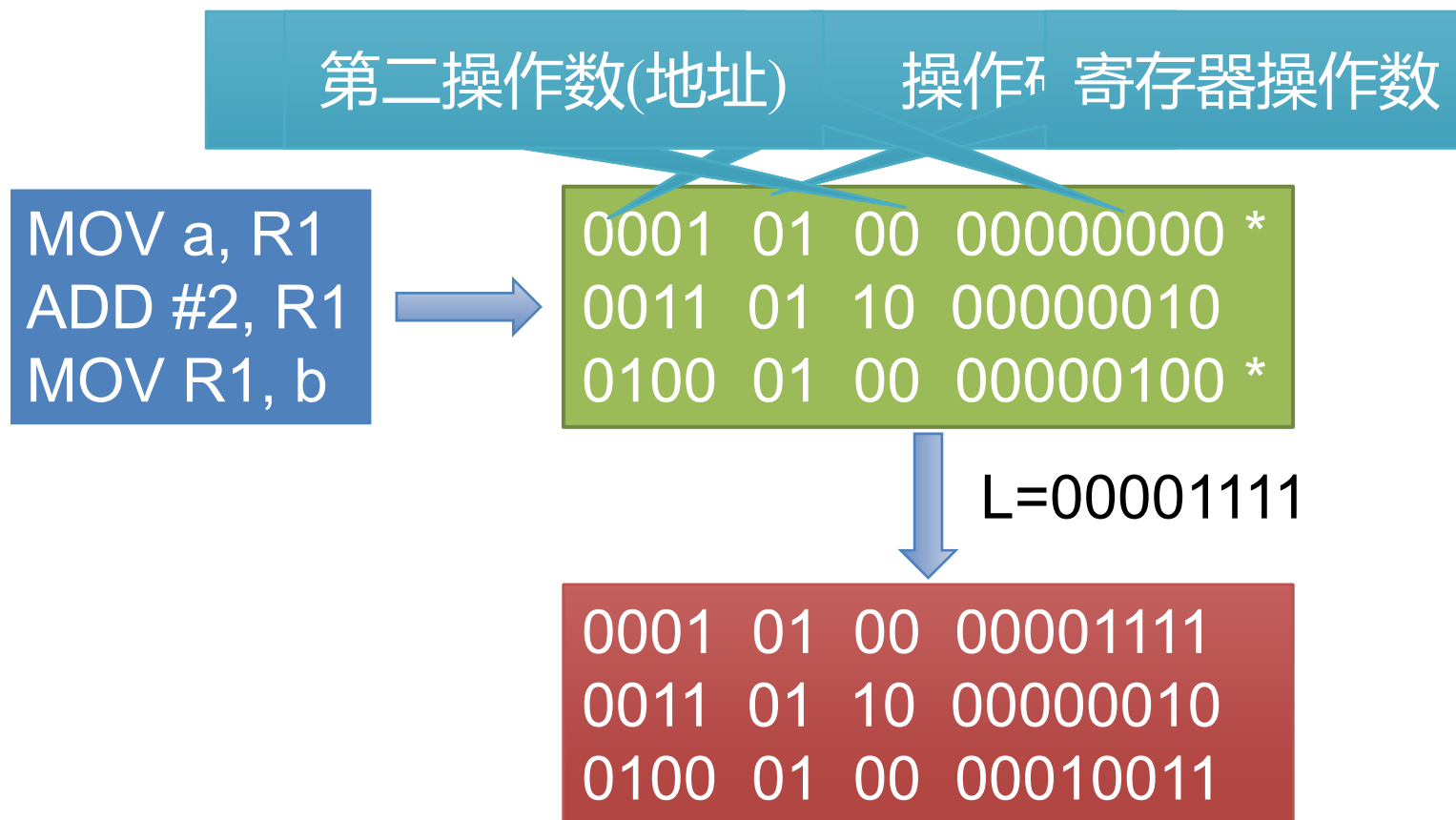
编译过程——目标代码产生

- ▶ 任务: 把中间代码变换成特定机器上的目标代码
- ▶ 依赖于硬件系统结构和机器指令的含义
- ▶ 目标代码三种形式
 - ▶ 汇编指令代码: 需要进行汇编
 - ▶ 绝对指令代码: 可直接运行
 - ▶ 可重新定位指令代码: 需要连接



编译过程——目标代码产生

► 例: $b = a + 2$



测试

► 可以直接运行的目标代码是 ()

A.汇编指令代码

B.可重新定位指令代码

C.绝对指令代码

编译原理

引论——编译程序的结构

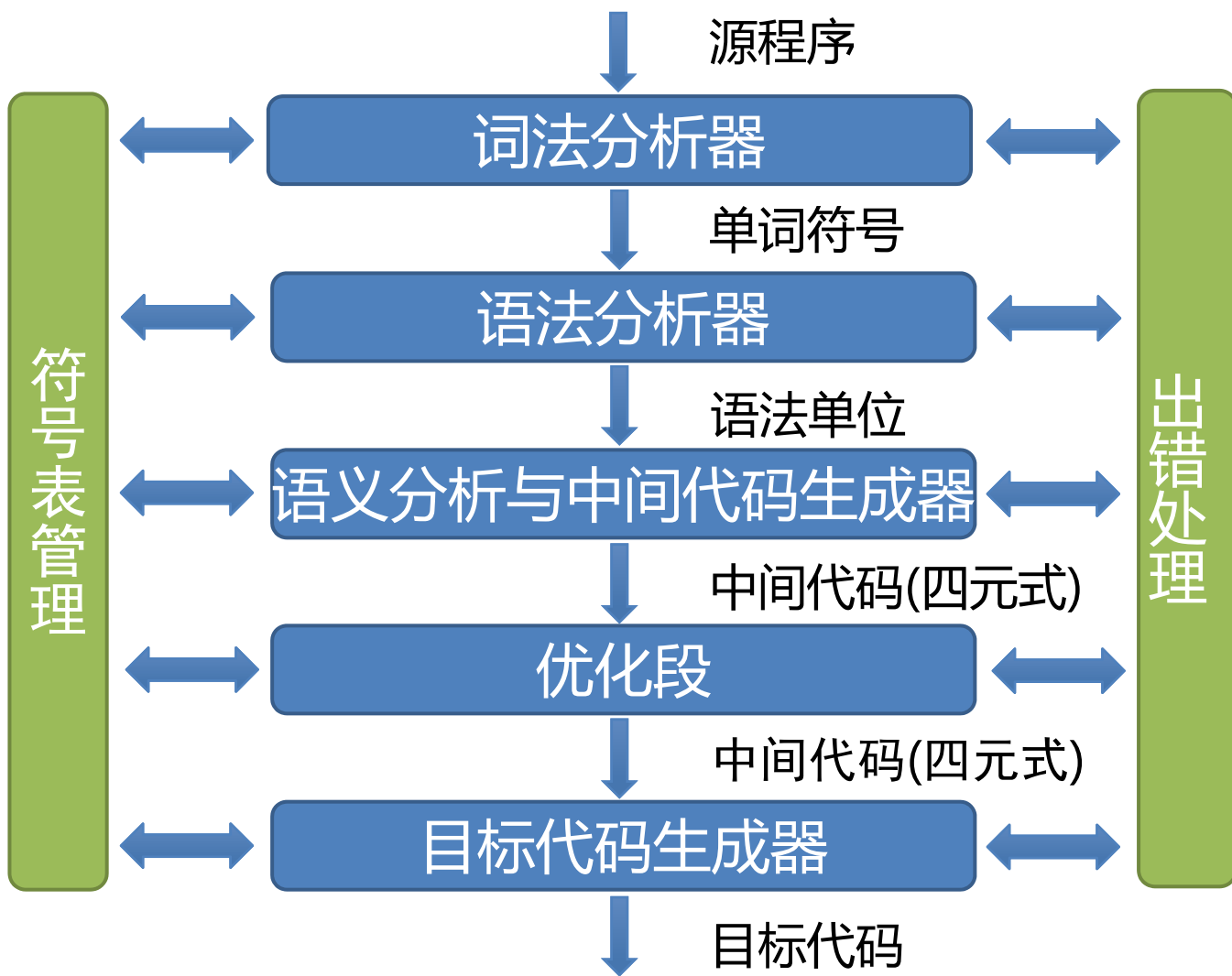
第一章 引论

- ▶ 什么是编译程序
- ▶ 为什么要学习编译原理
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

编译程序的结构

- ▶ 编译程序总框
- ▶ 遍
- ▶ 编译前端与后端

编译程序总框



出错处理

▶ 出错处理程序

- ▶ 发现源程序中的错误，把有关错误信息报告给用户

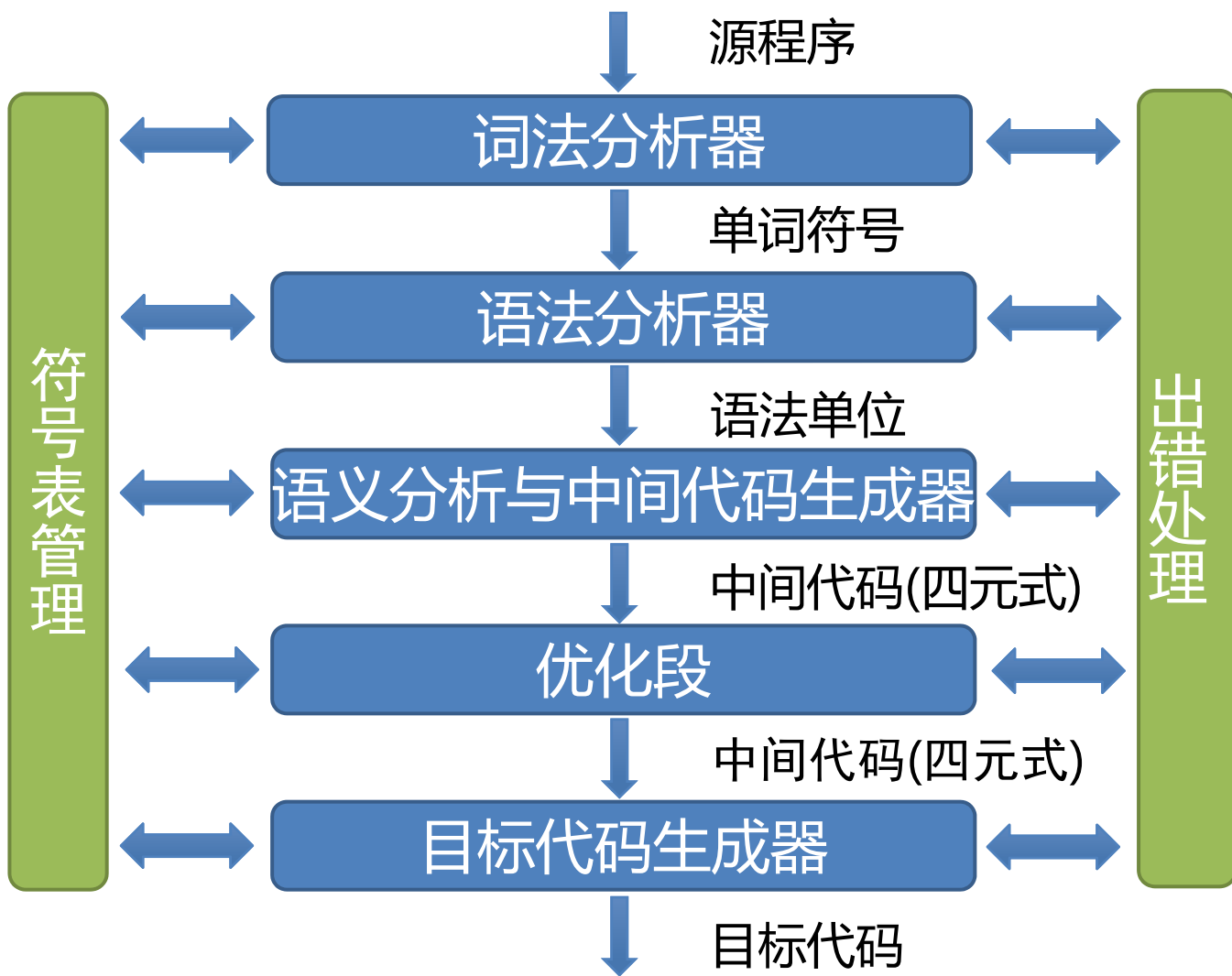
▶ 语法错误

- ▶ 源程序中不符合语法（或词法）规则的错误
- ▶ 非法字符、括号不匹配、缺少；、...

▶ 语义错误

- ▶ 源程序中不符合语义规则的错误
- ▶ 说明错误、作用域错误、类型不一致、...

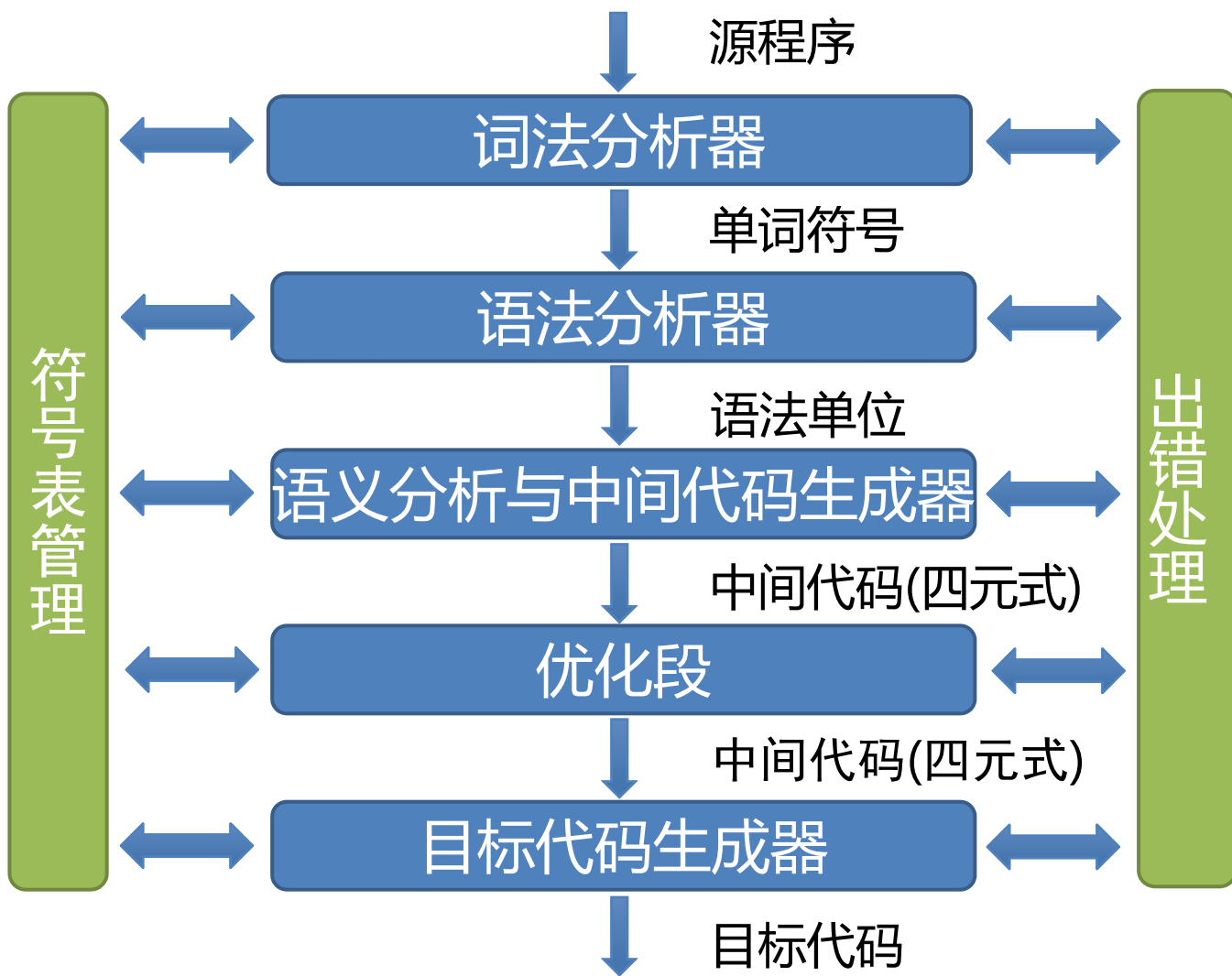
编译程序总框



遍(pass)

- ▶ 所谓"遍", 就是对源程序或源程序的中间表示从头到尾扫描一次
- ▶ 阶段与遍是不同的概念
 - ▶ 一遍可以由若干段组成
 - ▶ 一个阶段也可以分若干遍来完成

编译程序总框



- ▶ 计算思维
 - ▶ 分解
 - ▶ 权衡

编译前端与后端



▶ 编译前端

- ▶ 与源语言有关，如词法分析，语法分析，语义分析与中间代码产生，与机器无关的优化

▶ 编译后端

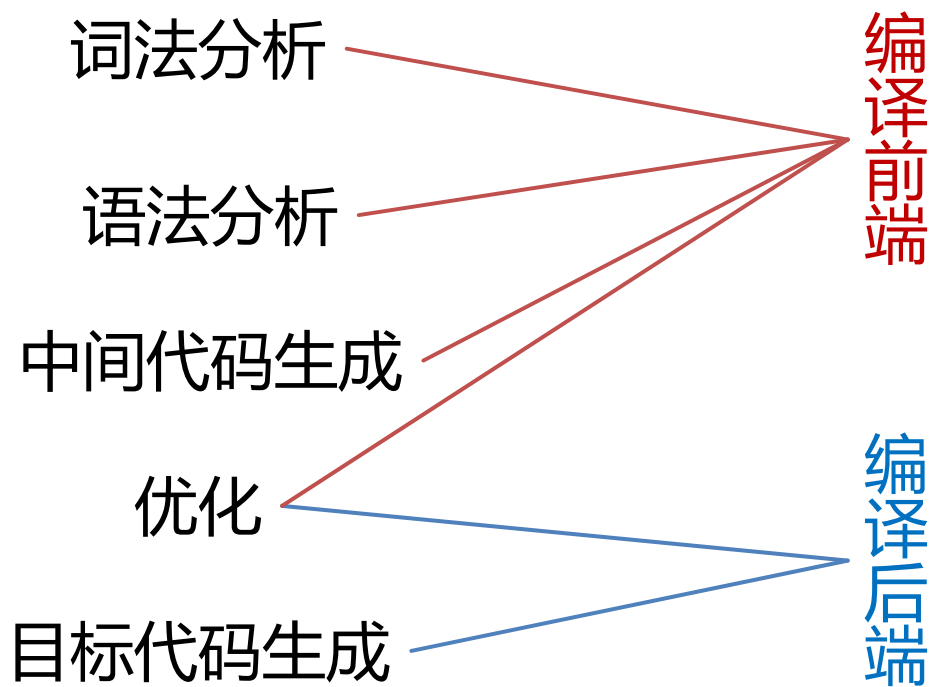
- ▶ 与目标机有关，与目标机有关的优化，目标代码产生

▶ 带来的好处

- ▶ 程序逻辑结构清晰
- ▶ 优化更充分，有利于移植

测试

- ▶ 请指出编译过程和编译程序结构之间的关系



编译原理

引论——编译程序的生成

第一章 引论

- ▶ 什么是编译程序?
- ▶ 为什么要学习编译原理?
- ▶ 编译过程
- ▶ 编译程序的结构
- ▶ 编译程序的生成

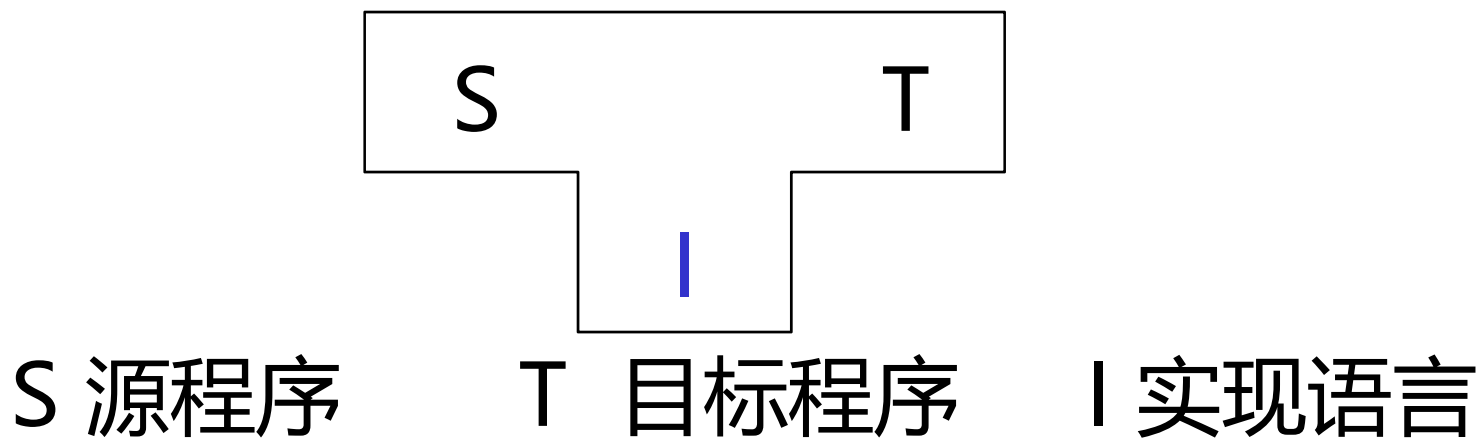
编译程序生成

▶ 以汇编语言和机器语言为工具

- ▶ 优点: 可以针对具体的机器, 充分发挥计算机的系统功能; 生成的程序效率高
- ▶ 缺点: 程序难读、难写、易出错、难维护、生产的效率低

编译程序生成

► 高级语言书写

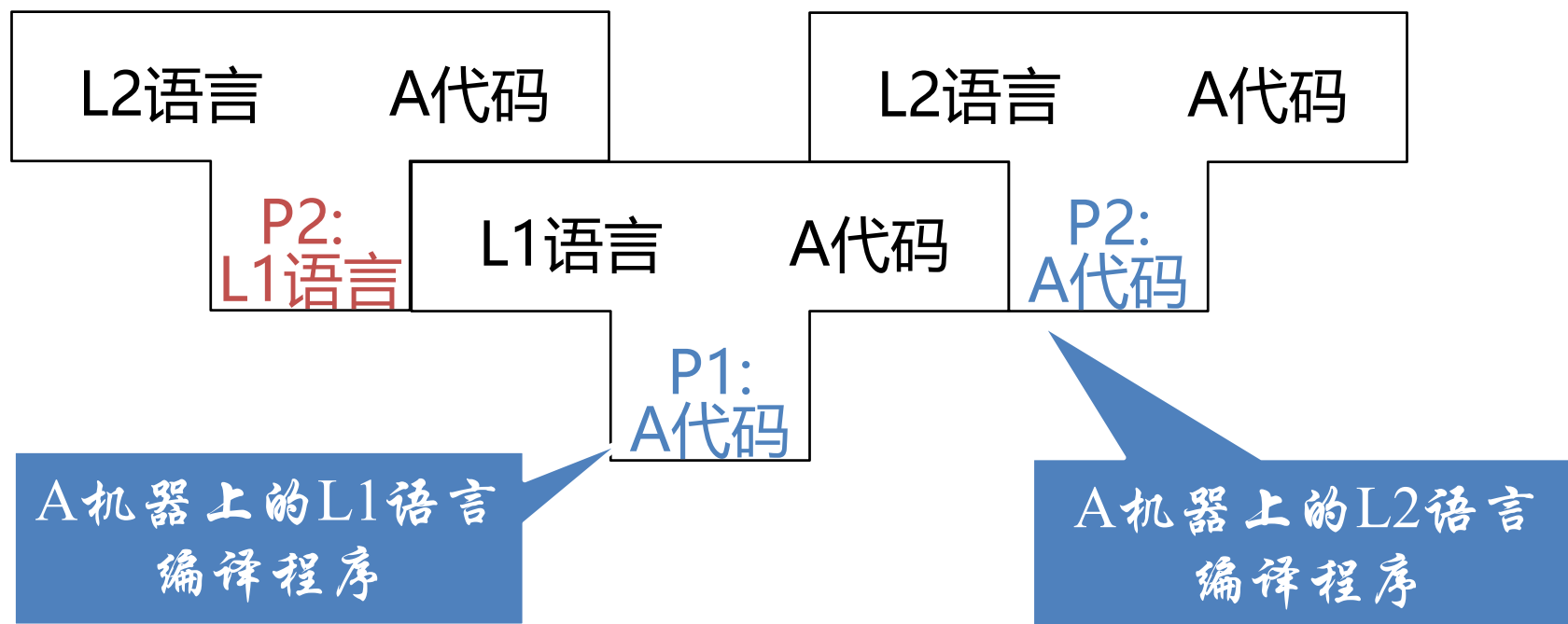


► 程序易读、易理解、容易维护、生产的效率高

编译程序生成

► 高级语言书写

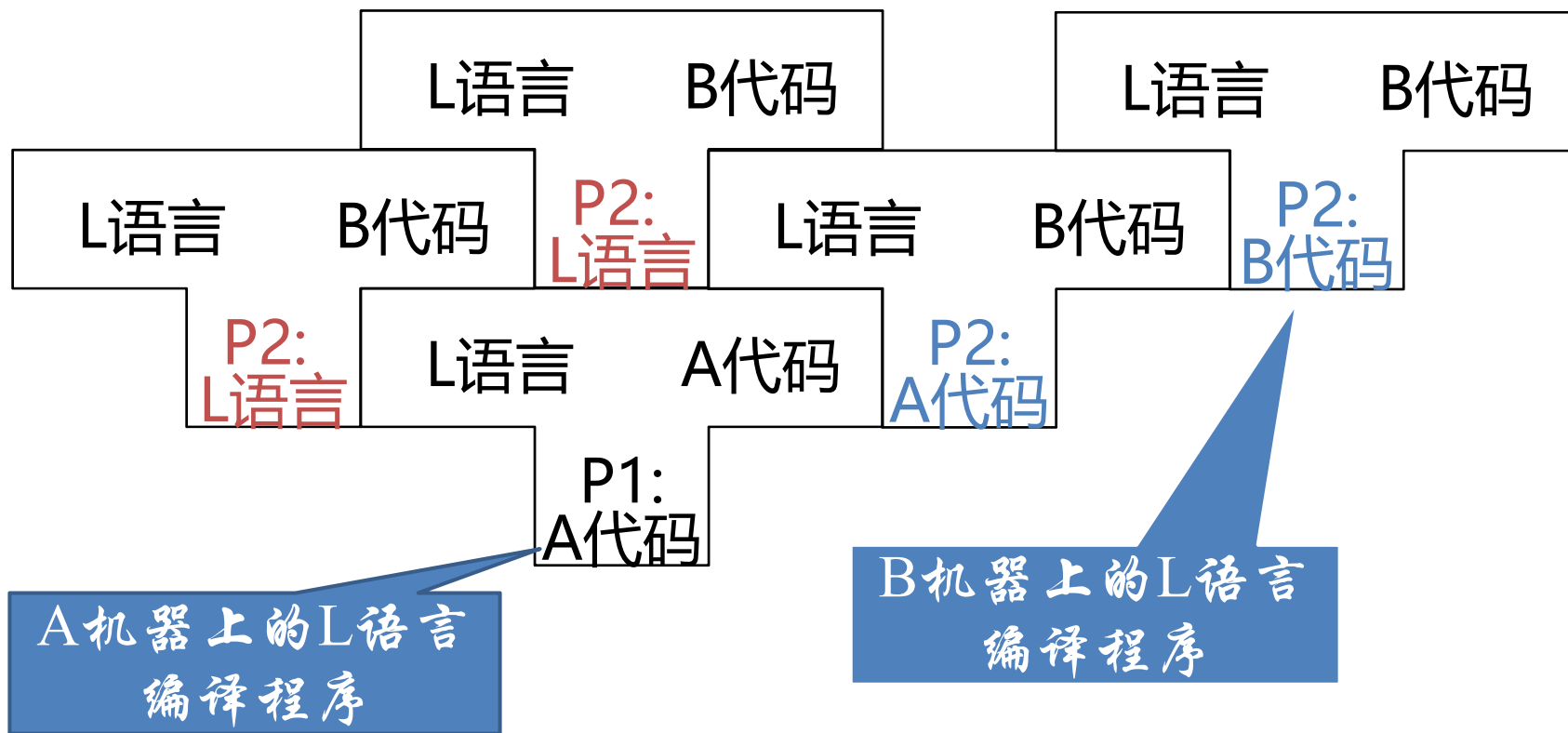
- 利用已有的某种语言的编译程序实现另一语言的编译程序



编译程序生成

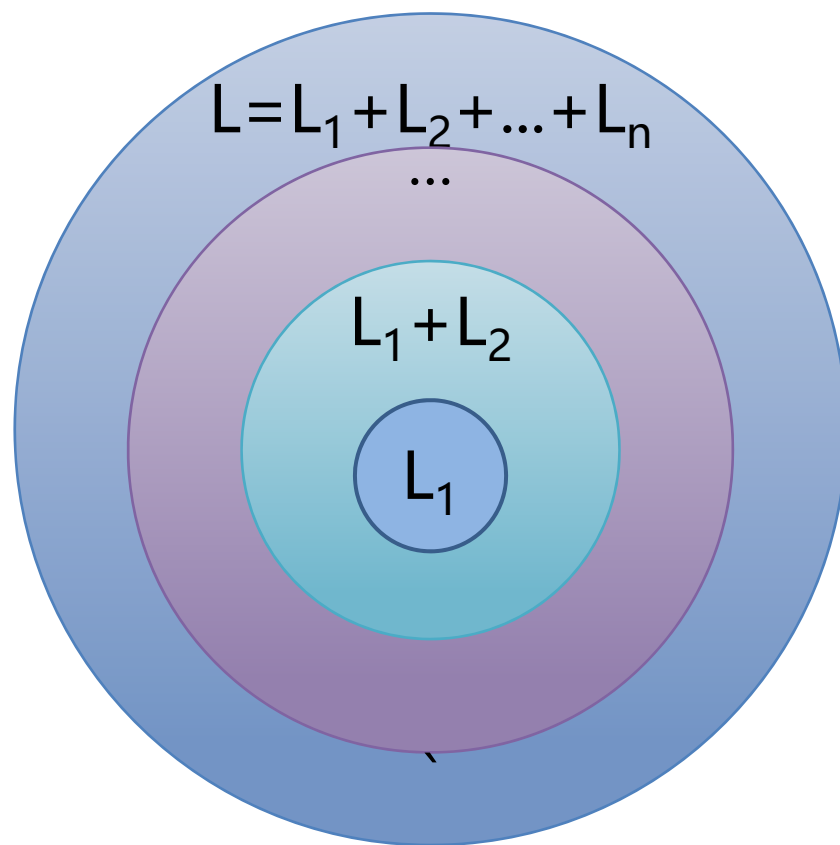
► 移植方法

- 把一种机器上的编译程序移植到另一种机器上



编译程序生成

► 自编译方式



编译程序生成

► 编译程序自动产生

- 编译程序-编译程序，编译程序产生器，编译程序书写系统
- LEX：词法分析程序产生器
- YACC：语法分析程序产生器



小结

- ▶ 课程概述
 - ▶ 内容、意义
- ▶ 什么是编译程序
 - ▶ 翻译、编译、解释
- ▶ 编译基本过程
- ▶ 编译程序的结构
 - ▶ 阶段、遍、前端/后端
- ▶ 编译程序生成的几种方法