

编译原理

第一章 引论

方徽星

扬州大学 信息工程学院(505)

fanghuixing@yzu.edu.cn

2018年春季学期

课程基本情况

参考书：

1. 薛联凤、秦振松，《编译原理及编译程序构造》（第二版），东南大学出版社，2013
2. 吕映芝、张素琴等编著，《编译原理》，清华大学出版社，2005
3. 陈火旺等编著，《程序设计语言与编译原理（第3版）》，国防工业出版社，2006
4. Alfred V. Ah等：《编译原理(本科教学版)》（第2版），译者：赵建华，郑滔，戴新宇，2009

课程考核：闭卷笔试

1. 选择题（15个× 2分/个，共30分，难度：★ ★ ）
2. 判断题（10个× 2分/个，共20分，难度：★ ★ ★ ）
3. 简答题（3个×10分/个，共30分，难度：★ ★ ★ ★ ）
4. 问答题（1个×20分/个，共20分，难度：★ ★ ★ ★ ★）

成绩评定：考试成绩（70%）+ 平时成绩（30%）

本章主要内容

一．程序设计语言与编译

二．编译程序概述

三．编译程序的构造

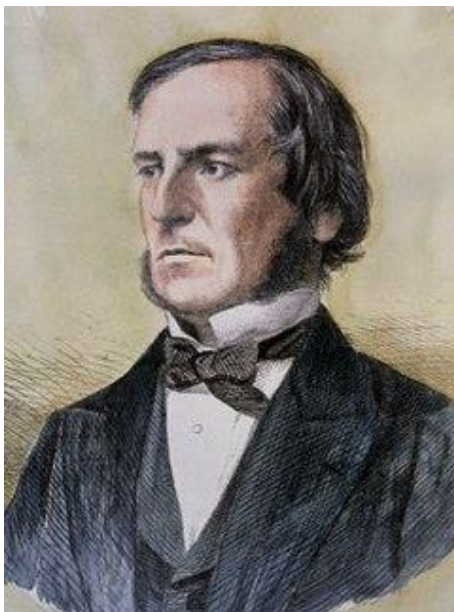
第一节 程序设计语言与编译

1.1 程序设计语言



莱布尼茨（1703年）

《论只使用符号0和1的**二进制算术**，兼论其用途及它赋予伏羲所使用的古老图形的意义》



乔治·布尔（1854年）

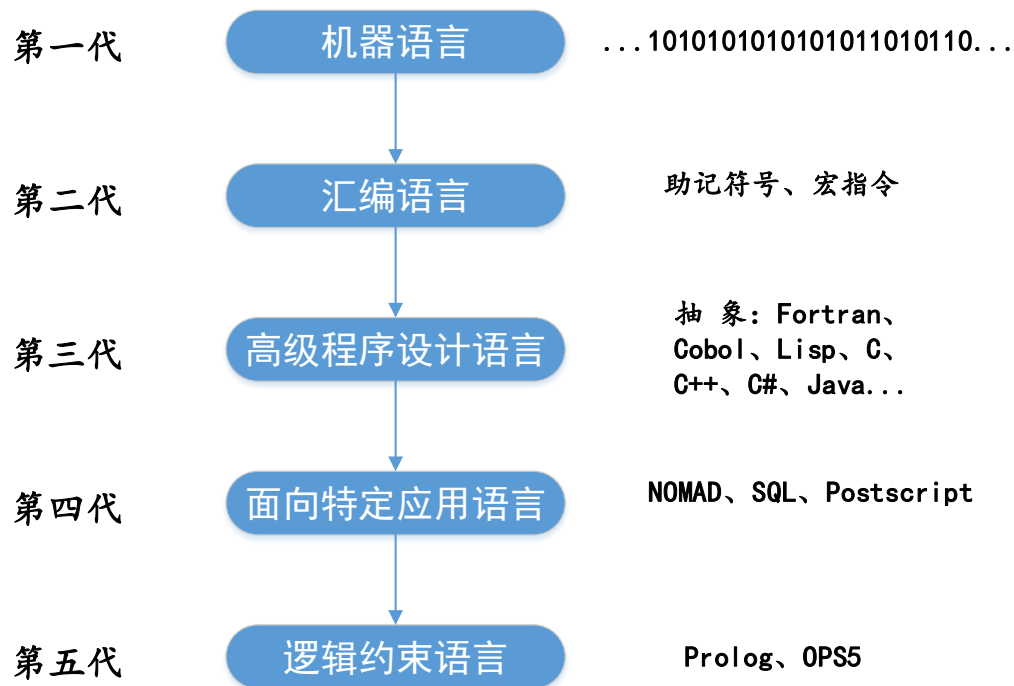
布尔代数：电子电路设计中基础性作用



乔治·斯蒂比兹（1937年

11月）：用继电器表示二进制的装置：第一台**二进制电子计算机**

1.1 程序设计语言



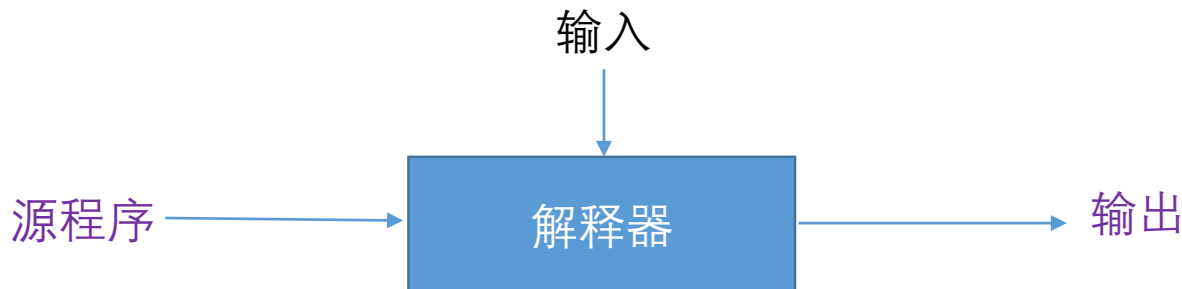
- 命令式 (Imperative) 语言：指明如何完成一个计算任务
如：C、C++、C#、Java
- 声明式 (Declarative) 语言：指明进行哪些计算，通过函数、推论规则或项重写规则，来描述变量之间的关系
如：ML、Haskell、Prolog

1.2 编译的概念

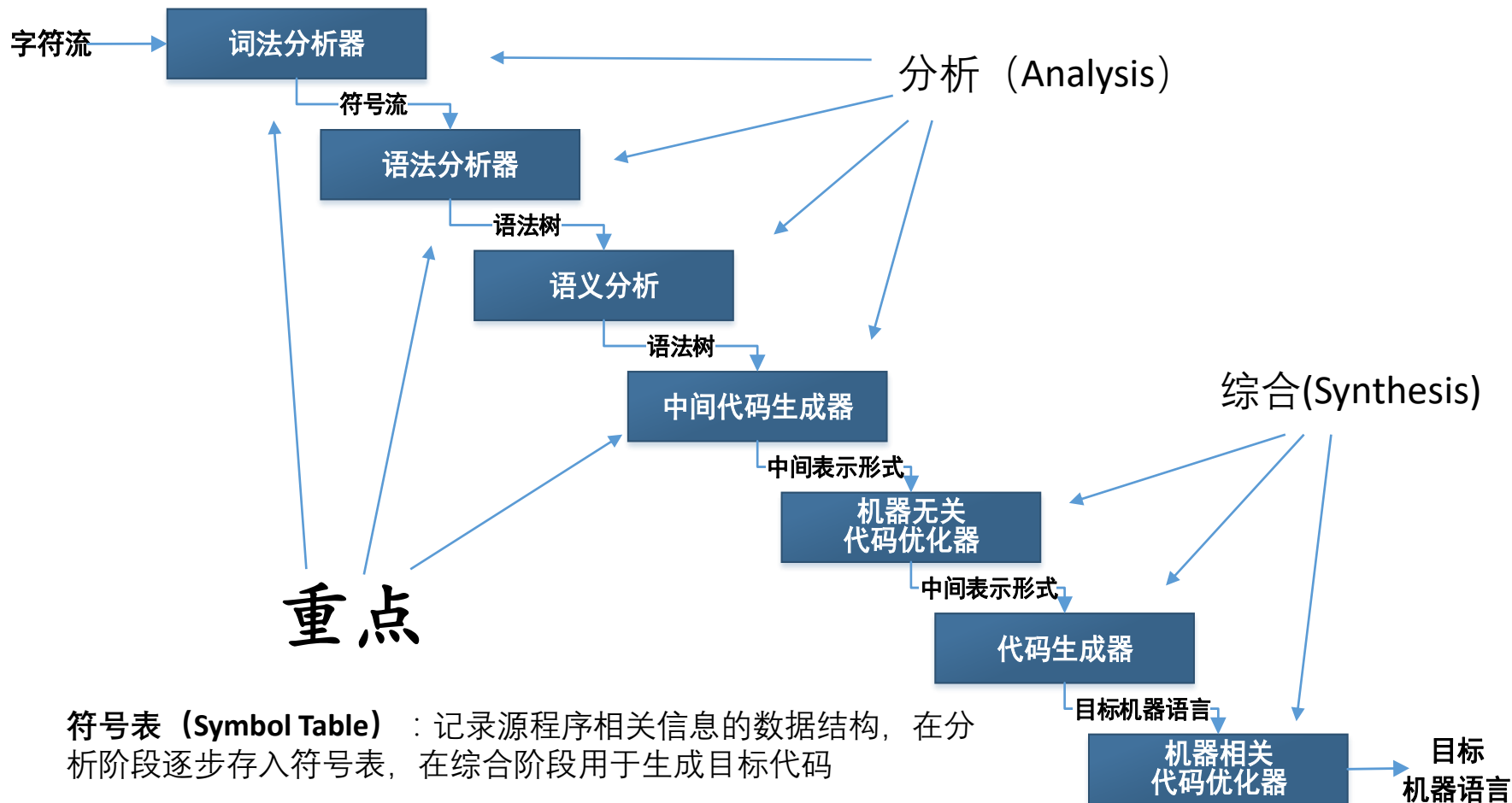
- 编译：读取以一种语言编写的程序，并把该程序翻译成为一个等价的、用另一种语言编写的程序



- 解释：直接利用用户提供的输入执行源程序中指定的操作（如：Java虚拟机对字节码的解释执行）



1.3 编译的过程



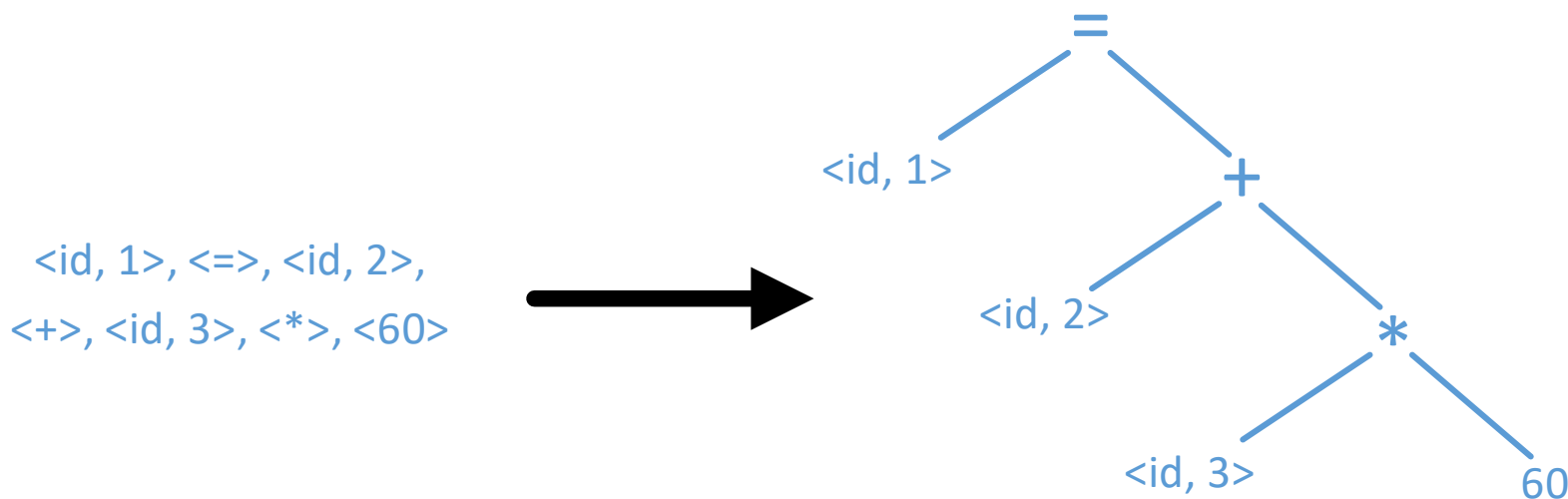
第二节 编译程序概述

2.1 词法分析

- 词法分析(Lexical Analysis)或扫描(Scanning)
 - 读取组成源程序的字符流, 组成词素(Lexeme)
 - 对于每个词素, 产生词法单元(Token, 有的书上称其为内码形式): $\langle \text{Token-name}, \text{Attribute-value} \rangle$
 - Token-name是由语法分析步骤使用的抽象符号
 - Attribute-value指向符号表中关于该词法单元的条目
- 例子:
 - 赋值语句: $z = x + y * 60$
 - 词法单元:
 $\langle \text{id}, 1 \rangle, \langle = \rangle, \langle \text{id}, 2 \rangle, \langle + \rangle, \langle \text{id}, 3 \rangle, \langle * \rangle, \langle 60 \rangle$

2.2 语法分析

- 语法分析(Syntax Analysis)或解析(Parsing)
 - 使用词法分析生成的词法单元第一分量创建树形中间表示，常用语法树(Syntax Tree)表示

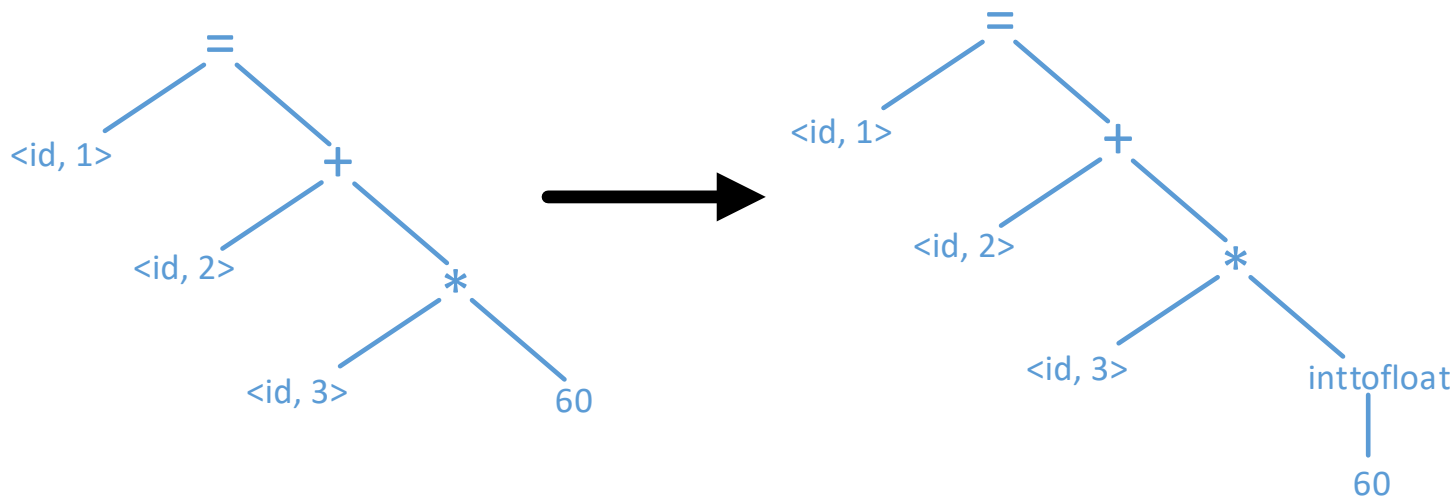


巴科斯范式(BNF: Backus-Naur Form)是由 John Backus 和 Peter Naur 首次引入的一种形式化符号来描述给定语言的语法 (ALGOL 60)

2.3 语义分析

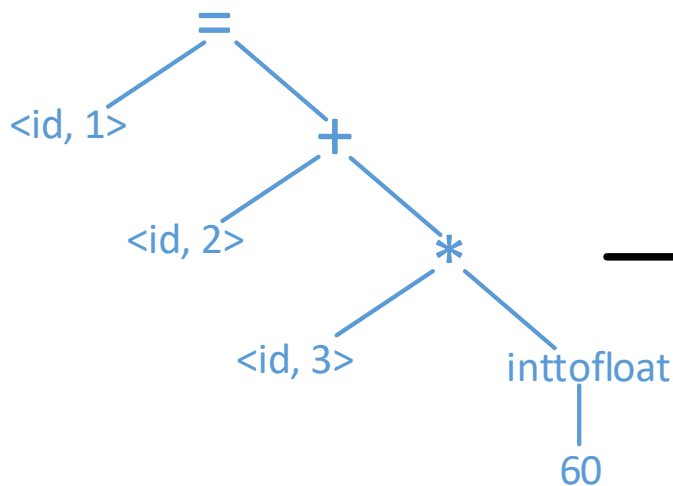
- 语义分析(Semantics Analysis)

- 基于语法树和符号表信息，检查源程序是否满足语义约束
- 收集类型信息，用于后续代码生成过程
- 主要完成类型检查、自动类型转换等功能



2.4 中间代码生成

- 在语法和语义分析之后，生成类机器语言的中间表示形式
 - 通常使用三地址代码 (Three-address Code, 也称四元式)
 - 形式上类似汇编语言指令，每条指令最多包含三个运算分量



→

```
t1 = inttofloat(60)
t2 = id3 * t1
t3 = id2 + t2
id1 = t3
```

2.5 代码优化

- 改进中间代码：满足执行速度更快、代码量更少、能耗更低等要求
- 机器无关/机器相关的优化
- 删除公共子表达式、合并已知量、删除无用赋值、循环优化等

```
t1 = inttofloat(60)
t2 = id3 * t1
t3 = id2 + t2
id1 = t3
```



```
t1 = id3 * 60.0
id1 = id2 + t1
```

2.6 代码生成

- 将中间代码转换为目标语言
 - 寄存器分配
 - 指令选择
 - 内存分配

t1 = id3 * 60.0
id1 = id2 + t1



LDF R2, id3
MULF R2, R2, #60.0
LDF R1, id2
ADDF R1, R1, R2
STF id1, R1

F表示处理的是浮点数

2.7 表格与表格管理

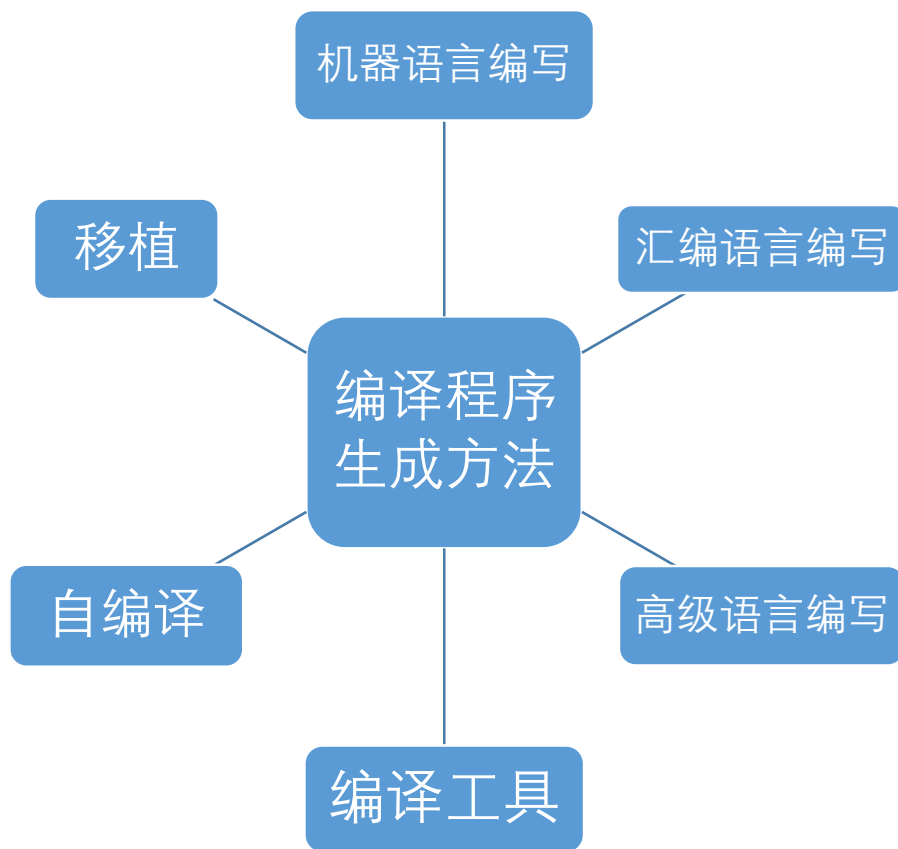
- 表格用于记录源程序各种信息，与分析阶段有关的表格：
 - 符号表：涉及常量名、变量名、数组名、过程名等
 - 常数表：记录源程序中出现的字面常量值
 - 标号表：记录源程序中出现的标号定义和引用情况
 - 分程序入口表：登记过程的层号、分程序符号表入口

2.8 遍

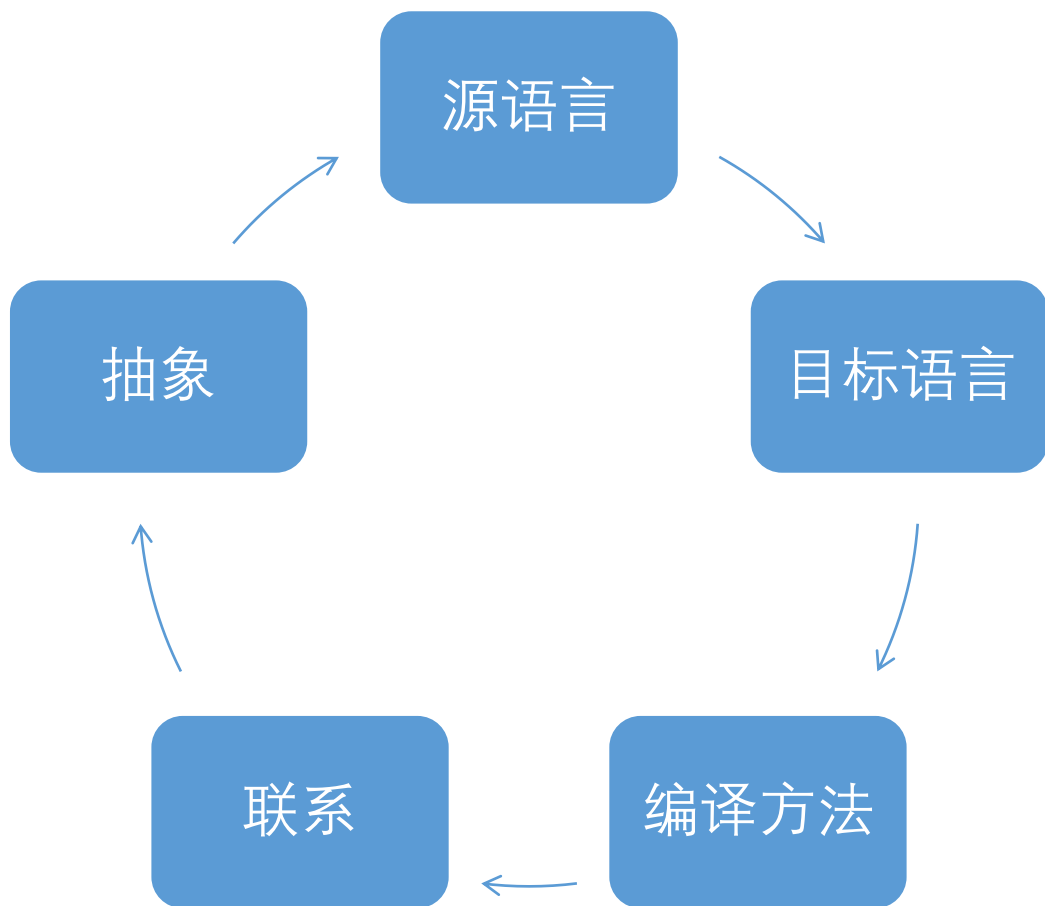
- 遍：对源程序或源程序中间结果从头到尾扫描一次，并做加工处理，生成新的中间结果或者目标代码的过程
- 多遍扫描可以节省内存空间、提高目标代码质量
- 多遍扫描的缺点是耗时长

第三节 编译程序的构造

3.1 编译程序的生成方法



3.2 编译程序的设计方法



小结

- 编译的基本概念
- 编译的基本过程
- 编译器构造方法