**编译原理教案**

**(适用计算机科学与技术专业)**

**王翠荣**

**东北大学秦皇岛分校计算机与通信工程学院**

**编译原理教案(总页)**

授课时间： 2019 年至 2020 年，第 1 学期，第 1 周至 12 周

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 编译原理 | 课程代码 |  | | | 总学时： 56  讲课学时： 48  实验学时： 8 | |
| 学 分 | 3.5 | | |
| 课程性质 | 必修课（√ ） 选修课（ ）  理论课（ ） 实验课（ ） | | | | |
| 任课教师 | 王翠荣 吕艳霞 | 职称 | | 教授 | |
| 授课对象 | 专业： 计算机科学与技术 　　年级： 大学三年级 　班级： 2017级6个班 | | | | | | |
| 教材和  主要参  考资料 | 教材: 张素琴,吕映芝. 编译原理(第2版). 北京:清华大学出版社, 2010.  主要参考教材:  [1] Alfred V. Aho：《现代操作系统》（第2版），机械工业出版社，2009. | | | | | | |
| 教学目  的和教  学要求 | 教学目的是让学生在了解一门高级语言的基础上，在理论上掌握形式语言的定义和词法分析、语法分析、语义分析和语法制导的中间代码生成算法，并能进行中间代码优化、目标代码生成算法和寄存器分配算法。实验课要完整实现PL0语言编译器全过程。课后实践方面要求学生学会C语言版本的Lex和YACC的使用，根据课上理论内容解析开源软件的源代码，然后自己动手用JAVA或者Python语言编程实现实现一个Lex或者YACC。 | | | | | | |
| 教学重  点和教  学难点 | 教学重点：  （1）文法分类，规则和语言的定义，句型、句子的推导，短语、句柄、素短语求法，左递归消除；（2）单词的三型文法的定义，自动机与三型文法的转换，自动机与正则表达式的转换，NFA转换成DFA的子集法，Lex的自动实现；（3）语法成分的上下文无关文法定义，LL(1)自顶向下的语法分析方法，算符优先分析方法，自底向上的LR(0)、SLR(1)、LR(1)、LAlR(1)语法分析方法；（4）中间代码形式，属性文法定义，语法制导的中间代码生成，高级语言的常用语法成分的目标代码结构；（5）程序的运行时内存存储分配；（6）中间代码优化，基本块和程序流图，基本块内的优化，循环体的优化；（7）目标代码生成、寄存器分配着色算法；（8）Yacc实现。  教学难点：  NFA到DFA转换，LL(1)自顶向下的语法分析方法，自底向上的LR(0)、LR(1)、LAlR(1)语法分析方法,中间代码优化,寄存器分配的着色算法。 | | | | | | |
| 教学进程 | | | | | | | |
| 授课周次  第1周  第2周  第3～4周  第5周  第6周  第7周  第8～9周  第10周  第11周  第12周 | 授课章节：  第1章 引论  第2章 PL/0编译程序的实现  第3章 文法和语言  第4章 词法分析  第5章 自顶向下语法分析方法  第6章 自底向上优先分析方法  第7章 自底向上的LR分析  第8章 语法制导翻译和中间代码生成  第9章 目标程序运行时的存储组织  第10章 代码优化  第11章 目标代码生成 | | | | 学时  2  2  6  8  4  4  8  6  2  4  2 | | 备 注  单周4学时，双周2学时  讲课48学时  实验8学时 |

第1章 时间：第 1 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第1章 引论** | | |
| 教学目  的和教  学要求 | 计划授课学时：本章需要2个学时  通过本章学习，掌握编译原理基本内容和特点；现有主要编译技术和未来发展趋势；掌握编译主要过程：词法分析、语法分析、中间代码生成、代码优化、目标代码生成和寄存器分配算法。 | | |
| 教学  重点  难点 | 编辑程序的结构，解释程序和编译程序的基本原理。 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 1.1 什么是编译程序  1.2 编辑程序的结构  1.3 编译程序的软件工具  1.4 现代编译器发展趋势 | 1学时  1学时 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | **课后习题1，2，3，4** | | |
| 备注 |  | | |

第 1 讲 时间：第 1 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第1章 引论**  **1.1 什么是编译程序** | | |
| 教学目的和学要求 | 了解本课程的性质、任务及应用领域，掌握编译程序过程，了解高级语言、汇编语言和机器语言的关系 | | |
| 教学难点 | 编译程序过程，解释程序和编译程序区别 | | |
| 教  学  进  程 | 教学内容与进程 | 课时分配 | 教学方法与手段 |
| 1.1什么是编译程序  编译程序是现代计算机系统的基本组成部分，是系统软件，是高级语言处理系统；  编译程序是一个语言翻译程序，它把一种语言(称作源语言)书写的程序翻译成另一种语言(称作目标语言)的等价的程序。  1.2 编译过程和编译程序的结构  编译过程包括词法分析、语法分析、语义分析、中间代码生成、中间代码优化、目标代码生成等阶段。  词法分析：从左至右扫描字符流的源程序、分解构成源程序的字符串，识别出(拼)一个个的单词（符号）。单词符号是语言中具有独立意义的最基本结构。  语法分析:层次分析.**依据**源程序的**语法规则**把源程序的单词序列组成语法短语(表示成语法树)。  **语**义分析：进一步分析语法结构正确的程序是否符合源程序的上下文约束、运算相容性等规定。  中间代码生成：在语法制导下根据语义规则将源程序翻译成四元式。  代码优化：应用一些技术对代码进行变换以使得编译产生的目标代码高效。  目标代码生成：将优化后的中间代码生成目标机汇编或者机器指令。  1.3 编译程序的软件工具  语言的结构化编辑器、语言程序的调试工具、程序格式化工具、语言程序测试工具、程序理解工具、高级语言之间的转换工具。  1.4现代编译器发展趋势  高性能计算、嵌入式计算、分布式计算环境下的编译技术。 | 10分钟  40分钟  30分钟  20分钟 | 理论和实践并重的课程  理论部分的题目出现于书练习，课堂小测和例题  实践题目1：  用高级语言（C或Pascal）实现扩充的PL/0编译程序  期末考试各部分权重：  作业 10%  实验 10%  期末考试 80% |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 2 讲 时间：第 1 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第2章** PL/0编译程序的实现 | | |
| 教学目的和教学要求 | 了解PL/0语言的BNF定义、编译器的词法分析、语法分析、语义分析、目标代码生成 | | |
| 教学难点 | PL/0语言的BNF定义 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 2.1 PL/0语言描述  PL/0语言是PASCAL语言的子集。PL/0程序示例、PL/0的语法描述图、 PL/0语言的EBNF表示。   * 1. PL/0编译程序的结构   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   * 1. PL/0编译器的词法分析   词法分析过程GETSYM完成从源程序读字符、滤空格、识别保留字、识别标识符、拼数、识别单字符单词、拼双字符单词。   * 1. PL/0编译器的语法和语义分析   递归子程序法   * 1. PL/0编译器的目标代码结构和代码生成   目标代码类p-code是一种栈式机的汇编语言。指令格式：f l a 。  2.6 PL/0编译器的语法错误  有TEST测试程序完成。  2.7 PL/0语言运行时存储分配  栈式动态存储分配内存空间。 | 10分钟  5分钟  5分钟  15分钟  5分钟  5分钟  5分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | **阅读附录A.2** | | |
| 备注 |  | | |

第 3 讲 时间：第 2 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第3章** 文法和语言  3.1文法和语言的形式定义 3.2 文法的类型 | | |
| 教学目的和教学要求 | 掌握文法和推导的概念、句型、句子的定义，了解文法分类，学会生成句型的语法树 | | |
| 教学难点 | 2型文法、3型文法、句型的推导和语法树生成 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第三章 文法和语言  **3.1 文法和语言的形式定义**  规则： 巴克斯范式BNFα→β  元符号::=, **→,|, <>,{}**  文法：四元式**文法G=（VN，VT，P，S）**  推导：α→β是文法G的产生式，若有v,w满足：v=γαδ,w= γβδ, 其中γ∈V\*,δ∈V\*  则称v直接*推导*到w,记作 v ⇒ w  句型：有文法G[S]，若S =>\* x，则称x是文法G的句型。  句子:有文法G[S]，若S =>\* x，且x∈VT\*，则称x是文法G的句子。  语言：文法中全部句子的集合成文语言。  文法等价：如果两个文法所表示的句子集合相等。  **3.3 文法的类型**  0型文法（短语结构文法）的能力相当于图灵机，可以表征任何递归可枚举集，而且任何0型语言都是递归可枚举的  1型文法（上下文有关文法）：产生式的形式为α1Aα2→α1βα2，即只有A出现在α1和α2的上下文中时，才允许β取代A。识别系统是线性界限自动机。  2型文法（上下文无关文法CFG）：产生式的形式为A→β，β取代A时与A的上下文无关。其识别系统是不确定的下推自动机。 描述语法的构成。  3型文法（正规文法RG）：文法G= (VN,VT,P,S),其中P中的产生式的形式为A→αB或者A→α，其中A和B是非终结符号，α∈VT\* 。 产生的语言是有穷自动机（FA）所接受的集合。 | 50 分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 课后习题：1-7 | | |
| 备注 |  | | |

第 4 讲 时间：第 2 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第3章** 文法和语言  3.3 3.4 | | |
| 教学目的和教学要求 | 掌握文法和推导的概念、句型、句子的定义，了解文法分类，学会生成句型的语法树 | | |
| 教学难点 | 2型文法、3型文法、句型的推导和语法树生成 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第三章 文法和语言  3.3 上下文无关文法和语法树  语法树定义。  给定文法G=(VN,VT,P,S)，对于上下文无关文法G的任何句型都能构造与之关联的语法树(推导树)。  文法二义性：若一个文法存在某个句子对应两棵不同的语法树，则称这个文法是*二义*的或者，若一个文法存在某个句子有两个不同的最左（右）推导，则称这个文法是*二义*的。  3.4 上下文无关文法的句型分析  *句型分析*就是识别一个符号串是否为某文法的句型，是某个推导的构造过程。  在语言的编译实现中，把完成句型分析的程序称为*分析程序*或*识别程序*。分析算法又称*识别算法*。  *从左到右的分析算法*，即总是从左到右地识别输入符号串，首先识别符号串中的最左符号，进而依次识别右边的一个符号，直到分析结束。  3.4.1自上而下的分析方法  自上而下分析法：从文法的开始符号出发，反复使用文法的产生式，寻找与输入符号串匹配的推导，或者说，为输入串寻找一个最左推导。  自上而下的语法分析存在的问题：左递归和回溯。  判断和消除左递归和回溯方法。  3.4.2自下而上的分析方法  自下而上分析法：从输入符号串开始，逐步进行归约，直至归约到文法的开始符号。  自下而上的分析方法中如何识别可归约的串：在分析程序工作的每一步，都是从当前串中选择一个子串，将它归约到某个非终结符号，该子串称为“可归约串”。如句柄, 素短语 | 50分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | **8-11,13** | | |
| 备注 |  | | |

第 5 讲 时间：第 3 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第3章** 文法和语言  3.5 有关文法实用的一些限制 习题课 | | |
| 教学目的和教学要求 | 文法化简，消除左递归和回溯，求句柄和素短语 | | |
| 教学难点 | 消除左递归，求句柄和素短语 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第三章 文法和语言  3.5 有关文法实用的一些限制  文法化简，消除左递归和回溯  3.8典型例题和解答  讲解例1  习题课  课后作业1-11,13讲解 | 50分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | **2015年期中考试题有关部分习题** | | |
| 备注 |  | | |

第 6 讲 时间：第 3 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第4章** 词法分析  4.1词法分析(lexical analysis)程序设计4.2 单词的的描述工具 | | |
| 教学目的和教学要求 | 掌握单词的描述工具正则表达式和三型文法，NFA与正则文法和正则表达式关系，NFA转化为DFA | | |
| 教学难点 | DFA的实现 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 4.1词法分析(lexical analysis)程序设计  逐个读入源程序字符并按照构词规则切分成一系列单词。 单词是语言中具有独立意义的最小单位，包括保留关键字、标识符、常量、运算符、标点符号、分界符等。  词法分析是编译过程中的一个阶段，在语法分析前进行 。也可和语法分析结合在一起作为一遍，由语法分析程序调用词法分析程序来获得当前单词供语法分析使用。  4.1.1词法分析程序和语法分析程序的关系  词法分析程序作为语法分析和语义分析的子程序  4.1.2词法分析程序的主要任务及输出  读源程序，产生用二元组表示的单词符号(单词种类，单词自身的值)。单词种类是语法分析需要的信息，单词自身的值是语义分析和代码生成阶段需要的信息。  滤掉空格，跳过注释、换行符、记录源程序的行号，以便出错处理程序准确定位源程序的错误。  4.2 单词的的描述工具  4.2.1 正规文法  三型文法用于描述单词的文法。  4.2.2 正规式  正规式也称正则表达式,是描述单词的构成语法的有效工具，是定义正规集的数学工具。  4.2.3 正规文法和正规式的等价性  一个正规语言可以由正规文法定义，也可以由正规式定义，对于一个任一个正规式，存在一个生成同一个语言的正规文法；反之亦然。两者可以互相转换。 | 50分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 阅读附录A.2：PL0语言的词法分析程序 | | |
| 备注 |  | | |

第 7 讲 时间：第 4 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第4章** 词法分析  4.3 有穷自动机 | | |
| 教学目的和教学要求 | 掌握单词的描述工具正则表达式和三型文法，NFA与正则文法和正则表达式关系，NFA转化为DFA | | |
| 教学难点 | DFA的实现 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 4.3 有穷自动机  确定的有穷自动机DFA，不确定的有穷自动机NFA  NFA转换为DFA，DFA的最小化。  4.3.1 确定有穷自动机  DFA定义： 一个确定的有穷自动机（DFA）M是一个五元组：M=（K，Σ，f，S，Z）其中K是一个有穷状态的集，它的每个元素称为一个状态；Σ是一个有穷字母表，它的每个元素称为一个输入符号；f是转换函数；S∈K是唯一的一个初态；Z⊂ K是一个终态集。  4.3.2 不确定有穷自动机  NFA定义： M=⎨K，∑，f，S，Z⎬，其中K为状态的有穷非空集， ∑ 为有穷输入字母表，f为K× ∑\* 到K的子集（2 K）的一种映射，S⊆K是初始状态集，Z ⊆K为终止状态集.  4.3.3 不确定自动机与确定自动机的关系  DFA是NFA的特例。对每个NFA N一定存在一个DFA　Ｍ ，使得 L(M)=L(N)。  将NFA转换成接受同样语言的DFA.这种算法称为子集法。 与某一NFA等价的DFA不唯一。  4.3.4 确定有穷自动机的化简  求DFA M =（K,∑,f, k0,, kt)的最小DFA M’  (1)构造状态的一初始划分∏：终态kt 和非终态K- kt两组(group)；(2)对∏施用过程PP 构造新划分∏new ；(3)如∏new  =∏,则令 ∏final=∏ 并继续步骤 4，否则∏:=∏new重复2；(4)为∏final中的每一组选一代表，这些代表构成M’的状态。若k是一代表且f(k,a)=t,令r是t组的代表，则M’中有一转换f’(k,a)=r | 50分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 课本P72作业1，3,5 | | |
| 备注 |  | | |

第 8 讲 时间：第 4 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第4章** 词法分析  4.4 正规式与确定有穷自动机的关系4.5正规文法与确定有穷自动机的关系 | | |
| 教学目的和教学要求 | 掌握单词的描述工具正则表达式和三型文法，NFA与正则文法和正则表达式关系，NFA转化为DFA | | |
| 教学难点 | DFA的实现 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 4.4 正规式与确定有穷自动机的关系  词法分析程序的自动构造基于**有穷自动机和正规表达式的等价性。**  (1) 对于∑上的一个NFA M，可以构造一个∑上的正规式R,使得L(R)=L(M)。  (2)对于∑上的一个正规式R，可以构造一个∑上的NFA M，使的L(M)=L(R)。  4.5正规文法与确定有穷自动机的关系  正规文法与正则表达式是等价的。  从正规文法G直接构造有穷自动机NFA M，满足L(G)=L(M)。  (1) M的字母表与G的终结符相同；  (2) 对G中的每个非终结符生成M的一个状态，G的开始符号S是M的开始状态S；  (3)增加一个新状态作为Ｍ的终态Z；  (4)对Ｇ中形如A→tB规则，构造Ｍ的ｆ(A,t)=B函数;  (5)对Ｇ中形如A→t规则，构造Ｍ的ｆ(A,t)=Z 函数。 | 50分钟  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 课本P72作业6-10 | | |
| 备注 |  | | |

第 9 讲 时间：第 5 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第4章** 词法分析  4.6 词法分析程序的自动构造工具； 习题课；优秀学生演示自己开发的LEX程序。 | | |
| 教学目的和教学要求 | 掌握单词的自动构造工具LEX的使用 | | |
| 教学难点 | LEX语法和使用 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 4.6 词法分析程序的自动构造工具  下载C语言版本的Lex，YACC工具，编译，运行，生成C语言的词法分析程序。  单词的结构用正规式描述 正规式⇒ NFA ⇒DFA ⇒min DFA。  用LEX建立词法分析程序的过程：  LEX源程序lex.1；LEX编译器；Lex.yy.c 。  习题课：书后所有习题讲解。  课堂学生演示自己开发的NFA-DFA转换程序并讨论。 | 20  50  30 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 10 讲 时间：第 5 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第4章** 自顶向下语法分析  5.1 自顶向下语法分析的基本思想 5.2 LL（1）文法5.3 递归子程序法 | | |
| 教学目的和教学要求 | 掌握LL文法的判别，学会LL(1)分析方法，完成递归子程序法的程序 | | |
| 教学难点 | LL(1)文法的判别 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| * 1. 自顶向下语法分析的基本思想   从推导的角度看，从开始符号出发，使用最左推导，试图推导出与输入符号串相同的句子。  从语法树的角度看，从根节点出发，反复使用所有可能的产生式，谋求输入串的匹配，试图向下构造一棵语法树，其末端节点正好与输入符号串相同。  5.2 LL（1）文法  一个上下文无关文法若满足下列条件，称为LL(1)文法：  (1)文法不含左递归  (2)文法中每个非终结符A的各个产生式的首终结符集两两不相交，即，若  A→ α1| α 2 |…| αn  则 FIRST（ αi ）∩FIRST（ αj ）=Φ  (3)文法中某个非终结符A,若有αi⇒\*ε，则 FIRST( αj )∩FOLLOW（A）= Φ  5.3 递归子程序法  对LL(1)文法才能使用递归下降法。 为每一个非终结符编制一个递归下降过程，过程的名字就是产生式左部的非终结符，过程体则是按产生式的右部符号顺序编写的，每匹配一个终结符，则再读入下一个输入符号；对于产生式右部的每个非终结符，则递归调用相应过程。  只有LL（1）文法，才可以实现确定的(不带回溯)自顶向下语法分析。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 11 讲 时间：第 6 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | **第5章** 自顶向下语法分析  5.4预测分析方法5.5 典型例题 | | |
| 教学目的和教学要求 | 掌握LL文法的判别，学会LL(1)分析表的求法，实现预测分析法的语法分析过程 | | |
| 教学难点 | LL(1)分析表的构造 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 5.4预测分析方法  根据当前要分析句子的输入符号，为当前要处理的非终结符唯一地选择一个产生式，要求文法是LL(1)的。 表驱动的预测分析器包含数据结构： 一个输入缓冲区，一个栈，一张分析表，一个输出流。  预测分析表是一个M[A，a]形式的矩阵，其中： A为非终结符，a为终结符或# 。  M[A，a]中存放着一条关于A的产生式，指出当A面临a时所应采取的候选。  M[A，a]中也可能存放一条“出错标志”，指出Ａ不应该面临a。  预测分析器的工作方式：  (1)如果X=a≠#，则POP，advance;  (2)如果X ∈Vn,查M[X，a]表  若M[X，a]=X→UVW,则用WVU替换栈顶；  若M[X，a]=error,则调用错误恢复程序  (3)如果X=a=#,分析成功**.**  5.5典型例题  例1：对于文法G  1 E→TE’  2 E’ → +TE’|**ε**  **3 T** →FT’  4 T’→\*FT’| **ε**  5 F→(E)|**id**  例2：文法G（S）  S→S\*aT|aT  T →+aT|ε  消去左递归，求FIRST和FOLLOW  写出句子a\*a\*a+a的分析过程 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 12 讲 时间：第 6 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第6章 自底向上优先分析方法  6.1 自底向上优先分析基本思想6.2 简单优先分析方法 | | |
| 教学目的和教学要求 | 规范归约，自上而下的算符优先分析方法及其相关概念，掌握自下而上分析的基本思想，规范规约的概念及过程 | | |
| 教学难点 | 优先文法、最左素短语、句柄的定义与判定;构造简单优先关系表 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 6.1 自底向上优先分析基本思想  设置一个栈，将输入符号逐个**移进**栈中，栈顶形成**某**产生式的右部时(句柄)，就用左部去代替，称为**归约**。重复这一过程，直到栈中只剩下文法的开始符号，就确认输入串是文法的句子，分析成功，否则出错。   * 1. 简单优先分析方法   简单优先分析法的思想是利用终结符和非终结符之间的优先关系确定句柄，是规范规约。  6.2.1 优先关系  对于文法G中的任意两个符号X和Y按其在句型中可能会出现的相邻关系来确定他们之间的优先关系。X和Y可以是非终结符也可以是终结符。  (1) X优先性等于Y ，记作X =Y。当且仅当G中存在规则A→…XY...  (2) X优先性低于Y， 记作X<Y。当且仅当G中存在规则A→…XB… , 且B→+Y…  (3) X优先性高于Y ，记作X>Y。当且仅当G中存在规则A→…BD… ，且B→+…X，D→\*Y…  6.2.2 简单优先文法  (1)在文法符号集V中的任意两个符号最多只有一种优先关系；  (2)文法中任意两个产生式没有相同的右侧。  6.2.3 简单优先语法分析方法  (1)将输入符号串a1a2…an#存入符号栈T中，直到遇到栈顶符号ai的优先级 >下一个待输入符号的优先级为止；  (2)栈顶当前符号ai为句柄尾由此向左找句柄的头符号ak，即找到ak-1 <ak为止；  (3)由句柄ak…ai在文法的产生式中找右部为ak…ai的产生式，若找到用左部代替右部，否则不是句子； | 20  30  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 13 讲 时间：第 7 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 6.3 算符优先分析法 | | |
| 教学目的和教学要求 | 了解算符优先文法的定义和构造，掌握算符优先分析算法 | | |
| 教学难点 | FirstVT、LastVT集求法，算符优先分析方法 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| * 1. 算符优先分析法   算符优先分析法的思想源于表达式的分析，是利用相邻终结符号之间的关系来寻找可归约串。  将句型中的终结符号当作“算符”，借助于算符之间的优先关系确定规约子串，分析过程是自下而上的归约过程，不是一种严格的规范归约。   * + 1. 算符优先文法的定义   在一个符号串中，任意两个相邻终结符号a和b之间，只可能存在三种优先关系：  (1) a优先性等于b ，记作a=b  (2) a优先性高于b， 记作a>b  (3) a优先性低于b ，记作a<b  算符优先关系的定义：文法G是一个不含ε-产生式的算符文法，定义终结符a、b之间的优先关系  ①a= b, G中有P🡪...ab...或P🡪...aQb... 产生式； ②a<b, G中有P🡪...aR...的产生式,且R=>+b...或 R=>+Qb... (注意ab相邻或只隔一个非终结符)； ③a>b, G中有P🡪...Rb...的产生式,且R=>+...a或R=>+...aQ (注意ab相邻或只隔一个非终结符)。   * + 1. 算符优先表的构造  1. 照定义手工计算   ②使用FirstVT和LastVT集算法  FirstVT集：对每个非终结符P, FirstVT(P)= {a|P=>a...或P=>Qa...,a为终结符,P，Q为非终结符}  LastVT集 ：LastVT(P)={a|P => ...a或  P =>...aQ,a为终结符,P，Q为非终结符}   * + 1. 算符优先分析算法   通过比较相邻终结符间的优先关系来实现，仍然采用“移进-归约”方式，不断移进输入符号，识别可归约串，并进行归约。据优先性“低于”来识别可规约前缀的头，根据优先性“高于”来识别可规约前缀的尾。各种优先关系已经存于优先关系表中。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 2,3,4,7 | | |
| 备注 |  | | |

第 14 讲 时间：第 7 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第7章 自底向上的LR分析  7.1 LR分析的基本思想 7.2 LR(0)分析 | | |
| 教学目的和教学要求 | 了解LR分析的基本思想，理解可归前缀，掌握识别活前缀的自动机 | | |
| 教学难点 | 识别活前缀的自动机、LR(0)项目集规范族的构造 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 7.1 LR分析的基本思想  分析器的逻辑结构及工作过程 LR(k)分析技术  L ----是指从左至右扫描输入符号串  R----是指构造一个最右推导的逆过程  K----是指为了作出分析决定而向前看的输入符号的个数。  根据栈中的符号串和向右顺序查看输入串中的k(k≥0)个符号，就能唯一确定分析器的动作是移进还是归约，以及用哪个产生式进行归约。  7.2 LR(0)分析  把每个句柄的识别过程划分为若干个状态, 每个状态下，自左向右地识别了句柄的一部分符号。  7.2.1 可归前缀和子前缀  活前缀定义: 规范推导 S⇒\*αAω⇒αβω，αβ的所有前缀称为活前缀。即规范句型的前缀，但右端不超过该句型句柄末端。  识别了句柄的一部分就相当于识别了当前规范句型的左起部分，这部分被识别的符号串称之为规范句型的活前缀。  7.2.2 LR(0)项目集规范族的构造  设I是文法G的一个LR(0)项目集合, closure(I)是从I出发用下面三个规则构造的项目集:(1) I中每一个项目都属于closure(I)；2)若项目A→α·Bβ ∈closure(I)且B→η∈P ， 则将B → ·η加进closure(I)中。 (3)重复执行(2)直到closure(I)不再增大为止。  7.2.3 识别活前缀的有穷自动机  由于S’仅在第一个产生式的左部出现，规定项目1为初态，其余每个状态都为活前缀的状态.圆点在最后的项目为句柄识别态，第一个产生式的句柄识别态为句子识别态，用\*标记。  分析算法程序见ppt。 | 20  30  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 1,2,3，7 | | |
| 备注 |  | | |

第 15 讲 时间：第 8 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 7.3 SLR(1)分析方法 | | |
| 教学目的和教学要求 | 掌握SLR(1)分析的方法 | | |
| 教学难点 | SLR(1)分析表 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 7.3 SLR(1)分析方法  主要问题也是分析表的构造.有的文法，识别活前缀的DFA状态集中，含有冲突项目。这种解决方法是比较简单的，因此称作SLR(1)分析，由此构造的分析表，称作SLR分析表。  面对当前读入符号为x  (1) 若x ∈{a1 ,an, …am},则移进。  (2) 若x∈ FOLLOW(Bi),则用Bi → **γi**  进行归约。  (3) 此外，报错。  这种解决方法是比较简单的，因此称作SLR(1)分析，由此构造的分析表，称作SLR分析表。  ACTION表和GOTO表构造如下：  (1)若项目 A→ α ⋅X β ∈Ik,且GO(Ik，X)= Ij,①X=a ∈ VT,则置ACTION[k,a]= Sj,意为把(j,a)移进栈; ② X∈ VN,则置GOTO(k,X)=j；  (2)若项目 A→ α ⋅∈Ik, 对于a ∈FOLLOW(A),则ACTION[k,a]= rj,意为用第j个产生式A→ α 进行归约；  (3)若 S’→ S ⋅∈Ik, 则置ACTION[k, #]= acc；  (4)凡不能用以上规则填入的入口 ,置“ERR”。  给出分析某个终结符号串是否为该文法的句子的算法，见ppt讲稿。  讲里两个例子，学生课堂练习一题。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 5,6,9 | | |
| 备注 |  | | |

第 16讲 时间：第 8 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 7.4 LR(1)分析方法 7.5 语法分析程序的自动构造工具YACC | | |
| 教学目的和教学要求 | 掌握LR(1)、LALR(1)分析的方法 | | |
| 教学难点 | LR(1)项目集和分析表构造 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 7.4 LR(1)分析方法  虽然SLR(1)方法能够解决部分非LR(0)文法的语法分析，但是仍然存在其它的一些SLR(1)解决不了的文法。  解决方法：通过分裂状态，使LR分析器的每个状态能确切知道**句柄α后紧跟哪些终结符时才能把α归约成A**。LR(k)分析方法。  LR(k) 项目：  形式：[A→α**⋅**β , a1a2…ak]  β≠ε,移进或待归约项目， a1a2…ak不起作用。a1a2…ak称向前搜索符号串。  7.4.1 LR(1)项目集的构造  当k=1时，此时的项目[A→α⋅β , a]称为LR(1)项目。搜索符a只对归约项目[A→α⋅ , a]起作用，意为只有当[A→α⋅ , a]所属项目集位于栈顶时，且当前输入符号为a时，才能用A→α进行归约。  这样一些a所构成的集合定是FOLLOW(A)的一个子集，可能是真子集。  初始状态I0:=closure( {[ S’→·S,#]})。设I是G的一个LR(1)项目集，closure(I)是从I出发，用下面三个规则构造的项目集 ：  (1)每一个I中的项目都属于closure(I)。  (2)若项目[A→α·Bβ，a]属于closure(I), 且B→η∈ P,则对任何b∈FIRST(βa), 把 [B→·η, b]加进closure(I)中。  7.4.2 LR(1)分析表的构造  若项目[ A→ α ⋅X β ,b]∈Ik,且GO(Ik,X)= Ij,①X=a ∈ VT,则置ACTION[k,a]= Sj,意为把(j,a)移进栈; ② X∈ VN,则置GOTO(k,X)=j若项目 [A→ α ⋅,a]∈Ik,且A≠S’,对于a ,则置ACTION[k,a]= rj,意为用第j个产生式A→ α 进行归约.  7.5 语法分析程序的自动构造工具YACC  Yacc构造语法分析器的构造过程和基本原理。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 10,11 | | |
| 备注 |  | | |

第 17讲 时间：第 9 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 语法分析习题课 | | |
| 教学目的和教学要求 | 自顶向下、算符优先和LR(0)、SLAR(1)、LR(1)、LALR(1) 之间关系  讲解5、6、7三章课后习题 | | |
| 教学难点 | 上下文无关文法不同分析方法之间的关系 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 总结复习5、6、7内容  给出三章内容之间的关系：  讲解习题：  1. 已知文法 G[S] ：S→MH|a H→LSo|ε K→dML|ε L→eHf, M→K|bLM 判断 G 是否是 LL(1) 文法，如果是，构造 LL(1) 分析表。  2. 文法 G[E] 是 LL(1) 文法:  E → [F] E′ E′ → E|ε  F → aF′ F′ → aF′|ε  其中 E,F,E’,F’ 为非终结符。  构造文法 G[E]的 的 LL(1) 分析表。  3.试消除下面文法G[A] 中的左递归和左公因子，并判断改写后的文法是否为LL(1) 文法?  G[A] ： A→aABe|a B→Bb|d  4.已知文法G[S] 为：S→a| ∧ |(T ) T→T,S|S (1) 计算 G[S]的 的 FIRSTVT 和 和 LASTVT 。 (2) 构造 G[S] 的算符优先关系表并说明 G[S] 是否为算符优先文法。 (3)给出(a,a)和(a,(a,a))的算符优先分析过程。  6. 若有定义二进制数的文法如下： S→L·L|L L→LB|B B→0|1 (1) 试为该文法构造 LR 分析表，并说明属哪类 LR 分析表。 (2) 给出输入串 101.110 的分析过程。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 |  | | |
| 备注 |  | | |

第 18讲 时间：第 9 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第8章 语法制导翻译和中间代码生成  8.1 属性文法 8.2 中间代码形式 | | |
| 教学目的和教学要求 | 了解属性文法的定义，掌握中间代码的形式 | | |
| 教学难点 | 属性文法 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 属性的计算以语义规则(semantic rules)的形式赋予文法的产生式；在语法分析推导或归约的每一步骤中，通过语义规则实现对属性的计算，以达到对语义的处理，语法分析同时生成中间代码。  为每一个产生式配上语义规则并且在语法分析过程中执行这些规则。  即当归约(或推导)到某个产生式时，除了按照产生式进行相应的代换之外(语法分析)，还要按照产生式所对应的语义规则执行相应的语义动作，如计算表达式、查和填符号表、产生中间代码(语义分析)。  8.1 属性文法  属性文法也叫语法制导定义(Syntax-directed definitions) ，在文法产生式右部适当的位置上插入语义规则而得到的新文法称为属性文法。  它是在上下文无关文法的基础上，通过每个文法符号和一个属性集合相关联，通过每一个产生式和一个语义规则集合相关联。语义规则用来计算与产生式中出现的符号相关联的属性的值 。  语法分析是主动的，语义分析是从动的，语法分析制导着语义分析。  8.2 中间代码形式  中间代码生成程序的任务是：把经过语法分析和语义分析而获得的源程序翻译为中间代码表示。中间代码生成方法：语法制导翻译。  中间语言有四种形式：   * 逆波兰表示(后缀式) * 三元式(三地址码)和树型表示 * 四元式   常用语句的四元式表示。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 1-3 | | |
| 备注 |  | | |

第 19讲 时间：第 10 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第8章 语法制导翻译和中间代码生成  8.3 简单赋值语句的翻译8.4 布尔表达式的翻译 | | |
| 教学目的和教学要求 | 掌握赋值语句和表达式的四元式翻译 | | |
| 教学难点 | 布尔表达式的翻译 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 8.3 简单赋值语句的翻译  为了实现赋值语句x=y 语法制导的自动翻译成四元式：  (= , y , ,x)  (1)需要定义赋值语句的属性文法;  (2)语法分析的同时执行语义规则，由emit自动生成四元式。  语义过程emit将生成的四元式送到输出文件上,如emit(+ , E1.place，E2.place， E.place);  **过程lookup ( id.name)负责在符号表中查找标示符id的名字;**  语义变量E.Place表示存放E的变量在符号表中的登录项;  语义过程newtemp表示生产一临时变量。  赋值语句的属性文法定义详细见课本。  8.4 布尔表达式的翻译  布尔表达式: 用布尔运算符号(and,or,not)作用到布尔变量或关系表达式上而组成。  布尔表达式的作用：  1. 用作计算逻辑值  2.用作控制流语句如if-then，if-then-else和while-do等之中的条件表达式。  根据布尔表达式的语义进行翻译。对于出现在条件语句 if E then s1 else s2中的布尔表达式E，其作用就是控制对S1和S2的选择。  作为条件的布尔表达式，设计有两个出口：E.true 和 E.false.对于IF语句，E.true 指向S1， E.false指向S2；  对于while语句E.true 指向循环的开始， E.false指向while外 的下一条语句。 | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 1-2 | | |
| 备注 |  | | |

第 20讲 时间：第 10 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第8章 语法制导翻译和中间代码生成  8.5控制语句的翻译8.6 说明语句的翻译 | | |
| 教学目的和教学要求 | 掌握for、if、while等控制语句的四元式翻译 | | |
| 教学难点 | 控制语句的目标代码结构 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 8.5控制语句的翻译  控制语句文法：  （1） S→ if E then S  （2） S→ if E then S else S  （3） S→ while E do S  （4） S→ begin L end  （5） S→ A  （6） L→ L；S  （7） L→ S  8.5.1 if语句的翻译  E.code  Goto E.false  S1.code  Goto s.next  S2.code  s.next  8.5.2 while语句的翻译  8.5.3 for语句的翻译  E(1) 代码  i:= E(1)值  E(2) 代码  t1:= E(2)值  L1 ： if i>t1 goto Lnext  S(1) 代码  i:=i+1  goto L1  Lnext： | 50  50 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 作业见讲稿 | | |
| 备注 |  | | |

第 21讲 时间：第 11 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第九章 目标程序运行时的存储组织  9.1 数据空间管理方法 9.2 栈式存储分配的实现 | | |
| 教学目的和教学要求 | 为编译后的执行代码对内存能动态分配 | | |
| 教学难点 | 栈式存储分配 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第九章 目标程序运行时的存储组织  为了使目标程序运行，编译程序从操作系统得到一块内存存储区，存储区容纳:生成的目标代码空间; 目标代码运行需要的数据空间，包括用户定义的变量和常量，临时工作单元，过程调用所需的联系单元，输入输出缓冲区;用于保存过程活动踪迹的一个控制栈。静态存储分配:FORTRAN；栈式存储分配:PASCAL, C；堆式存储分配: PASCAL, C。采用哪种分配策略是由源语言的语义决定的。  9.1 数据空间管理方法  9.1.1 静态存储分配  在编译时刻为每个数据项目确定出在运行时刻的存储空间中的位置，且这种绑定在运行时刻不再变化。   限制：(1)在编译时刻知道数据目标的大小和它们在内存位置上 约束；(2)不能递归调用过程；(3)不能动态地建立数据结构。  9.1.2 动态存储分配  对于具备递归调用、可变数组、允许用户申请和释放内存空间的高级语言，就需要采用动态存储管理技术管理内存，主要包括两种动态存储分配方式：栈和堆式。  与静态分配不同的是，在每次活动开始时把局部名字和新的存储单元绑定，在活动结束时，活动记录从栈中弹出，局部名字的存储空间随之释放。  9.1.3 栈式存储分配  基于栈存储分配的原理：存储空间被组织成栈，活动记录的推入和弹出，分别对应于活动的开始和结束。每当调用一个子程序时，它所需的数据存储空间就分配在栈顶，当该子程序运行结束就释放这部分内存存储数据空间。  9.2 栈式存储分配的实现  数据空间使用遵从“先申请后释放，后申请先释放”。 | 50  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 作业见ppt讲稿 | | |
| 备注 | **2017级增加GCC编译和反编译课堂演示，课下学生自己安装环境，体验实验过程。** | | |

第 22讲 时间：第 11 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第十章 代码优化  10.1 优化技术简介10.2 局部优化 10.2.3 基本块的有向图表示 | | |
| 教学目的和教学要求 | 解优化技术，掌握局部优化基本方法 | | |
| 教学难点 | 基本块的有向无环图DAG生成算法 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第十章 代码优化  10.1 优化技术简介  有些编译程序需要对中间代码或者目标代码进行优化。优化就是对代码进行等价变换，使得变换后的代码运行结果与变换前运行结果相同，运行速度加快或占用空间较少。  中间代码优化和(或)目标代码优化。  优化的程序范围分：局部优化、循环优化和全局优化。  10.2 局部优化  10.2.1 基本块的划分  是指程序中一顺序执行的语句序列，　　　　　　　　　　　　　　入口就是其中的第一个语句，出口就是其中最后一个语句。对一个基本块，执行时，只能从入口进入，从出口退出。  划分基本块的算法：  1、入口语句  ①程序的第一个语句  ②能由条件语句或无条件转移语句转到的语句  ③紧跟在条件转移语句后面的语句  2、出口语句  ①每一入口语句的上一语句  ②转移语句  ③停语句  3、基本块由某一入口语句到与之最近的出口语句之间的语句序列组成。  10.2.2 基本块的变换  删除公共子表达式；删除无用代码 ；重新命名临时变量；交换语句次序 。  10.2.3 基本块的有向图表示  结点带有下属标记的DAG生成算法见ppt讲稿。 | 50  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 作业见ppt讲稿 | | |
| 备注 |  | | |

第 23讲 时间：第 12 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第十章 代码优化  10.3 控制流分析和循环优化 | | |
| 教学目的和教学要求 | 学会生成基本块间的控制流图算法，掌握循坏优化基本方法 | | |
| 教学难点 | 循坏优化生成算法 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 10.3 控制流分析和循环优化  10.3.1程序的控制流程图(程序流图)  程序流图为一以基本块为结点的有向图；  含有程序中第一条语句的结点叫作首结点，首节点唯一。  程序流图构造方法：  (1)条件或无条件语句从B1转到B2;  (2)B2紧跟在B1后面(且B1最后语句不是无条件转移语句) .  **10.3.2循环的查找**  m DOM n 定义:  在程序流图中,对任意两个结点m和n,如果从流图的首结点出发,到达n的任意通路都要经过m,则称m是n的必经结点,记为m DOM n (∀a, a DOM a)  **回边组成的循环:**  假设有向边a→b是回边，则它组成的循环是由结点b、结点a以及有通路到达a而该通路不经过b的所有结点组成，并且b是该循环的唯一入口结点。  **10.3.3循环优化**  找出程序流图中的循环之后，可以针对循环进行优化。  循环有常用的三种优化技术：代码外提、删除归纳变量、强度消弱。  实行代码外提时，在循环入口处建立一个新结点，称为前置结点，对一个循环，前置节点是唯一的；把循环不变运算的代码外提到前置结点中； 前置节点以循环的入口节点为唯一后继。 | 50  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 书p272 ex.5-7 | | |
| 备注 |  | | |

第 24讲 时间：第 12 周

|  |  |  |  |
| --- | --- | --- | --- |
| 章 节 | 第11章 目标代码生成  11.1 代码生成概述 11.2 一个简单代码生成程序 11.3 寄存器分配 | | |
| 教学目的和教学要求 | 了解目标指令集，了解一个简单代码生成程序，了解寄存器分配算法 | | |
| 教学难点 | 寄存器分配算法 | | |
| 教  学  进  程 | 教学内容与进程 | 时间分配 | 教学方法与手段 |
| 第11章 目标代码生成  将经过语法分析或优化后的中间代码，转换成特定机器的目标代码。  11.1 代码生成概述  代码生成器的输入：中间代码和符号表  目标代码的形式：汇编语言代码；机器语言模块；机器语言代码。  **绝对机器代码：**所有地址均已定位， 可立即执行。  **待装配的机器语言模块：**当需要执行时，由动态链接 (请求分段)装入程序把它们和某些运行程序链接起来，转换成能执行的机器语言代码。  **汇编语言代码：**尚需经过汇编程序汇编，转换成可执行的机器语言代码。使代码生成变得容易。  指令集选择：寻找若干合适的目标机指令以实现给定的中间表示。好的机器指令序列会使生成的目标代码质量更高、速度更快。  指令选择的功能：多数CPU的指令集合具有冗余性， 同一计算可用两个或多个不同的指令序列 完成。指令选择器选择其中之一以产生 最好的代码。  指令选择的基本原则：减小产生目标代码的尺寸减小目标代码的执行时间。  11.2 一个简单代码生成程序  代码生成有多重实现方式，ppt讲稿给出一个具体实现程序。  11.3 寄存器分配。  （1）选择已被变量占用，但变量以后可能不被引用的寄存器,需要释放寄存器；  （2）已分配寄存器，且存放的变量在较远的结点才被引用，需要释放寄存器；  （3）分配空闲的寄存器；  （4）分配专用寄存器。 | 50  50分钟 | 多媒体与板书结合课堂教学，课堂练习、随机提问、习题课与辅导课、课下答疑，也可通过网络进行讨论。 |
| 思考题  与习题 | 阅读附录A.2,PL0语言的目标代码生成器。 | | |
| 备注 |  | | |