

Compiler2023

by 2111408 周钰宸

词法分析，语法分析是重点

先搞清基本概念

试题分布

- 知识点分布

- 概述10%、词法分析35%、
语法分析35%、语义分析及翻译20%

- 题型分布

- 单项选择20%左右（基本概念）、设计20%
左右、问答60%左右（算法应用）

- 难易分布

- 简单40%、中等55%、较难5%

一种复习方式

- 预习作业（及反馈解答）和复习中的选
择题→基本概念（期末考试选择题）

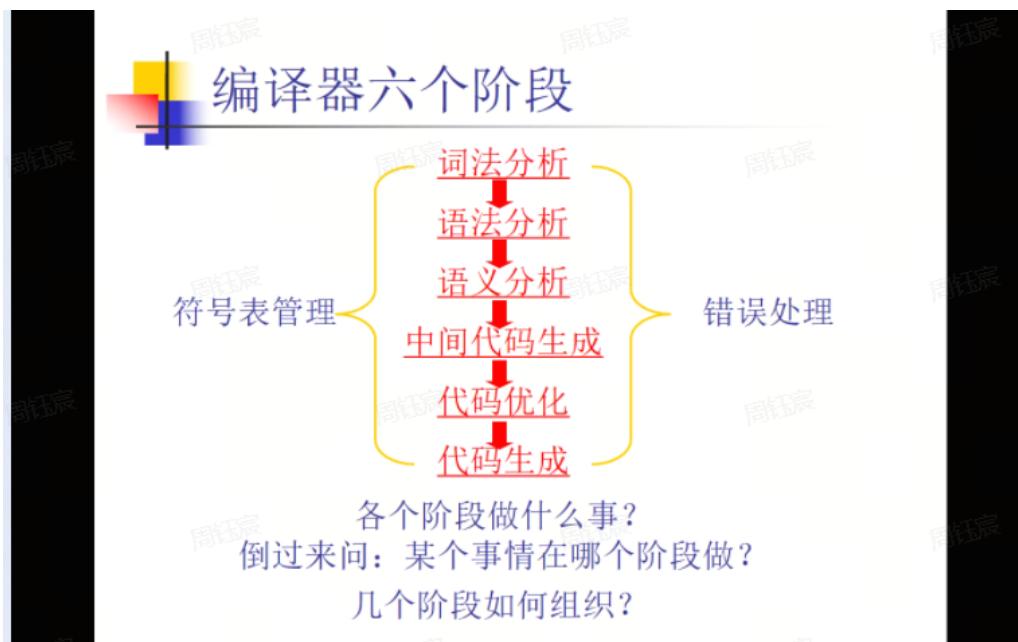
- 如完成不是很好，结合讲义、预习情况分析
及课堂习题强化基本概念的复习

- 平时书面作业→算法应用（期末考试设
计题、问答题）

- 如完成不是很好，结合平时书面作业、教材
课后习题相应强化掌握欠佳的算法

- 然后做一套往年试卷，根据完成情况继
续强化前两步的复习，如此往复

Chapt 1



符号表创立: 词法分析

词法语法分析

(回去听录音)

如何组织: 前端后端, 中端

1.5 编译器各阶段的组合

○ 1.5.1 前端 (front end)

- 主要依赖源语言, 很大程度与目标机器无关
- 词法分析、语法分析、符号表的创建、语义分析、中间代码的生成、部分代码优化、错误处理

后端

○ 后端 (back end)

- 只依赖目标机器、中间代码，不依赖源语言
- 代码优化、代码生成

○ 单前端，多后端

○ 多前端，单后端，.NET MSIL

○ 多前端，多后端

单选题 2分

周延辰

设置

编译器报告“缺少运算符”错误是在_____阶段。

A 词法分析
B 语法分析
C 语义分析
D 代码生成
E 代码加载
F 代码执行

提交

语法结构少了组成成分

单选题 2分

周延辰

设置

编译器对常量进行类型转换是在_____阶段。

A 词法分析
B 语法分析
C 语义分析
D 代码生成
E 代码加载
F 代码执行

提交

类型转换是在类型检查阶段中，语义

单选题 2分

将可重定位机器码中相对地址修改为绝对地址是在_____阶段。

A 词法分析
B 语法分析
C 语义分析
D 代码生成
E 代码加载
F 代码执行

提交

编译器之后的连接器等也要看

可重定位机器码——Load阶段

单选题 2分

早期BASIC语言是源程序逐条语句分析执行，因此它是一种_____。

A 预处理器
B 编译器
C 链接器
D 解释器

提交

早期Basic语言：解释器——源程序一条条的读出来，解释执行

编译器——可以脱离源代码

单选题 2分

PASCAL程序执行方式是源程序转换为目标平台的可执行程序再执行，因此它是一种_____。

A 预处理器
B 编译器
C 链接器
D 解释器

提交

雨课堂题目

C++编译器检查相容类型计算是否合规是在_____阶段

- A 词法分析
B 语法分析
C 语义分析
D 代码生成

✓正确

编译器识别出标识符是在_____阶段

- A 词法分析
B 语法分析
C 语义分析
D 代码生成

C++编译器过滤注释是在_____阶段。

- A 词法分析

C++编译器检查数组下标越界是在_____阶段

- A 词法分析
- B 语法分析
- C 代码生成
- D 以上皆错

✓正确

Intel的深度学习编译器nGraph支持TensorFlow、MXNet等深度学习框架，令用户可在Intel CPU、GPU等不同硬件平台上高效运行这些框架编写的程序，它的实现是一种_____方式。

- A 单前端单后端
- B 单前端多后端
- C 多前端单后端
- D 多前端多后端

✓正确

LaTex工具将.tex文件转换为PDF文件，因此它也是一种广义的编译器。这种说法是_____。

- A 正确的
- B 错误的



浏览器渲染过程相当于编译器的 _____ 阶段。

A 分析

B ✓ 综合

C 前端

D 后端

✓ 正确

符号表是在 _____ 阶段创建的。

A ✓ 词法分析

ε是一个 _____。

A 符号

B ✓ 符号串

C 符号串集合

D 以上皆错

以下关于CFG的四元式(V_T, V_N, S, P)，说法正确的是_____。

- (A) $S \in V_T$
- (B) $S \in V_N$
- (C) $S \in P$
- (D) 以上皆错

✓正确

关于一个文法 G 具有二义性的判定，下面错误的是_____。

- (A) 对一个单词串，有两棵语法树
- (B) 对一个单词串，有两个推导
- (C) 对一个单词串，有两个最左推导
- (D) 对一个单词串，有两个最右推导

✓正确

关于语法制导定义中的综合属性，下面说法正确的是_____。

- (A) 父节点属性值依赖孩子节点属性值
- (B) 孩子节点属性值依赖父节点属性值
- (C) 孩子节点属性值依赖兄弟节点属性值
- (D) 以上皆错

✓正确

词法分析——Lex



一个NFA有 n 个状态，转换为等价DFA最多有_____个状态。

A $n \log n$

B n^2

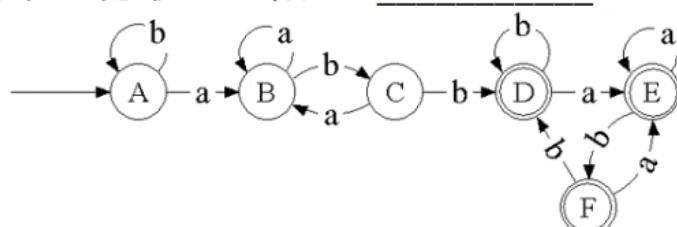
C $n!$

D 2^n

✓ 正确



下面哪个字符串可以区分状态D和E? _____



A a

B b

C bb

D 以上均不对

✓ 正确



下面的正则表达式和上下文无关文法，定义了_____的语言。

$1^*(0 \mid 01)^*$

$S \rightarrow 0A \mid 1S \mid \epsilon$
 $| A \rightarrow 0A \mid 1B \mid \epsilon$
 $B \rightarrow 0A \mid \epsilon$

A

相同

B

不同

✓ 正确

下列语言可以用DFA识别的是_____。

A

形如xx的a、b串集合

B

形如 xx^T (x^T 为x的逆序串) 的a、b串集合

C

C语言的while语句集合

D

SLR(1)文法的活前缀集合

✓ 正确

下列哪个语言可以用正则表达式描述? _____

。

- (A) $\{ a^n b^n \mid n \geq 0 \}$
- (B) $\{ (ab)^n \mid n \geq 0 \}$
- (C) $\{ wcw \mid w \in (a \mid b)^* \}$
- (D) 以上均不对

✓ 正确

下面等式错误的是_____。

- (A) $(0^* \mid 1)^*(2 \mid 3) = 0^*(2 \mid 3) \mid 1(2 \mid 3)$
- (B) $(0^* \mid 1)^*(2 \mid 3) = (0^* \mid 1)^*2 \mid (0^* \mid 1)^*3$
- (C) $(0^* \mid 1)^*(2 \mid 3) = (0 \mid 1)^*(2 \mid 3)$
- (D) $(0^* \mid 1)^*(2 \mid 3) = (0 \mid 1)^*2 \mid (0 \mid 1)^*3$

✓ 正确

1. 正则表达式 (静态) Pattern——单词的描述

语言<—>单词<—>符号串集合

符号串 (句子, 字。字母表中符号组成的有穷序列)

前缀和后缀可以为本身, 也可以为空

子串 (删除前缀或者后缀)

真 (不是空或者自身)

子序列 (可以不连续, 但要保持原来顺序)

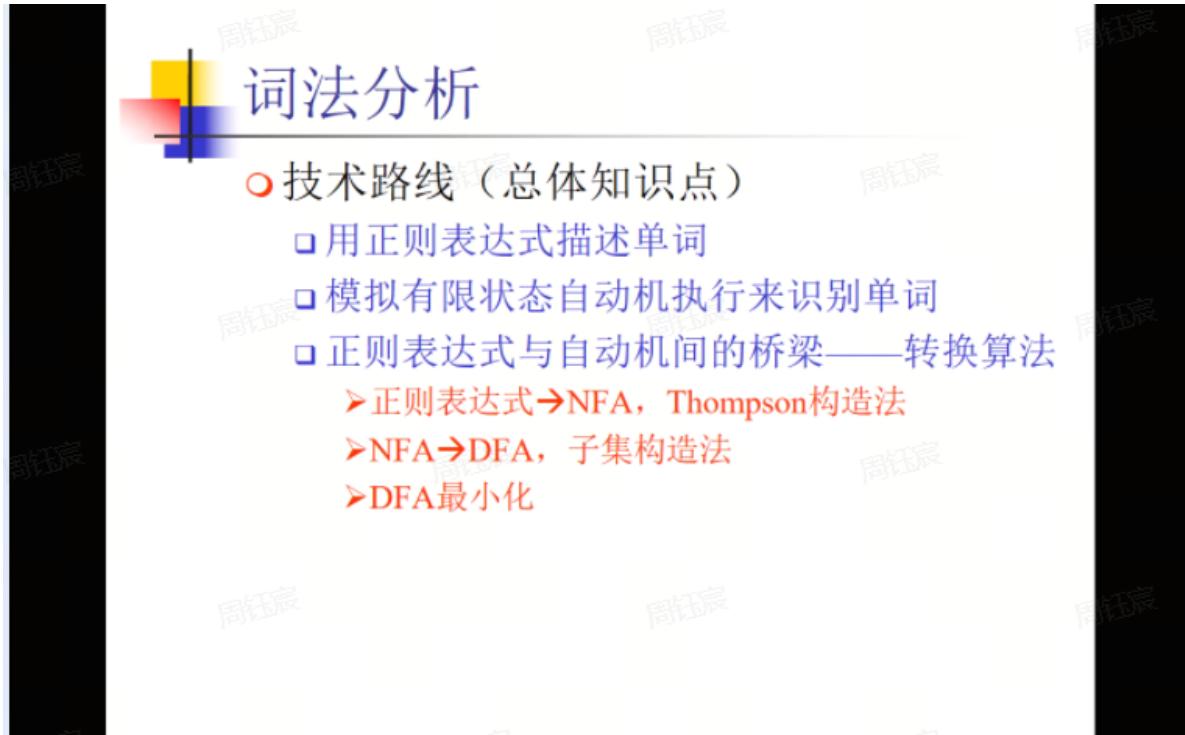
语言——一个给定字母表上的任意 符号串集合 (集合大小可以无限, 每个符号串长度有限)。

([]——空语言, {一部戏龙}——不为空集, 只含空串)

2. FA (动态) Pattern: 对单词的语言结构进行描述

设计正则表达式——模拟FA (算法是固定)

因此正则和FA之间的桥梁, 转换很重要



都是对符号串集合进行操作。某个符号串集合该属于哪个集合。

切成若干段——词素, 要把词素归类为单词 (单词是源程序)

模式 (自然语言描述一段话, 也是模式。但是需要对模式有严谨描述——计算机实现)

正则表达式——设计目的: 描述单词。有穷符号串集合好处理。主要为了解决困难: 标识符, 无符号数。

正则表达式 (正规表达式/正规式/正则式) ——符号串集合运算的表达式 (语言运算的简洁描述)

简单语言——基本正则表达式

语言运算——正则表达式运算

无穷符号串集合需要处理! 无穷符号串集合转换为有限简单符号串集合进行有限次的符号串集合运算得到。

运算对象与结果: 符号串集合。描述集合的运算的系统。

一部戏龙: 长度为0的符号串。特殊! 不是NT也不是T。

正规式 r 所表示的语言 $L(r)$ 。

正规式等价, 表达语言相同。

如果是正则表达式 (表达式符号串集合, 即{一部戏龙}简写为一部戏龙。表示语言{一部戏龙})

正则表达式递归定义: a 的含义也是 $\{a\}$, 长度为1.仅包含这么一个字符的集合。表示语言 $\{a\}$

基本概念

○ 符号串集合！！！

- 单词：类别，符号串分组
- 模式：单词 \leftrightarrow 词素
- 词素：实例

○ 正则式——模式

- 用简单语言（符号串集合）的运算描述复杂语言
- 正则式：语言运算的描述方法

递归结构设计CFG和正则表达式

正则表达式运算（优先级低->高） 符号串集合的运算。正则表达式的运算

| 或 (\cup 并集)

. 连接（符号串集合的连接）

*克林闭包（无穷变有穷， s 的n次幂：自身连接n次的运算。克林闭包相当于 $s^0 \cup s^1 \dots \cup s^n$ ）

() 最高

【字符串集合（语言运算）优先级一致】

//其他：

+正则闭包：排除0次

? : 0次或者1次连接并集

[]: 字符集，本质上还是并集例如[a-Z]，回去看看

1. 最基本形式（先定义基本情况：{一部戏龙}与{a}）

2. 递归：定义运算

正则定义：对子正则表达式起一个名字。更好理解

基本概念一定要清楚，设计题（设计正则表达式或者FA。或者说它接受的串）

正则表达式基本概念

- 字母表、符号串、语言
- 符号串（集合）的运算
- 正则表达式递归定义：|、连接、*、0
- 正则式运算法则，等价
- 正则定义
- 符号简写：+、?、[]

单选题 2分

设置

FIRST(α)= $\{\varepsilon\}$ 中的 ε 表示_____。

- A 字母表中符号
- B 长度为0的符号串
- C 空集
- D 包含一个长度为0的符号串的集合

提交

正则表达式运算理解为符号串集合的，就行。

单选题 2分

设置

s,t,p是相同字母表上的正则表达式，则下面正确的是_____。

A $st=ts$

B $(s \mid t)p = sp \mid tp$

C $(st) \mid p = (s \mid p)(t \mid p)$

D $(s \mid t)^* = s^* \mid t^*$

提交

$0(0 \mid 1)1 \mid 1(0 \mid 1)0$

首位符号相同，必须得至少有两个。提笔就写

主观题 6分

设置

设计正则表达式，接受首尾符号不同的0、1串

正常使用主观题需2.0以上版本雨课堂

作答

S-状态机

S0-初态(符号串集合)

NFA是一个M，定义的语言（NFA接受的符号串集合）——L（M）

F-终态集合（L（M））

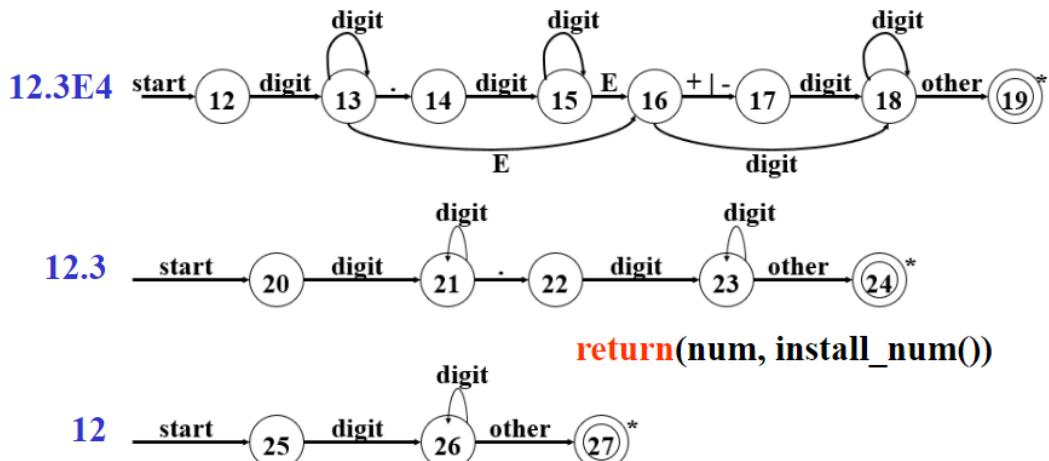
DFA可以有一部戏龙

2^S ——S的所有子集集合

确定的——一个状态发出的不同的边不可能标记相同的符号

例3.9：无符号数

$\text{num} \rightarrow \text{digit}^+ (\cdot \text{digit}^+)? (\text{E} (+|-) \text{digit}^+)?$



返回

有限自动机基本概念

- 非确定有限自动机，NFA
 - 五元组， $M = \{ S, \Sigma, \delta, s_0, F \}$
 $\delta: S \times \Sigma \cup \{\epsilon\} \rightarrow 2^S$
 - 工作方式
- 确定有限自动机，DFA
 - $\delta: S \times \Sigma \rightarrow S$
 - 工作方式
- 表示方式
 - 五元组
 - 状态转换图：节点（状态）、边的含义
 字符串集合！
 - 状态转换矩阵

DFA

优点：时间占优势

缺点：空间不占优势

NFA

通过子集构造法——NFA有n个，DFA最坏会有 2^n 个状态。

优点：与DFA相反。

缺点：

单选题 2分

设置

我们倾向于使用DFA而非NFA构造词法分析器，是因为_____。

- A DFA空间占用优于NFA
- B DFA时间复杂性优于NFA
- C 以上皆对
- D 以上皆错

提交

做法：状态表述。自己划分按照行除以3的余数。符号串集合。

DFA (初态：对应一部戏龙，终态：要是别的)

主观题 6分

设置

设计DFA，接受能被3整除的八进制数

正常使用主观题需2.0以上版本雨课堂

作答

Thompson：正则表达式->NFA

正则到DFA比较复杂。但是NFA时间性能又不好。

基本的正则表达式的语法结构只要被设计出来，其他递归组装即可

本质是：语法制导翻译方法

正则式→NFA

○ 正则式——静态描述→自动机——动态识别； DFA时间复杂性更优， NFA转换更方便——中间桥梁

○ Thompson构造法

□ 语法制导翻译方法

□ 基本正则式→基本NFA

□ 正则式操作→NFA组合构造

不会太复杂。总体上是前后的连接运算。

复习时候要瞬间反应起来。

主观题 8分

设置

对正则表达式 $(a \mid b^*)^*abb$ ，用Thompson构造法转换为NFA

正常使用主观题需2.0以上版本雨课堂

作答

如何体现NFA和DFA等价——任意符号串结果相同（接受或者拒绝）

DFA转出来一个状态。NFA转出来是一个状态集合。这两者相同。

NFA->DFA——子集构造法。

先算初态的一部戏龙闭包。

NFA→DFA——子集构造法

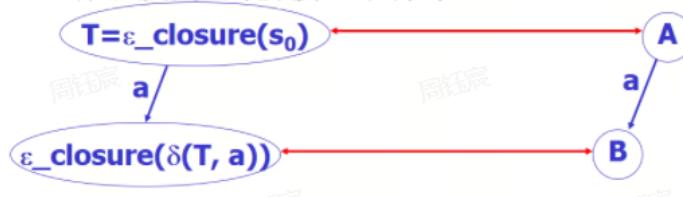
- 得到的NFA与原DFA等价 →

任何符号串 x 在DFA识别结果（单个状态）≡在NFA识别结果（状态集合）→

用NFA状态子集表示DFA来实现DFA构造→

不可能穷举所有“ x ”——由简单到复杂：

长度为0、长度为1、...，长度为 $k+1$ 的符号串看作长度为 k 的拼接一个符号

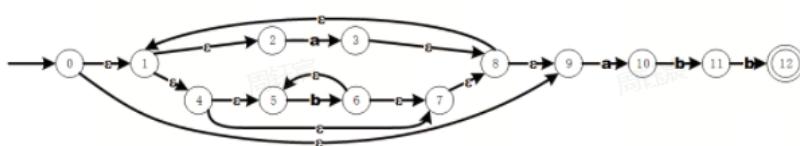


子集构造法（最复杂的？？

主观题 12分

设置

对正则表达式 $(a \mid b^*)^*abb$ ，将Thompson构造法得到的NFA用子集构造法转换为DFA，用其识别 $abbaabb$



正常使用主观题需2.0以上版本雨课堂

作答

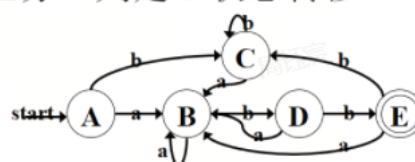
最小化DFA

最小化DFA

- 核心思想：“区分”概念→不可区分的状态合并

○ 实际算法：从状态全集开始，符号串 x 可区分→状态集分裂

○ 不可能穷举所有符号串，由简单到复杂，长度 $k+1$ 看作单个符号拼接长度 k
“区分”判定→状态转移



做状态迁移。AC不可区分了。最开始分出去D，然后分出去B。

主观题 6分

将前面得到的DFA最小化

正常使用主观题需2.0以上版本雨课堂

作答

Chapt 2

Chapt 3

Chapt 4

语法分析基本概念

- 上下文无关文法，CFG，四元式(V_T, V_N, S, P)
 $A \rightarrow \alpha$
- 推导、语言、句型、句子
- 最左推导、最右推导
- 语法树、二义性文法
- CFG的等价

VT VN P S

CFG与正则表示式

CFG>正则

- 正则——有限的重复与给定结构的无限重复

——不能描述：平衡或者嵌套的结构：



下列哪个语言可以用正则表达式描述? _____

。

- (A) $\{ a^n b^n \mid n \geq 0 \}$
- (B) $\{ (ab)^n \mid n \geq 0 \}$
- (C) $\{ wcw \mid w \in (a \mid b)^* \}$
- (D) 以上均不对

✓ 正确

• CFG——

正则不行, CFG行。两者都不行的例子需要熟悉。

与词法分析的联系

- CFG和正则式
 - 正则式、NFA/DFA改写为等价的文法
(正则文法, 3型文法)
 - 正则式可描述的语言
 - 正则式不能而CFG能描述的语言
 - 均不能描述的

一部戏龙和普通边转换?

回去听录音

为什么要改写后面:

预测分析法: 没有一部戏龙, 也没有左公因子

CFG的设计

- 消除左递归：直接、间接
- 消除 ϵ 产生式
- 消除回路
- 提取左公因子
- 为什么要做这些改写？

1. 消除左递归：直接、间接

2. 消除一部戏龙产生式

3. 消除回路：

4. 提取左公因子：

5. 改写的原因：

设计CFG。递归结构必须会写。不会比讲义和作业难

aibjck(j=i+k)作业题

递归分别设计aibi等。 S->一部利空 | aSb

分为j和k大小关系不同的两种拆分

主观题 6分

设置

设计接受语言 $\{a^i b^j a^k b^l \mid i+j=k+l, i, j, k, l \geq 0\}$ 的上
下文无关文法

正常使用主观题需2.0以上版本雨课堂

作答

设计上下文无关文法接受以两个0结尾的01串

正常使用主观题需2.0以上版本雨课堂

作答

自顶向下语法分析

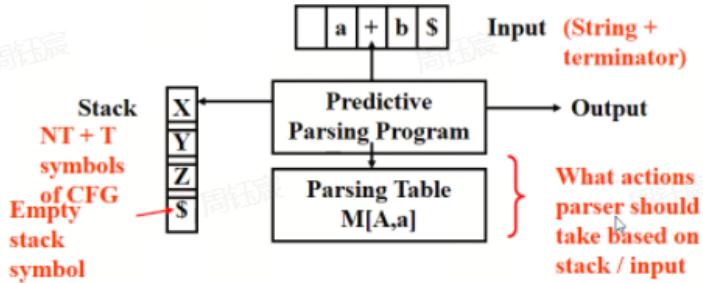
基本概念（选择题），大题对作业难度题算法比较熟悉

自顶向下语法分析（重点）

- 寻找最左推导
语法树构造根→叶
由整体→局部
- 预测分析法
 - 当前输入符号+待扩展NT→避免回溯
- 递归下降法构造预测分析器：
 $NT \rightarrow$ 递归函数

非递归实现

○ FIRST、FOLLOW
→ 预测分析表，LL(1)文法



细节

非终结符：STU

终结符

开始默认是最左部

一部戏龙，是空串，既不是NT也不是T！！！！！！

主观题 3分

对下面CFG，指出终结符集合、非终结符集合、开始符号

$S \rightarrow \%aT \mid U!$
 $T \rightarrow aS \mid baT \mid \epsilon$
 $U \rightarrow \#aTU \mid \epsilon$

正常使用主观题需2.0以上版本雨课堂

作答

一部戏龙写在First中，只是占位符。表示可以推导出空串。

S和T相互包含一致。

FOLLOW中不要加入一部李龙，不可能有空，是实际的非终结符！！！！

预测分析表绝对不要画一部戏龙的列！！！

一部戏龙产生式不一定只是 $S \rightarrow$ 一部戏龙！！！可以是推导好多次

具体的句子分析也要会！缓冲区\$开始，栈最开始 \$ 和S

T->一部戏龙分析时候遇到。弹出T，不要压入一个一部戏龙。不是实际上的字符。

两个 \$ 碰头-Accept

主观题 12分

对下面CFG，构造预测分析表，分析句子 #abaa%a!

$S \rightarrow \%aT \mid U!$

$T \rightarrow aS \mid baT \mid \epsilon$

$U \rightarrow \#aTU \mid \epsilon$

正常使用主观题需2.0以上版本雨课堂

作答

何时规约：栈顶句柄

句柄定义：最右推导的逆过程。产生式的一个右。

算符还是LR都是基于活前缀。

活前缀：都是最右句型的前缀。啥啥不超过。末端栈顶。保证栈中一直是活前缀。

自底向上语法分析

- 寻找最右推导
语法树构造叶→根
由局部→整体
- 句柄
 - $S \xrightarrow{*} \alpha Aw \xrightarrow{*} \alpha \beta w$
- “移进—归约”分析方法
 - 基本操作：移进、归约、接受、错误

明确知道优先级关系——自己写。

利用产生式画局部语法树构造优先级关系表。

算符文法

1. product右边不会有连续的NT (连续的子表达式，即没有运算)
2. 没有一部戏龙

算符优先分析方法

- 算符文法
- 算符优先级：低于、高于、等于——
算符优先文法
- 句柄的确定

□ < = ... = >

核心——子集构造法。

活前缀

项目：.就是活前缀的末尾，栈顶。当前阶段性目标进展到已经把X搞到栈顶了。

SLR分析法（重点）

- LR(0)项目： $A \rightarrow X \cdot YZ$
 - 已分析得到X，期待分析得到YZ
- 构造识别活前缀的DFA
 - LR(0)项目集——DFA状态
 - LR(0)项目集规范族 \rightarrow DFA
 - closure函数：存在 $A \rightarrow \alpha \cdot B\beta \rightarrow$ 添加 $B \rightarrow \cdot \gamma$
 - goto函数： $\text{goto}(I, X) \rightarrow A \rightarrow \alpha \cdot X\beta \rightarrow A \rightarrow \alpha X \cdot \beta$

分析表的归约，跟随符号放到其中。

规范LR分析法

- SLR分析能力不够
- LR(1)项目： $[S \rightarrow aA \cdot B e, c]$

C不在任何产生式右边——无用。

只有最后一个错了。句柄是a

单选题 2分

对下面上下文无关文法，下列说法不正确的是

$$S \rightarrow aA \quad A \rightarrow Bb \quad B \rightarrow Ba \mid a \quad C \rightarrow Ab$$

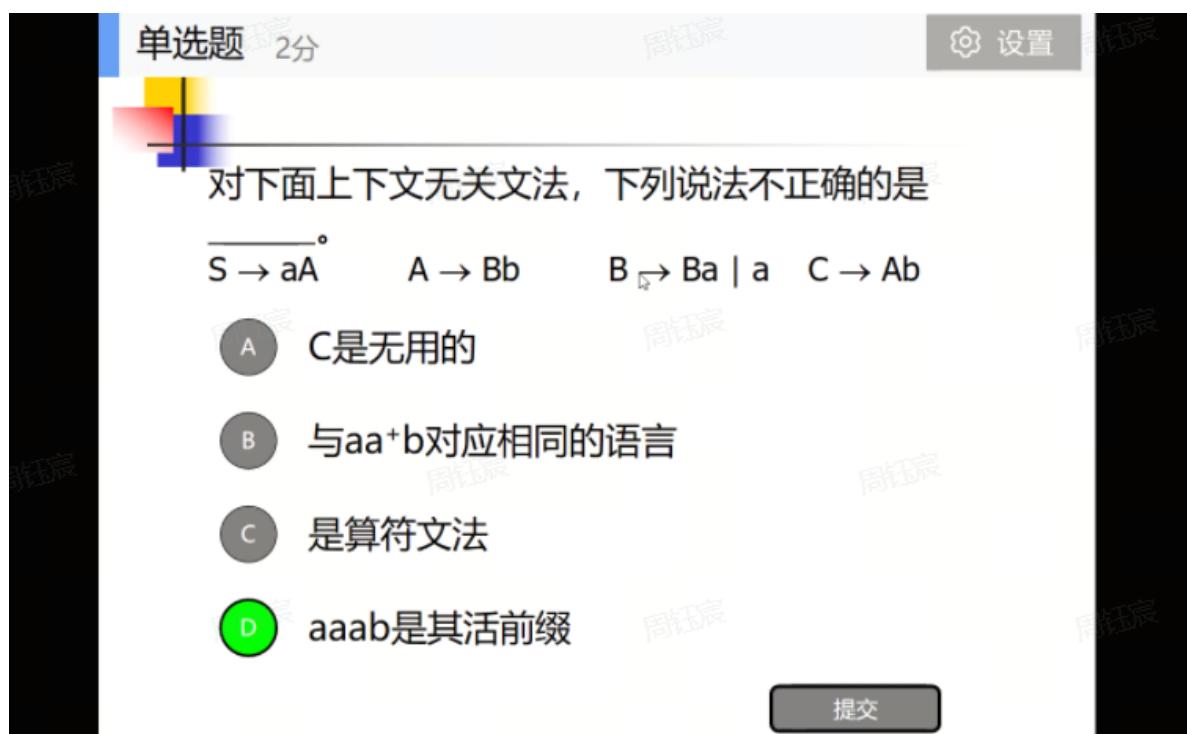
A C是无用的

B 与 aa^+b 对应相同的语言

C 是算符文法

D aaab是其活前缀

提交



这里少一道。

预测分析，算符，LR。选其中之一。

题型示例（续）

四、(11分) 给定上下文无关文法。

$$E \rightarrow id$$

$$E \rightarrow id(E)$$

$$E \rightarrow E + id$$

拓广文法，计算 LR(0)项目集规范族，构造 SLR 分析表，它是 SLR(1)文法吗？

解：

拓广文法。

(0) $S \rightarrow E$

(1) $E \rightarrow id$

(2) $E \rightarrow id(E)$

(3) $E \rightarrow E + id$

文法拓广！！

题型示例（续）

构造 SLR 分析表。

\varnothing	action \varnothing						goto \varnothing
\varnothing	$id\varnothing$	$(\varnothing$	$)\varnothing$	$+\varnothing$	$\$\varnothing$	$E\varnothing$	\varnothing
0 \varnothing	s2 \varnothing	\varnothing	\varnothing	\varnothing	\varnothing	$I\varnothing$	\varnothing
1 \varnothing	\varnothing	\varnothing	\varnothing	s3 \varnothing	acc \varnothing	\varnothing	\varnothing
2 \varnothing	\varnothing	s4 \varnothing	r1 \varnothing	r1 \varnothing	r1 \varnothing	\varnothing	\varnothing
3 \varnothing	s5 \varnothing	\varnothing	\varnothing	\varnothing	\varnothing	\varnothing	\varnothing
4 \varnothing	s2 \varnothing	\varnothing	\varnothing	\varnothing	\varnothing	$6\varnothing$	\varnothing
5 \varnothing	\varnothing	\varnothing	r3 \varnothing	r3 \varnothing	r3 \varnothing	\varnothing	\varnothing
6 \varnothing	\varnothing	\varnothing	s7 \varnothing	s3 \varnothing	\varnothing	\varnothing	\varnothing
7 \varnothing	\varnothing	\varnothing	r2 \varnothing	r2 \varnothing	r2 \varnothing	\varnothing	\varnothing

分析表没有冲突，所以它是 SLR(1)文法。

Chapt 5

Chapt 6 类型检查



类型检查

○类型表达式

- 基本类型表达式: int、...、void、type_error、
 类型名

- 类型构造符: 数组、指针、函数、结构、...

○类型表达式等价

- 结构等价: 语法树等价——内在

- 名字等价: 形式等价——外在

Chapt 7

Chapt 8

Chapt 9

Chapt 10

其他需要掌握的。

了解基本概念。

语法制导翻译需要掌握的知识点。

难度就是作业难度。

不考大题。

语法制导翻译

- 综合属性、继承属性
- S—属性定义
 - 仅综合属性
- L—属性定义
 - 继承属性依赖父结点，或左兄弟结点
- 语法制导定义/翻译模式的设计

单选题 2分

设置

综合属性计算的依赖关系是父节点依赖孩子节点，所以综合属性的计算_____。

- A 容易与预测分析法相结合
- B 容易与算符优先分析算法相结合
- C 以上皆对
- D 以上皆错

提交

语法制导定义的核心思想

前两个属于基本正则表达式，调用makeleaf。

应该说明把明把子树存在哪。ptr这个东西类似地。

比如R'|'R: R.ptr = makenode('|', R1.ptr, R2.ptr)

回去看算术表达式的。

下面文法描述了正则表达式，设计语法制导定义实现构造正则表达式对应的表达式树。假设已有辅助函数mkleaf(char)（及mkleaf_epsilon()）和mknnode(op, child1, child2)分别为基本正则表达式和正则表达式运算创建叶节点和内部节点

$$R \rightarrow \text{char} \mid ' \epsilon ' \mid R \mid R \cdot R \mid R^* \mid (R)$$

正常使用主观题需2.0以上版本雨课堂

作答

设计其CFG或者

类型表达式运算。结构化类型。

等价概念搞清楚。

类型检查

○类型表达式

- 基本类型表达式：int、...、void、type_error、类型名

- 类型构造符：数组、指针、函数、结构、...

○类型表达式等价

- 结构等价：语法树等价——内在

- 名字等价：形式等价——外在

C——对函数结构等价。对struct名字等价。

Pascal——默认名字等价。

整体上是函数，参数也是函数。C语言参数只能传递只能传递函数指针



关于下面类型表达式，正确说法是_____。
 $(\text{pointer}(\text{char}) \rightarrow \text{int}) \rightarrow \text{pointer}(\text{char})$

- A C语言对这种类型的等价判定采用名字等价方式
- B Pascal语言对这种类型的等价判定采用结构等价方式
- C 在C语言中，这种类型会引发类型错误
- D 以上皆错

提交



中间代码生成考察：作业题



中间代码生成

- 表达式翻译
- 临时名字重用
- 布尔表达式和控制流语句的翻译
- 真假值出口

临时名字重用：c计数器

布尔表达式（短路+not>and>or+goto隔一句）

控制流加入布尔 (while和if_else都是连着两个goto, **while一定记得最后结尾有个goto回去！！！**)
程序之外的可以用Lnext替代!

题型示例

9. 将下面表达式转换为三地址码，采用临时变量重用算法，可将临时变量数目减少到 A 个。
 $a + (a + a + a * (a + a + a))$
- A. 2 B. 3
C. 4 D. 5

代码优化

代码优化

- 基本块、流图、循环、下次引用
- 基本优化方法
 - 消去公共子表达式
 - 复制传播
 - 无用代码删除
 - 循环优化
 - 代码外提
 - 强度削弱
 - 归纳变量删除

1. 基本块构建——找leader

- 程序第一条语句（看题目有没有提到程序）
- 无/有条件跳转目的语句
- 无/有条件跳转下一条语句（容易忘！）

后面代码基本块不用显式构造

2. 流图：前驱与后继

看一个基本块最后一个语句：

- 有 有/无条件跳转到B2——B2（可能是false分支的无条件跳转）1
- 无 有/无条件跳转，紧跟B3——B3

后面代码基本块通过goto Lnext语句指向空气即可。

3. 循环：

- 这组节点是强连通的（任何两个节点之间都存在一条完全包含在循环内的路径）
- 这组节点具有唯一的一个入口（entry）——从循环外的节点到达循环内的节点，唯一的途径是先到达入口节点

4. 下次引用：从后往前扫描。

遇到指令 i, $x = y \text{ op } z$:

- 在指令后加入引用信息（使用/引用(use)）：语句 i 赋值 x, j 以 x 作为运算对象，且 i 到 j 无其他赋值语句
- x 非活跃，无下次引用
- y 活跃，有下次引用为 i

后两步不可以交换。最后得到临时变量生存周期。数据流分析）临时变量名字的重用与存储分配。
嵌套不可以，一上一下可以。

5. 基本优化方法(global与local):

- 消去公共子表达式 (local)：要求两个重复的表达式之间没有对 operand 的定义（再赋值）。
- 复制传播：
 - 单路径传播：直接利用复制的源头（赋值语句的源头，即赋值语句的右式）
 - 多路径传播：两个路径有相同右式的赋值表达式。不管从哪个路径来，用同一个临时变量代替。
- 无用代码删除：

利用复制传播，把变量直接替换为源头的常量（debug->false，也就做常量合并）。之后可以删除一些语句（源头的赋值语句，或者 if(false) 语句）

◦ 循环优化：

- 代码外提：提至循环外。

某些代码（表达式计算）与循环次数无关【或者说做的是相同的赋值操作】，即循环不变计算。

或者和循环条件有关的变量在循环内部没有对其的定义（赋值运算）——> 循环不变计算。

eg. `while(t <= limit - 2)`

从循环内提到循环入口前。具体而言，提到一个前驱自定义基本块中。

- 强度削弱：与 i, j, k 多有关。乘方 > 乘除 > 加减。

同步变化的归纳变量，记住可能要多加使用某个归纳变量的赋值。

eg. `j = j - 1 ; t4 = 4 * j --> t4 = 4 * j ; j = j - 1 ; t4 = t4 - 4 ;`

- 归纳变量删除：归纳变量即 i, j, k。

- 如果前面有归纳变量给临时变量的赋值，则用临时变量替换。（PPT）
- 如果之前没有，新加入临时变量的赋值，必须能减少一些。

9.6 和 9.7 的寄存器分配。



目标代码生成

- 目标代码生成算法
 - 基本块内局部最优
 - 计算结果尽量保存在寄存器内
- 寄存器分配优化
 - 着眼于循环，选定某些变量一直放于寄存器

1. 目标代码生成算法（掌握）
2. 寄存器分配优化算法（理解）