


# [编译原理]LL(1)分析法+例题 学习

原创 踩踩芽 已于 2023-12-28 20:27:22 修改 阅读量1.4w 收藏 156 点赞数 32

分类专栏: 笔记 文章标签: 编译原理

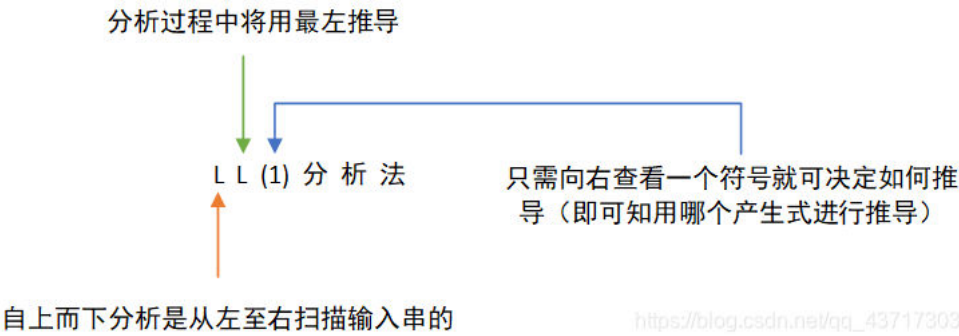
版权

 笔记 专栏收录该内容

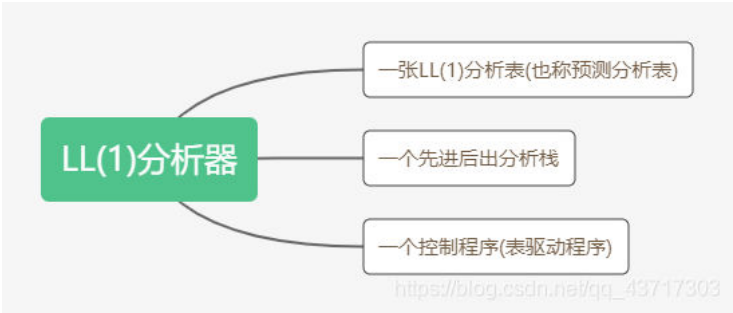
0 订阅 2 篇文章 订阅专栏

## 一、LL(1)分析法

LL(1) 分析法又称预测分析法，是一种 不带回溯的非递归自上而下 分析法。



## 二、LL(1)分析器



## 三、LL(1)分析表

行：终结符和界符(##)

列：非终结符

行数：终结符数+界符数(1)+1

列数：终结符数+1

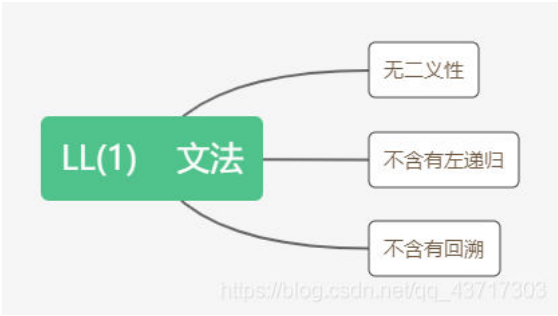
	i	+	*	(	)	#
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow i$			$F \rightarrow (E)$		

仅包含一个候选式的产生式

空白 (ERROR) : 推导不出来

[https://blog.csdn.net/qq\\_43717303](https://blog.csdn.net/qq_43717303)

四、LL(1)文法：分析表M不含多重定义入口的文法



[https://blog.csdn.net/qq\\_43717303](https://blog.csdn.net/qq_43717303)

- 1、一个LL(1)文法所定义得语言恰好就是它的分析表所能识别的全部句子。
- 2、一个上下文无关文法是LL(1)文法的充要条件（判断一个文法是否是LL(1)文法）：对每一个非终结符A的任何两个不同的产生式 $A \rightarrow \alpha | \beta$ ，有下面条件（都是避免了多重入口）成立
- (1)  $FIRST(\alpha) \cap FIRST(\beta) = \emptyset$ ：A的每个候选是都不存在相同的首字符

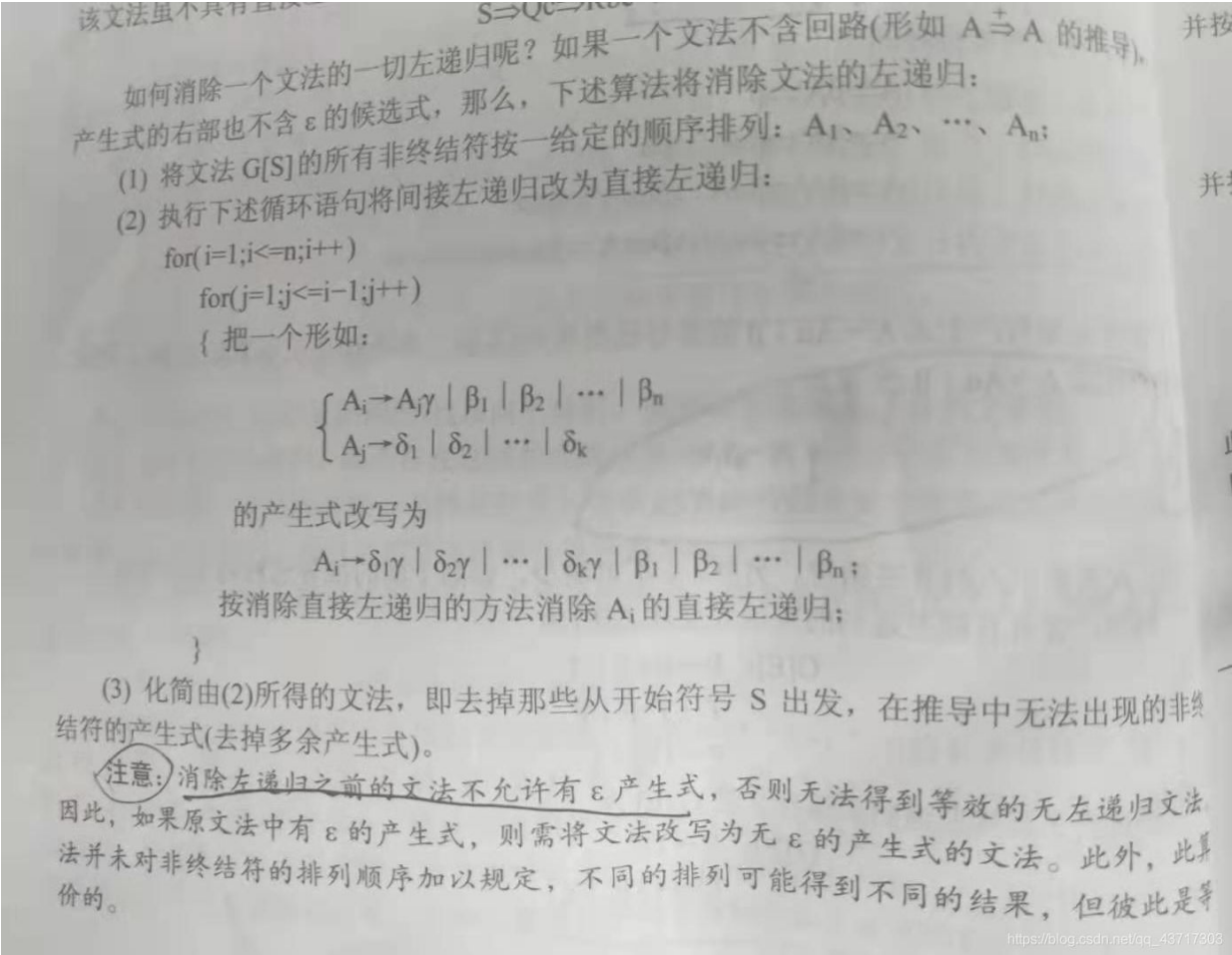
(2) 假若 $\beta \xrightarrow{*} \epsilon$ ，则有 $FIRST(\alpha) \cap FOLLOW(A) = \emptyset$ ：避免了在分析表同一栏目内出现 $A \rightarrow \alpha$ 和 $A \rightarrow \epsilon$ 的情况。

五、给出算术表达式文法求某输入串的分析过程求解步骤

1、消除左递归（可利用数学中的分配律）

- 形如  $A \rightarrow A\alpha | \beta$  其中， $\alpha$ 、 $\beta$ 是任意的符号串且 $\beta$ 不以A开头。这时，可将A的产生式改写为右递归形式：
$$\begin{cases} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' | \epsilon \end{cases}$$
- 一般的， $A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_m | \beta_1 | \beta_2 | \dots | \beta_n$ 有
$$\begin{cases} A \rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_n A' \\ A' \rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_m A' | \epsilon \end{cases}$$

消除一切左递归



2、消除回溯

- 回溯产生的原因：候选式存在公共的左因子
- 一般的，设文法中关于 A 的产生式为 $A \rightarrow \delta\beta_1 | \delta\beta_2 | \dots | \delta\beta_i | \gamma_1 | \dots | \gamma_j$ 那么，可以把这些产生式改写为：
$$\begin{cases} A \rightarrow \delta A' | \gamma_1 | \dots | \gamma_j \\ A' \rightarrow \beta_1 | \dots | \beta_i \end{cases}$$
  - ① 提取公共左因子： $A \rightarrow \delta(\beta_1 | \beta_2 | \dots | \beta_i) | \gamma_1 | \dots | \gamma_j$
  - ② 将产生式中由“(”和”)”括起的部分以非终结符 A' 命名 则得到上述结果。

3、求解文法的FIRST集和FOLLOW集（方法：[https://blog.csdn.net/qq\\_43717303/article/details/110210180](https://blog.csdn.net/qq_43717303/article/details/110210180)）

4、构造LL(1)分析表

- 首先求出每个非终结符的 FIRST 和FOLLOW 集
- 然后按以下四个步骤构造分析表
  - ①对文法 G 的每个产生式  $A \rightarrow \alpha$  执行 ② 和③ 步；
  - ②对每个终结符  $a \in FIRST(A)$ ，把  $A \rightarrow \alpha$  加至  $M[A, a]$ 中，其中  $\alpha$  为 含有首字符 a 的候选式或唯一的候选式
  - ③若 $\epsilon \in FIRST(A)$ ，则对任何  $b \in FOLLOW(A)$  把  $A \rightarrow \epsilon$  加至  $M[A, b]$ 中
  - ④把所有无定义的  $M[A, a]$ 标“出错 标志”。

5、若预测分析表 M 含有多重定义入口冲突项，则该文法不是LL(1)文法。遵从就近匹配原选定唯一候选式得到无二义的LL(1)分析表。

6、输入串分析过程：分析开始时栈底先放入一个“#”，然后再压入文法的开始符号；当分析栈中仅剩“#”，输入串指针也指向串尾的“#”时，分析成功。

符号栈		输入串		所用产生式			
符号栈		当前输入符号		输入串		说明	



符号栈	当前输入符号	输入串	所用产生式	说明
#E'T'i	$i_2$	$i_2+i_3\#$	$F \rightarrow i$	弹出 F, 将 i 压栈
#E'T'	$i_2$	$i_2+i_3\#$		匹配, 弹出栈顶符号 i 并读出输入串的下一个输入符号 +
#E'	+	$+i_3\#$	$T' \rightarrow \epsilon$	弹出 T', 因M[T,+]中为T' $\rightarrow \epsilon$ , 故不压栈
#E'T+	+	$+i_3\#$	$E' \rightarrow +TE'$	弹出 E', 将 +TE' 逆序压栈
#E'T	+	$+i_3\#$		匹配, 弹出栈顶符号 + 并读出输入串的下一个输入符号 $i_3$
#E'T'F	$i_3$	$i_3\#$	$T \rightarrow FT'$	弹出 T, 将 FT' 逆序压栈
#E'T'i	$i_3$	$i_3\#$	$F \rightarrow i$	弹出 F, 将 i 压栈
#E'T'	$i_3$	$i_3\#$		匹配, 弹出栈顶符号 $i_3$ 并读出输入串的下一个输入符号 #
#E'	#	#	$T' \rightarrow \epsilon$	弹出 T', 因M[T,#]中为T' $\rightarrow \epsilon$ , 故不压栈
#	#	#	$E' \rightarrow \epsilon$	弹出 E', 因M[E,#]中为E' $\rightarrow \epsilon$ , 故不压栈
分析栈中仅剩"#", 输入串指针也指向串尾的"#", 分析成功。				

显示推荐内容



踩踩芽

关注

32 0 0

3 条评论



我尊的不会写代码 热评 为什么 (E', ) ) 那里要写 E'  $\rightarrow \epsilon$ 啊

写评论

LL(1)分析法(二) ——预测分析表构建与语法分析实现(编译原理) 12-26

LL(1)预测分析表生成 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是终结符,M[A,a]存放的是一条A的产生式,指出当面对输入字符a时所...

编译原理:LL(1)文法-构造预测分析表\_11(1)分析表 12-28

(1)文法不含左递归 (2)对文法中每一个非终结符A的各个产生式的FIRST集合两两不相交。 (3)对文法中每一个非终结符A,若存在某个FIRST集合包含...

【编译原理】LL1分析表的构造 g14365893的博客 1万+

首先要构造FIRST集合和FOLLOW集合 例题 首先构造比较简单的FIRST集合 上面两个的FIRST集合非常好计算 直接提取候选式最开始的终结符即...

【编译原理】LL(1)分析法: C/C++实现 最新发布 Sar Programmer的博客 1万+

史上最全的编译原理之LL(1)分析法实验讲解汇总,这篇万字详解是您深入了解编译原理中LL(1)分析法的最佳选择! 它提供了超全面、超详细的内容...

编译原理:LL(1)文法 语法分析器(预测分析表法)\_使用实现的llparserexamp... 12-24

设计要求:对于任意输入的一个LL(1)文法,构造其预测分析表,并对指定输入串分析其是否为该文法的句子。 思路:首先实现集合FIRST(X)构造算法和...

编译原理学习笔记·语法分析(LL(1)分析法/算符优先分析法OPG)及例子... 12-28

接着便是构造优先关系表 给几个语法分析的例题 再看一个例子,判断某个上下文无关文法是否是LL(1)分析文法(或者将某个文法改造成LL(1)分析文法...

LL(1)文法系列(一)first集和follow集 shadowam的博客 5037

Problem Description 已知文法G[S]的表达式, 计算文法中终结符的first集和follow集。在文法G[S]中使用'@'代表空。 现在我们规定文法G[S]中每个...

语法分析: LL(1)分析 热门推荐 kafmws的博客 1万+

本篇介绍的LL(1)分析,这是一种自上而下分析的方法 关于语法描述的概念 自上而下分析 从文法的开始符号出发, 向下推导。推出句子 根据输入串...

LL(1)文法判断例题记录\_预测分析表无多重入口是什么意思 12-16

select(M->K)=First(k)-{ε} ∪ Follow(M)={d,e,#,α} select(M->bLM)={b} select(M->K)∩select(M->bLM)=ε 这里可以推出是LL(1)文法 预测分析表: 预测分...

【编译原理】最全复习大纲及例题\_画出编译程序的总体结构图 12-29

3、该文法的每个非终结符,都有某个候选首符集包含ε,则First(T)∩Follow(T)=空 4、给出输入串的分析过程(反正压入) 5、画出LL(1)的预测分析表 五...

LL(1)分析表的构造 Acapplella的博客 1476

例题 LL(1)分析表如下图: 注: acc意为accepted, 接受的意思

编译原理LL(1)语法分析实验报告 编译原理LL(1)语法分析实验报告	05-10
语法设计——基于LL(1)文法的预测分析表法 通过对基于LL(1)文法的预测分析表法DFA模拟程序实验,使学生掌握确定的自上而下的语法分析的实现技术,及具体实现方法。通过本实验加深对语词...	12-25
...FOLLOW集,预测分析表算法思想_II(1)文法c++ 根据某一文法编制调试LL(1)分析程序,以便对任意输入的符号串进行分析。本次实验的目的主要是加深对预测分析LL(1)分析法的理解。 简单要求: 至...	12-19
编译原理实验报告LR(1)分析法.doc 编译原理实验报告LR(1)分析法	05-11
编译原理预测分析LL(1)文法 JAVA实现 1、项目开发工具为MyEclipse2014 2、使用了Struts2框架作为控制层 3、HTML5图形用户界面 4、JAVA语言写后端 该项目完成的功能有: 自动消...	06-21
编译原理之NULL集、first、follow集C语言实现 编译原理之NULL集、first、follow集C语言实现,实现中句子的转换符号由#代替,数组默认由""作为结束符	12-10
编译原理 First集和Follow集 文法 预测分析表 对文法拆分 并求First集和Follow集和预测分析表	11-16
First集和Follow集生成算法模拟 编译原理课程设计First集和Follow集生成算法模拟 【问题描述】 设计一个由正规文法生成First集和Follow集并进行简化的算法动态模拟 【基本要求...	05-08
【编译原理】典型题型: 判断一个文法是否为LL(1)文法并构造预测分析表 (First集和Follow集的求解方... skawngur111的博客 3987 判断一个文法是否为LL(1)文法并构造预测分析表 (First集和Follow集的求解方法&LL(1)文法的判断条件&预测分析表的构造) 目录判断一个文法是...	
编译原理实验LL(1)分析法 LL(1)分析法 LL(1)分析法又称预测分析法,是一种不带回溯的非递归自上而下分析法。LL(1)的含义是: 第一个L表明自上而下分析是从左至右扫描输...	qq_52252193的博客 7274
求文法的FIRST集, FOLLOW集 之前学编译原理做实验做了一个求first集, follow集的程序, 一直想在网上保存一下, 最近才有时间, 之前做时也算是花了好几天。也算是体验了一...	放下矜持的博客 2669
编译原理——第四章-LL(1)文法的判别条件、构造文法的预测分析表 LL(1)文法的判别条件 若一个文法满足以下条件, 则称该文法G为LL(1)文法:	大胖东的博客 1万+
编译原理II1预测分析表 LL(1)预测分析表是一种自顶向下的语法分析方法, 它是基于一个文法的First和Follow集合构造出来的。它可以用于分析LL(1)文法, 即文法中的任意...	06-08