从开始符号到输入串的推导 🕜 实现过程 浪费大量语法、语义分析工作 回溯 危害 效率低 出现的问题 难以确定出错位置 左递归 危害 陷入无限循环 L: 自左向右扫描 含义 L:最左推导 (1)每次读取一个符号 LL(1)分析法 自上而下分析方法 若FIRST集相交,提取左公因式 消除回溯 若FIRST集不相交,根据所读符号进行选择 🐼 解决出现的问题 直接消除 不含左递归的式子放左边 消除左递归(改写文法) 间接左递归转换成直接左递归后消除 间接消除 例有 G[S]: S→aAb|b A→**|* L(G[S])={ b, <u>a**b</u>, a*b} 改写G[S]为 S→aAb|b A→* (*| ε) N N N SYM= 'a SYM= Y ADVANCE SYM= '* 出错 Y ADVANCE N SYM= '*' ADVANCE 读符号 N SYM= 'b - 田错 2402 FFF6EER ROOK AND N ADVANCE 出口 HI 递归下降分析程序构造法 种类 文法G[E]的预测分析表: (1)分析栈倒序写入 (2)分析栈顶和余留串符号相 替换规则 余留串 |分析栈 (1) #E E-+TE' i*i# (2) #E'T i*i# T-FT' (3) #E'T'F i*i# F⊸ri ☑ 预测分析方法 (4) #E'T'i i*i# (5) #E'T' 源程序 词法分析器 取下一 单词 *i# T'← *FT' 后续部分 (6) #E'T'F* *i# (7) #E'T'F i# F⊸ri (8) #E'T'i i# 符号表 (9) #E'T' 3 -- 'Τ 语法分析在编译原理程序中的地位 语法分析方法 (10) #E' Ε'-- ε 预测分析程序 (11) # 语法分析 求取FIRST 产生从开始符号到输入串的推导序列 从右边找符号 语法分析原理 产生从开始符号为根、输入串为叶子的语法树 符号右边的非终结符号找FIRST-{ε} 预测分析表的构造 所求非终结符号右边有ε时,加入**FOLLOW(A)** 按照**FIRST**集,如果**FIRST**中有ε,加入 FOLLOW集 定义 从输入串开始逐步进行规约直至开始符号 最左规约是最右推导的逆过程 规约与分析树 每步规约是某产生式的右部 移进、规约 🐼 自上而下分析的实现 问题:不能确定可归约串 短语 终结符号的一次规约就能产生的式子 🐼 规范规约 直接短语 句柄 一个句型的最左直接短语称为句柄 不是规范规约 特点 定义 根据优先关系寻找可归约串 例: E -> E+E | E*E | (E) | i 自下而上分析方法 1) 如果当前所读a是运算对象,则进OPND。 2) 如果当前所读a是运算符,则OPTR 运算 栈内运算符与栈外当前所读a比较优先级, 算符优先分析 若, 栈内运算符 〈 栈外运算符 : 进OPTR。 栈内运算符 > 栈外运算符 : 退栈,归约 出一个运算符 栈内运算符 □ 栈外运算符 出两个 运算对象 分析方法 算符文法(OC) 🧭 算符优先文法(OPG) 算符优先文法定义 优先关系定义 算符优先文法(OPG) 手工构造 🐼 优先表的构造 自动生成 最常见 种类 文法限制少 优点 大多数上下文无关文法都可用 效率高 LR分析法 能够及时发现错误 储存容量大 缺点 手工实现工作量大 🐼 LR(0)项目 拓广文法 作用 让推导结果更正确 (1) $A \rightarrow \alpha$. 归约项目 LR(0)分析法 (2) S'→ S· 接受项目 (3) $A \rightarrow \alpha \cdot X \beta$ 当X∈ Vt, 为移进项目 当X∈Vn, 为待约项目 句柄识别程度

自上而下地产生输入串的语法树

🕜 LR(O)分析表的构造

实现原理