

学院：数据科学与计算机学院

专业：计算机科学与技术

姓名：郑康泽

学号：17341213

编译原理

TINY+编译器构造

一. 词法分析程序实验

实验目的：通过扩充已有的样例语言TINY语言的词法分析程序，为扩展TINY语言TINY+构造词法分析程序，从而掌握词法分析程序的构造方法

实验内容：了解样例语言TINY及TINY编译器的实现，了解扩展TINY语言TINY+，用C语言在已有的TINY词法分析器基础上扩展，构造TINY+的词法分析程序

实验要求：将TINY+源程序翻译成对应的TOKEN序列，并能检查一定的词法错误

以下是扩展的TINY+词法定义：

1. TINY+语言的关键字包括：

or	and		
int	bool	string	
while	do		
if	then	else	end
repeat	until		
read	write		

2. 特殊符号的定义如下：

<	>	<=	>=	=
+	-	*	/	:=
{	}	()	'
,	;			

3. 其他种类的单词包括标识符ID，数字NUMBER以及字符串STRING，他们的正规表达式的定义如下：

$ID \rightarrow \text{letter} (\text{letter} \mid \text{digit})^*$

标识符是以字符开头，由字母和数字混合构成的符号串。

$NUMBER \rightarrow \text{digit} \text{ digit}^*$

$STRING = \text{'any character except'}$

一个字符串类型的单词使用单引号括起来的字符串'...'，引用内可出现除了'以外的任何符号，一个字符串不能跨行定义。

$\text{letter} \rightarrow \text{a} \mid \dots \mid \text{z} \mid \text{A} \mid \dots \mid \text{Z}$

$\text{digit} \rightarrow 0 \mid \dots \mid 9$

4. 空白包括空格、回车以及水平制表符、垂直制表符。所有的空白在词法分析时，被当做单词ID，NUMBER以及保留字的分隔符，在词法分析之后，他们不被当做单词保留。
5. 注释是用花括号括起来的字符串 {...}，注释不能嵌套定义，但注释的定义可以跨行。

二. 语法分析程序实验

实验目的：通过扩展已有的样例语言TINY的语法分析程序，为扩展TINY语言TINY+构造语法分析程序，从而掌握语法分析程序的构造方法

实验内容：用EBNF描述TINY+的语法，用C语言扩展TINY的语法分析程序，构造TINY+的递归下降语法分析器

实验要求：将TOKEN序列转换成语法分析树，并能检查一定的语法错误

以下是TINY+语法的EBNF定义

1. $\text{program} \rightarrow \text{declarations stmt-sequence}$
2. $\text{declarations} \rightarrow \text{decl} ; \text{declarations} \mid \epsilon$
3. $\text{decl} \rightarrow \text{type-specifier varlist}$
4. $\text{type-specifier} \rightarrow \text{int} \mid \text{bool} \mid \text{string}$
5. $\text{varlist} \rightarrow \text{identifier} (, \text{identifier})^*$
6. $\text{stmt-sequence} \rightarrow \text{statement} (; \text{statement})^*$
7. $\text{statement} \rightarrow \text{if-stmt} \mid \text{repeat-stmt} \mid \text{assign-stmt} \mid \text{read-stmt} \mid \text{write-stmt} \mid \text{while-stmt}$
8. $\text{while-stmt} \rightarrow \text{while or-exp then do stmt-sequence end}$
9. $\text{if-stmt} \rightarrow \text{if or-exp then stmt-sequence (else stmt-sequence)} \mid \epsilon \text{ end}$

10. $\text{repeat-stmt} \rightarrow \text{repeat stmt-sequence until or-exp}$
 11. $\text{assign-stmt} \rightarrow \text{identifier} := \text{or-exp}$
 12. $\text{read-stmt} \rightarrow \text{read identifier}$
 13. $\text{write-stmt} \rightarrow \text{write or-exp}$
 14. $\text{or-exp} \rightarrow \text{and-exp} \mid (\text{or or-exp}) \mid \epsilon$
 15. $\text{and-exp} \rightarrow \text{comparison-exp} \mid (\text{and and-exp}) \mid \epsilon$
 16. $\text{comparison-exp} \rightarrow \text{add-sub-exp} \mid (> \mid < \mid >= \mid <= \text{and-sub-exp}) \mid \epsilon$
 17. $\text{add-sub-exp} \rightarrow \text{mul-div-exp} \mid (+ \mid - \text{add-sub-exp}) \mid \epsilon$
 18. $\text{mul-div-exp} \rightarrow \text{factor} \mid (* \mid / \text{mul-div-exp}) \mid \epsilon$
 19. $\text{factor} \rightarrow \text{ID} \mid \text{NUMBER} \mid \text{STRING} \mid \text{true} \mid \text{false} \mid \text{ch}('') \text{ or-exp ch}('') \mid \text{not factor}$
-

三. 词法分析程序实验

实验目的：构造TINY+的语义分析程序并生成中间代码

实验内容：构造符号表，用C语言扩展TINY的语义分析程序，构造TINY+的语义分析器，构造TINY+的中间代码生成器

实验要求：能检查一定的语义错误，将TINY+程序转换成三地址中间代码