第二章：词法分析

词法分析的基本功能

正则表达式

1. 词法分析的基本功能

- **词法分析程序**是编译程序的一部分，是整个编译过程的第一步工作。
- **词法分析器**读取源程序的字符序列，逐个拼出单词并构造相应的内部表示。同时检查源程序中的词法错误。它的核心作用即为将字符序列转化为计算机内部表示。

1. 词法分析的基本功能

- 单词： 是指语言中具有独立含义的最小的语义单位。
- 单词不是程序设计语言中的语法概念，是编译程序中引进的一个概念。

```
if (position > 10)  rate = 3.14 * initial;
```

例如3.14*initial就可以划分成3.14，*，initial这三个单词。
但是3.14不可以继续划分成3, ., 14

1.1 抽取单词序列的例子

例 某程序片段如下：

```
VAR  sum, first, count: real;
BEGIN
sum:=first + count * 10
END.
```

● **源程序**一般表现为字符序列的形式；

V	A	R	□	s	u	m	,	f	i	r	s	t
,	c	o	u	n	t	:	r	e	a	l	;	↵
B	E	G	I	N	↵	s	u	m	:	=	f	i
r	s	t	+	c	o	u	n	t	*	1	0	↵
E	N	D	.	↵	EOF							

● **期望的源程序表示形式**

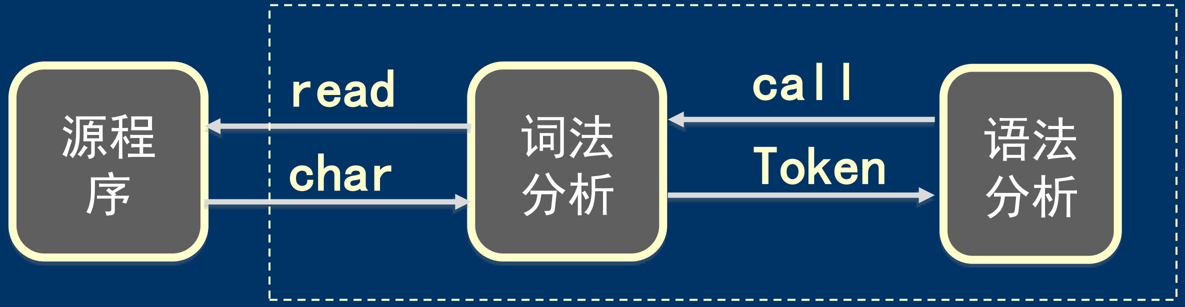
例 某程序片段如下：

```
VAR  sum, first, count: real;
BEGIN
sum:=first + count * 10
END.
```

VAR	sum	,	first	,	count	:
real	;		*BEGIN*		sum	:=
first	+	count	*	10		END
.						

1.2词法分析器的接口

- 词法分析器有两类，一类是仅作为语法分析的子程序：



- 另一类是作为编译器的独立一遍处理器：

1.3 单词类型的划分

常用程序设计语言的单词可以分为以下几类：

- **标识符：**用来标识程序中各个对象的名称. 它们由用户定义，用来表示变量名、常量名、数组名和函数名等。
- **保留字：**保留字一般是由语言系统自身定义的，通常是由字母组成的字符串。如C语言中的int，if，for，do等等。这些字在语言中具有固定的意义，是编译程序识别各类语法成分的依据。

1.3 单词类型的划分

- **常量：**主要包括整数常数、实数常数、字符常量、字符串常量等。
- **特殊符号：**包括运算符、界限符和控制符（格式符）。
  - 运算符表示程序中算术运算、逻辑运算、字符运算、赋值运算的确定的字符或字符串。如各类语言通用的+、-、*、/、<、>=、<=等。
  - 界限符在语言中是作为语法上的分界符号使用的，如逗号、分号、单引号等。
  - 控制符主要用于控制语言的格式，如回车、空格等。

1.3 单词类型的划分

1. 4思考如何实现词法分析

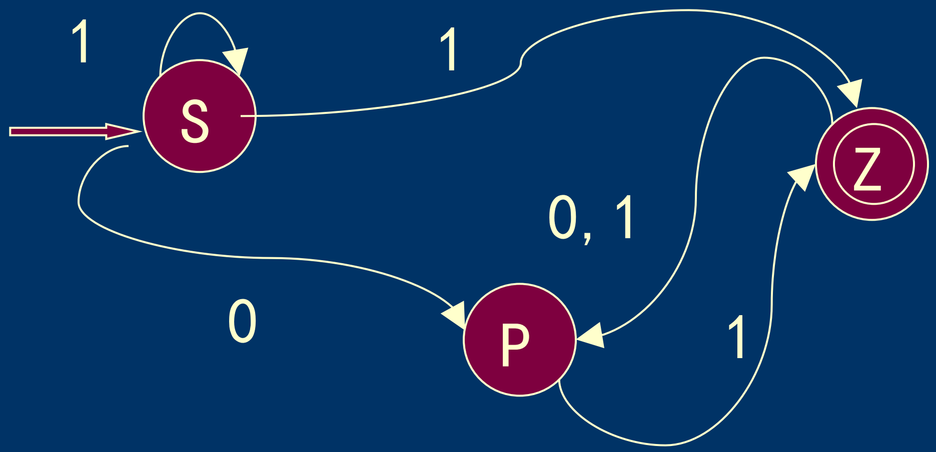
- 把问题分析清楚
- 采用合理的描述方式
- 设计算法

1.5 单词的描述工具

- 正则表达式

  ((y|z)*x(y|z)*x)*(y|z)*

- 自动机



2. 正则表达式

- 主要内容

a) 基本概念
b) 正则表达式
c) 正则表达式的性质
d) 如何基于正则表达式描述单词
e) 正则表达式的应用
f) 正则表达式的局限性

2.1 基本概念

1. 字母表（alphabet ）

字母表是元素的非空有穷集合，字母表中的一个元素称为该字母表的一个字母（letter），也可叫做符号（symbol）或者字符（character）。

注意：字母表具有非空性和有穷性。

- 字母表有时也称为符号表，通常用Σ表示。

例如： $\Sigma=\{a, b, c, d\}$

2.1 基本概念

2. 符号串

由字母表中的符号组成的任何有穷序列称为字母表上的符号串。一般用α，β，⋯.，x，y，z表示。

ε表示空串。对任一字母表Σ，都有ε是Σ上的符号串。

空串集{ε}不同于空集∅ 。

3. 符号串连接

设α和β 均是字母表Σ上的符号串， α和β的连接是把β的所有符号顺次地接在α的所有符号之后所得到的符号串。记为：αβ。

例如：设 α = abc ， β = de ，则α和β的连接：

αβ = abcde

2.1 基本概念

4. 符号串的方幂

设α是字母表Σ上的符号串，把α自身连接n次得到的符号串α'，称作符号串α的n 次幂，记作 $\alpha' = \alpha^n$。

$\alpha^0 = \varepsilon$

$\alpha^1 = \alpha$

$\alpha^2 = \alpha\alpha$

$\alpha^3 = \alpha^2\alpha = \alpha\alpha^2 = \alpha\alpha\alpha$

. . .

$\alpha^n = \alpha^{n-1}\alpha = \alpha\alpha^{n-1} = \alpha\alpha\cdots\alpha$ （n个α）

2.1 基本概念

5. 符号串集合

若集合A中的所有元素都是某字母表Σ上的符号串，则称A为该字母表上的符号串集合。

6. 符号串集的乘积

设A、B 是两个符号串集合，AB表示A与B的乘积，具体定义为：

$$AB = \{ xy \mid (x \in A) \wedge (y \in B) \}$$

例 设 $A = \{ a, bc \}$， $B = \{ de, f \}$，则：

$$AB = \{ ade, af, bcde, bcf \}$$

特别有：1、$\varnothing A = A\varnothing = \varnothing$ ，其中$\varnothing$表示空集。

2、$\{\varepsilon\}A = A\{\varepsilon\} = A$

2.1 基本概念

7. 符号串集合的方幂

设A为符号串的集合，则称 A^i 为符号串集 A 的方幂。具体定义如下：

$A^0 = \{ \varepsilon \}$

$A^1 = A$

$A^2 = AA$

……

$A^n = A^{n-1}A = AAA \cdots\cdots A$ （n个）

例： $A = \{ a, b \}$ 则：

$A^0 = \{ \varepsilon \}$

$A^1 = \{ a, b \}$

$A^2 = AA = \{ a, b \}\{ a, b \} = \{ aa, ab, ba, bb \}$

$A^3 = A^2A = \{ aa, ab, ba, bb \}\{ a, b \}$

$= \{ aaa, aab, aba, abb, baa, bab, bba, bbb \}$

……

$A^n = A^{n-1}A = AAA \cdots\cdots A$

2.1 基本概念

8. 符号串集合的正闭包

设A是符号串集合，则称A^+是符号串集合A的正闭包 $A^+=A^1 \cup A^2 \cup A^3 \cdots \cup A^n \cdots$

9. 符号串集合的星闭包

设A是符号串集合，则称A*是符号串集合A的星闭包 $A^*=A^0 \cup A^1 \cup A^2 \cup A^3 \cdots \cup A^n \cdots = A^0 \cup A^+$

2.1 基本概念

例:设A = {ab, cd}, 则 :

A^+ = {ab, cd, abab, abcd, cdab, cdcd, ababab, ababcd, … }

A* = {ε, ab, cd, abab, abcd, cdab, cdcd, ababab, ababcd, … }

2.2正则表达式

设Σ为有限字母表，在Σ上的正则表达式可递归定义如下：

（1）ε 和∅是Σ上的正则表达式；

（2）对任何a∈Σ，a 是Σ上的正则表达式；

（3）若r，s都是正则表达式，则(r)、r|s、r•s、r* 、r+也是正则表达式；

（4）有限次使用上述三条规则构成的表达式，称为Σ上的正则表达式.

2.2正则表达式

- 正则表达式的语义函数：给正则表达式赋予一种语义解释的函数。
- 不同的语义解释会使得正则表达式具有不同的语义，其操作结果也会不同。

例如，1+1这个表达式，不同语义解释所赋予表达式的含义和操作结果并不相同：

$$1+1=\begin{cases}2,\text{被解释为算术运算时} \\ 1,\text{被解释为逻辑运算时}\end{cases}$$

2.2正则表达式

单词的本质是字符串，在词法分析中，为了用正则表达式描述单词，我们用语义函数为正则表达式和字符串集合建立一种映射关系，使得正则表达式的语义解释被描述成字符串的形式。

在词法分析中，正则表达式e根据语义函数解释所得到的符号串集合称为正则表达式e的正则集。

2.2正则表达式

若设e、e1、e2为Σ上的正则表达式，则e所对应的正则集L(e)取值如下：

1. 当e=∅时，L(e)=∅;
2. 当e=ε时，L(e)={ε}；
3. 对于Σ中一个字符a，若e=a，则L(e)={a}；
4. 当e=e1·e2 时，L(e)=L(e1)L(e2)；
5. 当e=e1|e2时，L(e)=L(e1)∪L(e2)；
6. L((e))=L(e)
7. L(e*)=L(e)*;
8. L(e^+)=L(e)$^+$.

容易理解的定义形式

- 若用RE表示Σ上的正则表达式，L(RE)表示RE的正则集，且A、B都表示正则表达式，a表示字母表中的任意符号。有：

  1) $\varnothing \in RE$ $L(\varnothing)=\{\}$　　2) $\varepsilon \in RE$ $L(\varepsilon)=\{\varepsilon\}$

  3) $a \in RE$ $L(a)=\{a\}$　　4) $(A) \in RE$ $L((A))=L(A)$

  5) $A|B \in RE$ $L(A|B)=L(A) \cup L(B)$

  6) $A \cdot B \in RE$ $L(A \cdot B)=L(A)L(B)$

  7) $A^* \in RE$ $L(A^*)=L(A)^*$;

  8) $A^+ \in RE$ $L(A^+)=L(A)^+$;

正则表达式示例（1）

- Σ={ a, b }.

正则表达式e	L(e)
1. a	{a}
2. a|b	{a, b}
3. ab	{ ab }
4. (a|b) (a|b)	{aa, ab, ba, bb}
5. a*	{ ε , a, aa, aaaa, …}

正则表达式示例（2）

- ∑={ a, b }.

L(a(a|b)*)
=L(a) L((a|b)*)
=L(a) (L(a|b))*
={a}{a,b}*

正则表达式e	L(e)
1. ab*	∑上所有以a为首后跟任意多个（包括0个）b的字符串集
2. a(a|b)*	∑上所有以a为首的字符串集

2.3正则表达式的性质

正则表达式的性质

- $^{+}$=*>·>|　　运算优先级
- A | B = B | A　　| 的可交换性
- A |（B | C）=(A | B) C　　| 的可结合性

  A（B C）=(A B)C　　连接的可结合性
- A（B | C）=A B | A C　　连接的可分配性

  (A | B) C =A C | B C　　连接的可分配性
- A** =A*　　幂的等价性
- Aε=εA=A　　同一律

2.4正则表达式如何描述单词

- 标识符：L(L|D)*，其中L=A|B|…|a|b|…|z;D=0|1|…|9
- 常数
1. 整数：(+|-|ε)(D1D*)|0，其中 D1=1|2|…|9
2. 实数： (+|-|ε)(D1D*|0).D*
- 特殊符号：用枚举的方式来表示
1. 保留字：while|if|for|…
2. 运算符：+|-|*|…
3. 分界符：{|}|;|…
4. 控制符：\t|\0|…

2.5正则表达式的应用

- 手机中常用的号码地区识别软件
- 软件的安全监测方法
- 程序分析技术

2.6正则表达式的局限性

- 正则表达式不能用于描述配对或嵌套的结构
- 正则表达式不能用于描述重复串

例：{w c w | w是a和b的串}无法用正则表达式表示（保证两边w是相同的）.

例： $n \in AE$; $(AE) \in AE$; $AE+AE \in AE$

例子

- 设字母表Σ={0，1}，求二进制数字集合且为2的倍数。
  1. 所有Σ上定义的串的正则表达式为（1|0）*
  2. 则二进制数表示为1（1|0）*|0
  3. 其中能被二整除的表示为1（1|0）*0|0

作业

- 设字母表Σ={x，y，z}，
  1. 包含偶数个x的所有符号串。
  2. 不包含连续两个y的所有符号串集合。

((y|z)*x(y|z)*x)*(y|z)*

((x|z|yx|yz)*(y|ε)