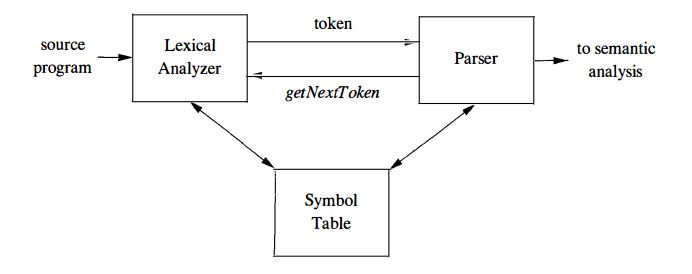
第三章 词法分析

(问1) 词法分析在编译器哪个地方?

(问2) 词法分析的作用是什么?



(1)

答: 词法分析是编译的第一阶段。

(2)

答: 词法分析器(Lexical Analyzer)读取了源程序(source code/program)，将其打碎成一个个的词法单元(token)之后传入语法分析器 (Parser)。

词法分析器的任务：

* 读取源程序，过滤掉源程序的注释和空白。
* 将编译器生成的错误信息与源程序的位置联系起来。
* 宏的扩展
* 生成**词法单元**

3.1.1 词法分析及语法分析

(问3) 为什么在编译过程中，将词法分析和语法分析分开?

3.1.2 三个概念: 词法单元、模式和词素

词法单元: 由一个词法单元名和一个可选的属性值组成.

模式: 描述了一个词法单元的词素可能具有的形式。

词素: 源程序中的一个字符序列。

|  |  |  |
| --- | --- | --- |
| 词法单元 | 模式(非形式描述) | 词素示例 |
| if | 字符 i，f | if |
| for | 字符 f，o，r | for |
| comparison | < 或 <= 或 = 或 … | <，<=，!=，… |
| id | 由字母开头的字母数字串 | pi，score，D2 |
| number | 任何数值常数 | 3.1，10，2.8E12 |
| literal | 引号“和”之间任意不含引号本身的字符串 | “core dumped” |

例(1) 分析下面C语句

print("Total=%d\n", score);

的词法单元。

(问4) 程序设计语言，一般需要定义哪些词法单元??

3.1.3 词法单元的属性

|  |  |  |
| --- | --- | --- |
| 词法单元 | 模式(非形式描述) | 词素示例 |
| for | 字符 f，o，r | for |
| id | 由字母开头的字母数字串 | pi，score，D2 |

例(2) Fortan语句

E = M \* C \*\* 2

中的词法单元名字和相关的属性值为?

|  |  |  |
| --- | --- | --- |
| 词法单元 | 模式(非形式描述) | 词素示例 |
| assign\_op | 字符 = | = |
| mult\_op | 字符\* | \* |
| exp\_op | 字符\*, \* | \*\* |
| number | 任何数字常量 | 1, 1.0 |
| id | 由字母开头的字母数字串 | pi，score，D2 |

例(3) 将下面的C++程序

float limitedSquare(x) {

float x;

/\* returns x-squared, but never more than 100 \*/

return (x<=-10.0 || x>=10.0) ? 100:x\*x;

}

划分成正确的词法单元序列, 并给出词法单元属性值（如果有的话）。

正则表达式: 用于形式描述语言，可以用来描述词法单元由什么样的词素构成的。

注意： 正则表达式也不是万能的，它也有很多模式是不能描述出来的。

3.3.1 串和语言 (ε)

字母表 (alphabet): 一个有限的符号集合 Σ

串 (string): 该字母表中符号的一个有穷序列

串的长度: |s|

语言 (language): 某个给定字母表上一个任意的可数的串的集合。

空集: Φ

仅包含空串的集合: {ε}

在语言理论中，术语“句子”常常被当作“串”的同义词。

串的运算:

* 连接: 假设x和y是串，则x和y的连接(concatenation)记作 xy。特别地εx=xε=x
* 幂运算（指数运算）: s^0 = ε, s^1 = s, s^n = s^{n-1}s

3.3.2 语言上的运算

语言运算:

* 并: L ∪ M = {s | s ∈或 s∈ M }
* 连接: LM = {st | s∈L 且 t∈M}
* 幂: L^n = L^{n-1}L, L^0 = {ε}
* 闭包: L\* = L^0 ∪ L^1 ∪ ….
* 正闭包: L+ = L^1 ∪ ….

问(4) ε一定不属于L+。 (判断题)

例(4) L={ A, B, …, Z, a, b, …, z }, D={ 0, 1, …, 9 } ，描述下面各项操作分别得到的新语言。

* L ∪ D：英文字母和数字的集合
* LD：由一个英文字母后跟随一个数字组成的串的集合
* L6：所有由6个英文字母组成的串的集合
* L\*：所有英文字母串（包括  ）的集合
* L(L ∪D )\* ：以英文字母开头的所有字母数字串的集合
* D+ ：不含空串的数字串的集合

3.3.3 正则表达式

|  |  |  |
| --- | --- | --- |
| 词法单元 | 模式(非形式描述) | 词素示例 |
| for | 字符 f，o，r | for |
| id | 由字母开头的字母数字串 | pi，score，D2 |

使用正则表达式来描述模式

正则表达式(正则式) 可以描述所有通过对某个字母表上的符号应用这些运算符而得到的语言。

例(5): C语言标识符的描述: 由字母或下划线开头，后接任意个字母、下划线或数字。

假设:

letter\_ 表示字母或下划线

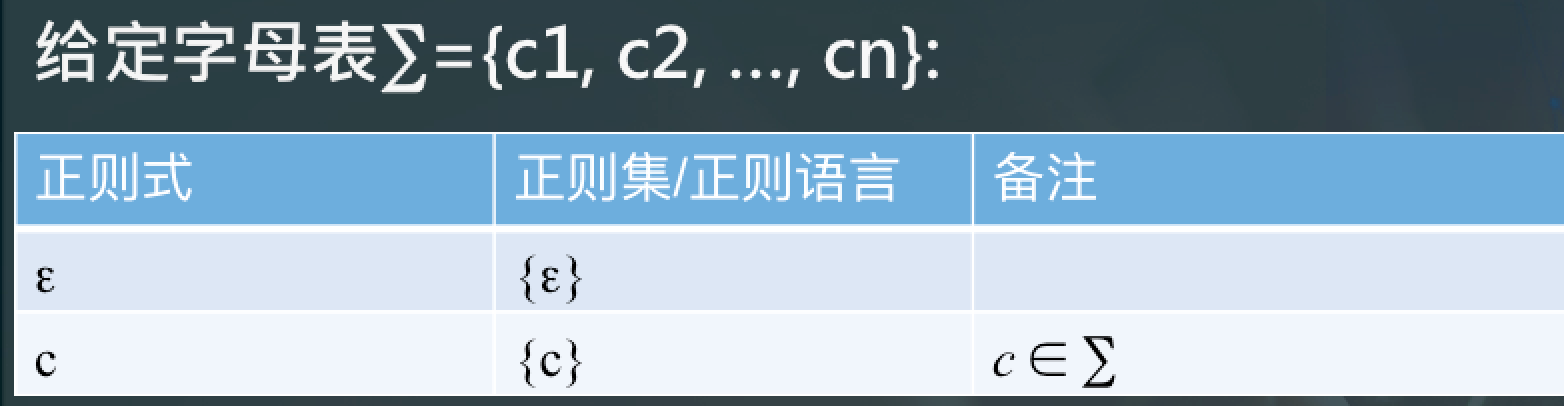
digit表示数字,

则, C语言标识符可以表示为:

正则表达式r

每个正规式r对应一个正则集或正则语言L(r)。

正则表达式 算术表达式





4种运算之间的优先级:

(a)|((b)\*(c)) ==> a|b\*c

例(6): 令Σ = {a, b},

(1) 正则表达式 a|b表示语言 ??

(2) 正则表达式 (a|b)(a|b)表示语言??

(3) 正则表达式a\*表示?

(4) 正则表达式(a|b)\*表示?

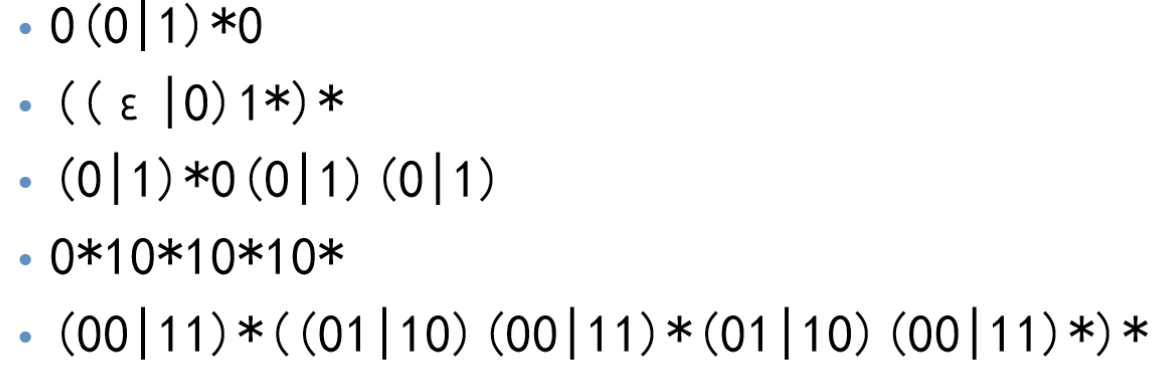
(5) 正则表达式 a|a\*b表示?

正则式等价

如果正则式r和正则式s表示的语言是相同的，那么说r和s是等价的，记为 r = s。

|  |  |
| --- | --- |
| 公理 | 描述 |
| r | s = s | r | | 是可交换的 |
| r | (s | t) = (r | s) | t | | 是可结合的 |
| (rs) t = r (st) | 连接是可结合的 |
| r (s | t) = rs | rt  (s | t) r = sr | tr | 连接对 | 是可分配的 |
| εr = r  rε = r | ε 是连接的恒等元素 |
| r\* = (r | ε) \* | 闭包中一定包含 ε |
| (r\*)\* = r \* | \* 是幂等的 |

例(7) 叙述由下列正则表达式描述的语言



以0开头和结尾的长度至少是2的01串.

所有的01串 (包含空串ε)

倒数第三位是0的01串

含有3个1的01串

含有偶数个0和偶数个1的所有01串 (包含空串ε)

例(8) 练习题3.3.2, 试描述下列正则表达式定义的语言:

1) a(a|b)\*a

2) ((ε |a)b\*)\*

3) (a|b)\*a(a|b)(a|b)

4) a\*ba\*ba\*ba\*

例(9) 假设C语言的标识符由字母或下划线开始，后接任意个字母、数字或下划线的串的集合。

该正则表达式可以写为?

3.3.4 正则定义 →Σ ε

例(10) （整数或浮点数）无符号数是形如5280、0.01234、6.336E4或1.89E-4的串。该正则表达式可以写为?

3.3.5 正则表达式的扩展

四种运算: 连接、或(选择)、闭包、括号

1) 一个或多个实例 : +

2) 零个或1个: ?

3)字符类: a|b|c ==> [abc] ==> [a-c]

4) {i, j}: 出现i到j次

5) .:除'\n'之外的任意字符

练习:

1) 包含5个元音的所有小写字母串, 这些串中的元音按顺序出现;

5个元音a, e, i, o, u

不含5个元音的任意字符：[b-df-hj-np-tv-z]，记为α

α\*(a)α\*(e)α\*(i)α\*(o)α\*(u)α\*

2) 所有由按词典递增序排列的小写字母组成的串 (包含空串).

a\*b\*…z\*

正则表达式扩充

<https://www.runoob.com/regexp/regexp-metachar.html>

<https://regex101.com/>

元字符

|  |  |
| --- | --- |
| 代码 | 说明 |
| . | 匹配除换行符以外的任意字符 |
| \w | 匹配字母或数字或下划线或汉字 |
| \s | 匹配任意的空白符 |
| \d | 匹配数字 |
| \b | 匹配单词的开始或结束 |
| ^ | 匹配字符串的开始 |
| $ | 匹配字符串的结束 |

出现次数

|  |  |
| --- | --- |
| 代码/语法 | 说明 |
| \* | 重复零次或更多次 |
| + | 重复一次或更多次 |
| ? | 重复零次或一次 |
| {n} | 重复n次 |
| {n,} | 重复n次或更多次 |
| {n,m} | 重复n到m次 |

懒惰限定符

|  |  |
| --- | --- |
| 代码/语法 | 说明 |
| \*? | 重复任意次，但尽可能少重复 |
| +? | 重复1次或更多次，但尽可能少重复 |
| ?? | 重复0次或1次，但尽可能少重复 |
| {n,m}? | 重复n到m次，但尽可能少重复 |
| {n,}? | 重复n次以上，但尽可能少重复 |

更多见 <https://www.runoob.com/regexp/regexp-metachar.html>

<https://regexone.com/>

问(5) 如何获取连续重复出现2次的字符串? 如aa, bb等等

练习:

长度为8-10的用户密码（以字母开头、数字、下划线）:

^[a-zA-Z]\w{7,10}$

电子邮箱验证：

^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$

URL地址验证：

^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$

简单的身份证号验证：

\d{15}|\d{18}$

提取并捕获html标签内容：

<a(?: [^>]\*)+href=([^ >]\*)(?: [^>]\*)\*>