编　译　方　法

第二章 词法分析
第一部分

冯速

feng_bnu@xinhuanet.com

北京师范大学

计算机科学系

本章学习内容

- 扫描处理的过程和相关概念

- 词法的形式描述（单词结构的表示）：
  正则表达式
- 单词（结构）的识别方法：
  有穷自动机（DFA）
- 词法分析的算法：扩展的DFA
- 识别、分析算法的实现：扫描程序
- TINY语言扫描器的实现过程
- C-Minus扫描器

2.1 扫描处理

- 扫描程序的任务：将源程序做为字符文件读入，将其分为若干个单词并产生相应的记号

  - 记号：源程序中的信息单元（逻辑单元），表示一类单词（命名）
  - 典型的记号种类： 关键字、标识符、特定符号：运算符、分隔符
  - 每一种又可细分为若干个记号
- 问题2.1： 什么是记号？举例说明典型的记号种类

TINY语言的记号

保留字		特殊符号		其它	
记号	串值	记号	串值	记号	串值
IF	if	PLUS	+	NUM	数
THEN	then	MINUS	–	ID	标识符
ELSE	else	TIMES	*		
END	end	OVER	/		
REPEAT	repeat	EQ	=		
UNTIL	until	LT	<		
READ	read	LPAREN	(		
WRITE	write	RPAREN	)		
		SEMI	;		
		ASSIGN	:=		

- 实践题2.1：列出C-Minus语言的记号

TINY记号的C语言说明

```
1  typedef enum
2  { /* book-keeping tokens */
3    ENDFILE, ERROR,
4    /* reserved words */
5    IF, THEN, ELSE, END, REPEAT,
6    UNTIL, READ, WRITE,
7    /* multicharacter tokens */
8    ID, NUM,
9    /* special symbols */
10   ASSIGN, EQ, LT, PLUS, MINUS,
11   TIMES,OVER,LPAREN,RPAREN,SEMI
12 }
```

- 实践题2.2：给出C-Minus记号的C语言说明

词法分析的主要工作

- 格式匹配
  - 概念
    - 记号
    - 单词
  - 过程
    - 格式说明
    - 识别算法

- 记号：格式的名字，表示单词的一个集合
- 单词：具体的字符串，匹配的内容
- 格式说明：正则表达式
- 识别算法：有穷自动机

记号、单词、属性

- 记号和单词是不同的概念
- 组成单词的字符串称为对应记号的串值或词义
- 任何与记号相关的值都是该记号的属性
- 串值是记号的一个属性
- 记号还有其它属性：ID的类型、NUM的值等
- 例（互动）：
  - 与字符串“1234”对应的记号是NUM
  - “1234”是 这个NUM的串属性
  - 1234是 它的值属性
  - 整型是 它的类型属性

记号和属性的用途

- 记号主要在词法分析和语法分析中使用
- 属性主要在语义分析中使用
- 问题2.2： 记号和单词有何不同？举例说明什么是属性？记号和属性各有什么用途？

记号和相关属性的存储结构

- 可以说明为结构体类型：

```
1  typedef struct
2    { TokenType tokenval;
3      char *stringval;
4      int numval;
5    } TokenRecord;
```

- 也可以说明为联合类型
- TINY编译器把记号currentToken作为扫描程序的返回值，而把串属性tokenString作为全局字符串变量
- 实践题2.3：考虑C-Minus编译器对记号和属性的处理方式

TINY扫描器的(主)函数说明

- 主函数：TokenType getToken(void);

  每次被调用时，(i)从输入中读取一个单词，(ii)返回该单词对应的记号，(iii)将记号的串属性置于全局变量tokenString中

2.2 正则表达式

- 在这里，把正则表达式作为词法的描述工具
- 例(表达式的含义，互动)：
  考虑整数集合$\mathcal{N}$和其上的运算$+, -, *$
  - $\mathcal{N}$的元素是数据对象：$1,2,3,\cdots$
  - $2+3$是式子，**不是数据对象**，**但表示一个数据**5 。同时，也表示一个集合$\{5\}$
  - $x+x$是式子（公式），不是数据对象，但对于任意的整数x，$x+x$表示一个整数
  - $x+x$是一个*描述特定集合(中的任意元素)结构的整数四则运算表达式*
  - $x/y(y \neq 0)$也是（整数集合上的）的表达式，但表示的不是整数的集合：
    构造新集合(新数据)

正则表达式的用途

- 正则表达式是描述、构造特定单词集合（正则语言）的结构的公式
- 正则表达式所表示的数据对象是某个字符集合Σ上的字符串集合
- 这样的字符集合Σ称为字母表
- 字母表Σ随处理对象的不同而不同

正则表达式的直观结构和含义

- 例：考虑在字母表$\Sigma = \{0, 1\}$上描述（构造）无符号二进制整数的集合

1 一位数：任意的一个一位二进制数是 $\Sigma = \{0, 1\}$ 的一个元素，反之， $\Sigma = \{0, 1\}$ 的任意元素都是一位二进制数，因此，一位二进制数的集合就是Σ，用公式Σ表示

2 两位数：由两个一位数粘合（连结）在一起构成，用$\Sigma \cdot \Sigma$表示

3 三位数：由一个两位数和一个一位数粘合在一起构成，用 $(\Sigma \cdot \Sigma) \cdot \Sigma$表示，简化成$\Sigma^3$

4 $n + 1$位数：由一个n位数和一个一位数粘合在一起构成： $\Sigma^{n+1} = \Sigma^n \cdot \Sigma$

正则表达式的直观结构和含义(续)

5 把所有这些一位数、两位数、· · · “掺合”到一起，就构成无符号二进制数的集合

$\Sigma|\Sigma^2|\Sigma^3\cdots$

简化表示成Σ^+

- 我们使用这样的“正则表达式”来表示我们要编译的程序设计语言的各类单词，即记号，以及所有单词的集合；表示这样的集合及集合中元素的结构

正则语言的直观含义

C语言的单词集合：需要表示的字符串集合

{ if, while,... ⟵ 保留字
  +, –, ++, ==,... ⟵ 运算符
  „ ;, /*, ... ⟵ 分隔符
  1, 2, 0.5, ..., ’G’, ’0’, "Good",... ⟵ 常量，文字量
  ab12, numval, func1,... ⟵ 标识符(变量名,函数名等)
}

单词集合中所含的字符有一定的范围：字母表 Σ

这样的单词集合是无穷集合，但是结构单纯，可以按一定的简单规则构造出来

正则集合，正则语言

正则语言的表示、描述 $\Longrightarrow$ 正则表达式

正则表达式的内涵和外延

正则表达式 { 表示正则语言的结构
是字符串的格式

- 本节学习正则表达式的定义、记法、扩展表示和使用
- 问题2.3： 说明正则表达式的内涵和外延

2.2.1 正则表达式的定义

正则表达式 $\Longleftarrow$ 正则语言 $\Longleftarrow$ 集合运算

- 集合的若干运算
  - 集合的并运算
  - 集合的连结运算： 设A和B为集合，
    则A与B的连结 $A\cdot B=\{w\cdot v | w\in A, v\in B\}$

    $\underbrace{A\cdots A}_{n}$:n次连结(n次幂) $\Longrightarrow A^n$
  - 集合的闭包运算： 集合A的闭包
    $A^*=\{w_1\cdot w_2\cdots w_n \mid n\geq 0, w_i\in A$ $(i=1,\cdots,n)\}$， 即，
    $A^*=\{\epsilon\}\cup A\cup A\cdot A\cup A\cdot A\cdot A\cup\cdots$

    $=A^0\cup A\cup A^2\cup A^3\cup\cdots$

集合运算示例（互动）

- 设A={a,b}，B={ab,ac}，则
  A·B={aab,aac,bab,bac}
  B·A={aba,abb,aca,acb}
  注意：A·B≠B·A
  A^0 ={ϵ}=B^0
  A^2 ={aa,ab,ba,bb}
  B^2 ={abab,abac,acab,acac}
  A*={ϵ,a,b,aa,ab,ba,bb,
  aaa,aab,aba,abb,baa,bba,bbb,· · · }
  B*={ϵ,ab,ac,abab,abac,acab,acac,· · · }
  注意：若A不为空集和{ϵ}，则A*是（可数）无穷集
  {ϵ}*={ϵ}， ϕ*={ϵ}
  注意：所有集合的闭包都包含ϵ，不能是空集

正则语言的定义

- 定义：字母表Σ上的正则语言递归定义如下：
  1. (基本正则语言：)
     (a) 对于任意的$a \in \Sigma$，单集$\{a\}$是正则语言；
     (b) 由长度为0的字符串，既空串ϵ组成的集合$\{\epsilon\}$是正则语言；
     (c) 空集ϕ是正则语言；
  2. (运算：) 若A和B是正则语言，则
     (a) （并）$A \cup B$是正则语言；
     (b) （连结）$A \cdot B$是正则语言；
     (c) （闭包）A^*是正则语言；
  3. 没有其它正则语言。
- 问题2.4： 给出正则语言的定义

正则语言示例（互动）

- 设Σ是计算机能表示的字符集，则以下集合都是正则语言

  If={if}={i}·{f}

  rev={if, then,else,end,repeat,until,read,write}

  ={if}∪{then}∪···

  sym={+,–,*,/,=,<,(,),;,:=}

  letter={a,b,···,z,A,···,Z}

  Id={ω|ω是TINY标识符} =letter·letter*

  dig={0,1,2,···,9}

  dig1={1,2,···,9}

  num={0,1,···,9,00,01,02,···} =dig·dig*

  num1={0,1,···,9,10,11,···} ={0}∪dig1·dig*

  Token=rev∪sym∪Id∪num

- 注意：正则集合是集合，本身没有结构
- 实践题2.6：考虑TINY和C-Minus的各单词集

正则表达式及其所表示的集合

定义：字母表Σ上的正则表达式 r及其所表示的集合 L(r)递归定义如下：

1. (基本表达式)
   (a) 对于任意的$a\in\Sigma$,符号 a是正则表达式,L(a)={a}；
   (b) 符号 ϵ 是正则表达式， $L(\epsilon)=\{\epsilon\}$；
   (c) 符号 ϕ 是正则表达式， $L(\phi)=\{\ \}=\phi$；
2. (运算) 设r和s是正则表达式，则
   (a) (选择) $r|s$(r或s)是正则表达式,$L(r|s)=L(r)\cup L(s)$；
   (b) (连结)rs(r与s的连结) 是正则表达式,L(rs)=L(r)·L(s)
   (c) (闭包) r* (r的闭包)是正则表达式， L(r*)= L(r)*；
   (d) (括号) (r) (括号r)是正则表达式， L((r))= L(r)；
3. 只有以上定义的是正则表达式。

问题2.5： 给出正则表达式及其表示的集合的定义

正则表达式示例（互动）

设Σ是计算机能表示的字符集，则

符号i是正则表达式，L(i)={i}

符号f是正则表达式，L(f)={f}

式子if是正则表达式，L(if)=L(i)·L(f)={i}·{f}={if}

i*是正则表达式，L(i*)=L(i)*={ϵ,i,ii,iii,$\cdots$}

ie|f是正则表达式，L(**ie|f**)=L(ie)∪L(f)={ie,f}

i|ef是正则表达式，L(**i|ef**)=L(i)∪L(ef)={i,ef}

(i|e)f是正则表达式，L(**(i|e)f**)=L(**i|e**)· L(f)

=(L(i)∪L(e))·L(f)={if,ef}

优先度： 选择运算<连结运算<闭包运算

括号的作用是改变运算的优先度

问题2.6： 举例说明括号和优先度对正则表达式及其所表示的集合的影响

正则表达式示例（续）及命名

- 表示整数的表达式是
  $(0|1|2|\cdots|9)(0|1|\cdots|9)*$
- 命名
  $digit = 0|1|\cdots|9$
  $nat = digit\ digit*$
- 注：在一个名字的定义中，不能直接或间接地使用该名字
- 例(续)：
  $letter = a|b|\cdots|z|A|B|\cdots|Z$
  $ID = letter\ letter* = letter^+$ (正则闭包)

教科书中的例子

- 对于Σ ={a,b,c}，考虑正好包括一个b的所有串的集合B及其表示
  设w是一个这样的串，则w中正好包含一个b
  w可以表示成
  $s_1 s_2 \cdots s_m$|b|$t_1 \cdots t_n$
  其中，$s_1, \cdots, s_m$及$t_1, \cdots, t_n$或者是a或者是c
  B=
  $\{s_1 s_2 \cdots s_m | m \geq 0, s_j \in \{a,c\} (j = 1, \cdots, m)\}$
  $\cdot\{b\} \cdot \{t_1 \cdots t_n | n \geq 0, t_j \in \{a,c\} (j = 1, \cdots, n)\}$
  $\therefore$ 令r= (a|c)*b(a|c)*， 则 L(r)=B

教科书中的例子（续）及等价性

- 对于Σ ={a,b,c}，考虑最多包括一个b的所有串的集合C及其表示
  C中元素或者不包含b，或者包含一个b，
  既，C=D∪B，其中D为由a和c组成的串的集合
  ∵ D可由(a|c)*表示, B可由(a|c)* b(a|c)*表示
  ∴ C可由(a|c)*|(a|c)*b(a|c)*表示
  同样C也可由(a|c)*(ϵ|b(a|c)*) 表示
  $\Rightarrow$ (a|c)*|(a|c)*b(a|c)* 与 (a|c)*(ϵ|b(a|c)*) 等价
- 定义：若r和s表示同一个集合，即L(r)=L(s)，则称r和s 等价，记为r=s
  例如：$r|s = s|r, (r|s)|t = r|(s|t)$
- 习题：2.1，2.4，2.5（2.4,2.5需要证明）
- 不是所有的集合都能用正则表达式表示：括号匹配问题

编写正则表达式的基本手法

- 掌握字符串集与表示该集合的正则表达式的*关联*
  $\Longleftarrow$ 研究对象
- 观察所给字符串集中字符串的(一般)结构，并对字符串作适当的划分 $\Longleftarrow$ 分解
  必要时
  1. 把字符串集A按串的结构分成若干个子集$B, C, D, \cdots$
  2. 对各子集编写正则表达式, $r_B, r_C, r_D, \cdots$
  3. 则表示所给字符串集A的正则表达式$r_A = r_B|r_C|r_D|\cdots$ $\Longleftarrow$ 综合
- 问题2.7： 叙述为正则语言编写正则表达式的基本手法

习题 2.1的意思

习题2.1： 为以下的字符串集编写正则表达式；若没有正则表达式，则说明原因：

a. 以a开头和结尾的所有小写字母串。

b. 以a开头或结尾的所有小写字母串。

c. 第1个不为0的所有数字串。

d. 所有表示偶数的数字串。

e. 每个2均在每个9之前的所有数字串，即，在任意的2之前都没有9，在每个9之后都没有2。

f. 所有由a和b组成的，不包含3个（或以上个）连续的b的字符串。

g. 所有由a和b组成的，包含单数个a或单数个b的字符串。

h. 所有由a和b组成的，包含偶数个a或偶数个b的字符串。

i. 所有由a和b组成的，a和b的数目相等的字符串。

TINY语言的记号

保留字		特殊符号		其它	
记号	串值	记号	串值	记号	串值
IF	if	PLUS	+	NUM	数
THEN	then	MINUS	–	ID	标识符
ELSE	else	TIMES	*		
END	end	OVER	/		
REPEAT	repeat	EQ	=		
UNTIL	until	LT	<		
READ	read	LPAREN	(		
WRITE	write	RPAREN	)		
		SEMI	;		
		ASSIGN	:=		

TINY语言记号的正则表达式

- 特殊符号：分成单字符记号和赋值符
  - **SINGLE** $= +|-|*|/|=|<|(|)|;$
  - **ASSIGN** $= :=$
- 数：只考虑无符号的整数
  - **NUM** $= digit\ digit*$
- 标识符：由字母组成的字符串
  - **ID** $= letter\ letter*$
- 保留字：作为特殊的标识符，通过查表判别
- 注释：由{ }括起的字符串，字符串中不能有}
  - **COMMENT** $= \{(\sim\})*\}$
- 这些正则表达式用选择运算符连接在一起，就构成TINY语言记号的正则表达式

C-Minus语言的正则表达式

- C-Minus语言的记号与TINY语言不同
- C-Minus语言的标识符与TINY语言不同
- 实践题2.7：给出C-Minus语言各记号的正则表达式和完整的正则表达式

2.2.2 正则表达式的扩展

缩写规定

(1) 一个或多个重复(正则闭包)

$\mathbf{r}^{+}$ =rr* $\Longrightarrow$ $digit^{+}$

(2) (字母表中的)任意字符

· L(·)= Σ

(3) 字符范围

对于字母表Σ中的元素s 和t，$[s-t]$ 表示Σ中所有排在s和t之间的元素所成的集合

$[s-t\,s_1-t_1]=[s-t]|[s_1-t_1]\Longrightarrow[a-zA-Z]$

(4) 不在给定集合中的任意字符

若r表示字母表Σ的一个子集，则~ r(非r)表示

$\Sigma-L(r)\Longrightarrow\{(\sim\})$*}

(5)可选的子表达式

对于表达式r，r? = r|ϵ

程序设计语言记号的正则表达式

- 讨论表示记号的标准种类的正则表达式与记号识别的问题

1) 数: 数字，自然数，十进制数，指数表示的数

$digit =$[0–9]

$nat = [0-9]^+$

$signedNat = (+|-)?nat$

$number =$

$signedNat(\text{“}\cdot\text{”}nat)?(\text{E}\, signedNat)?$

2) 保留字和标识符

保留字： $reserved =$ if|while|do|$\cdots$

字母： $letter =$[a-zA-Z]

标识符： $identifier = letter\,(letter|digit)$*

3）注释：？

二义性，空白格和先行

- **最长子串原理**：可组成单个记号的字符的最长串代表一个（单个的）记号：ifthen
- **优先选择性**：if
- **分隔符**： 不能属于该单词的一部分的字符
  1. **记号分隔符**： a[index]=4+2
     记号分隔符是下一个单词的一部分
     $\Longrightarrow$先行和回填
     单字符先行，多字符先行
  2. 空白格（包括注释）：$\cdots$ do/*注释*/it
     $Wspace = (newline|blank|tab|comment)^+$
     空白格不是下一个单词的一部分 $\Longrightarrow$舍弃
- 自由格式语言，固定格式语言
- 本节习题： 2.1, 2.4, 2.5，2.6