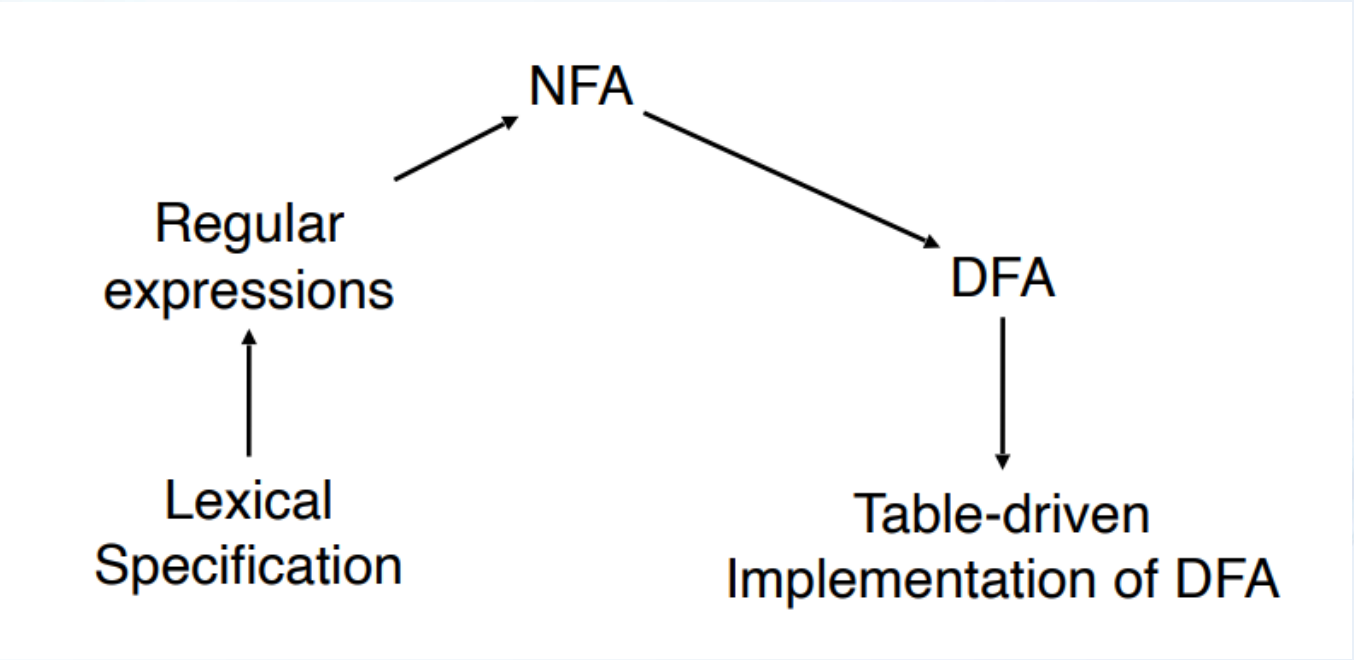
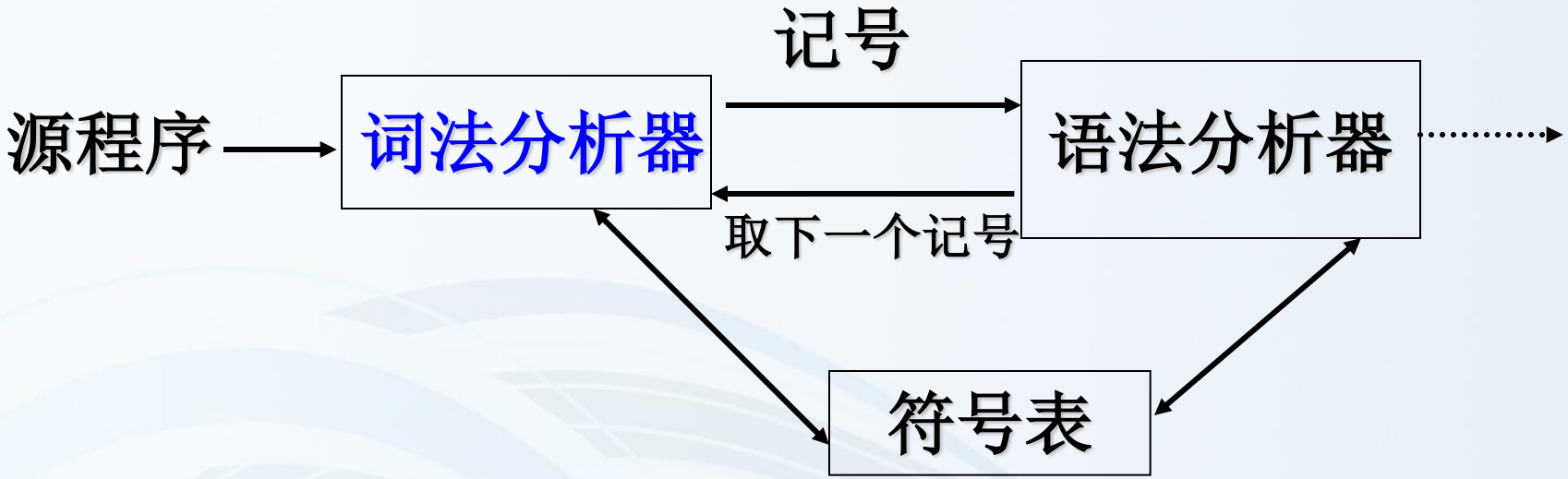
词法分析


NFA
Regular
expressions
DFA
Lexical
Specification
Table-driven
Implementation of DFA

3.1 词法分析器



词法分析器的功能

- 读入源程序
- 去除注释、空格、制表符、换行符等
- 建立符号表
- 识别词素，抽象成记号
- 将记号插入符号表
- 将记号提交给语法分析器
- 发现词法错误

单独划分词法分析阶段的原因

- 简化编译器的设计
- 提高编译器的效率
- 增强编译器的可移植性

词法分析器的三种构造策略

- 使用词法分析器生成器，从基于正规式的说明自动产生
- 使用传统的程序设计语言手工编写
- 使用汇编语言手工编写

词法错误

- 局部化

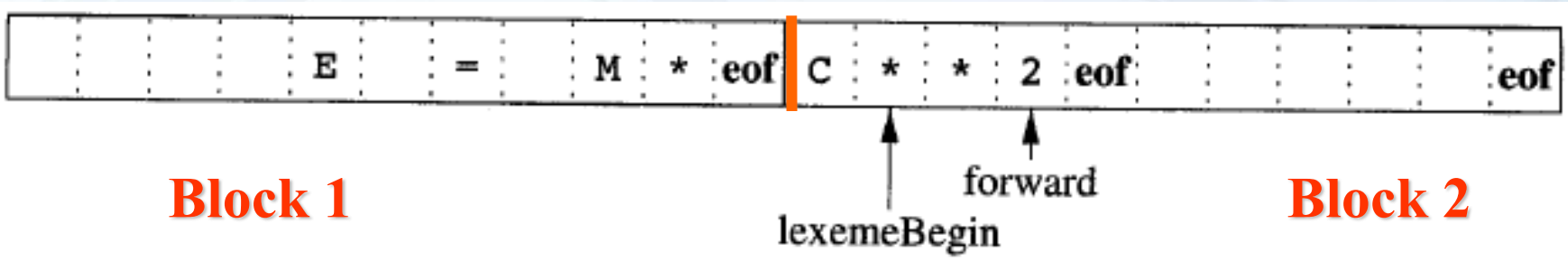
例：whil (x = 0) do　　并不产生词法错误

- 当前输入的前缀无法匹配任何记号时报词法错误，如：
  - 出现了当前语言中不会出现的字符，如 @ 或 #
  - 整数越界
  - 标识符名太长(有些语言最大长度为32个字符)
  - 字符串太长(有些语言最大长度为256个字符)
  - 字符串跨行

3.2 双缓冲双指针策略

- ❖采用大块存储空间作为缓存
- ❖一次将大批数据读入缓存
- ❖采用双缓冲
  - 在一个缓冲 I/O
  - 另一缓冲词法分析

当第一块完成后，
可继续读第二块



双缓冲双指针策略

```
switch (*forward++ ) {
   case eof:
         if (forward is at end of first buffer ) {
            reload second buffer;
            forward = beginning of second buffer;
         }
        else if (forward is at end of second buffer ) {
            reload first buffer;
            forward = beginning of first buffer;
            }
        else /* eof within a buffer marks the end of input */
            terminate lexical analysis;
        break;
    cases for the other characters: ......
}
```

3.3 词法单元

- 记号（词法单元、单词） *Token*
  - 具有独立含义的最小语法单位，每个记号代表一类符号串
  - 常见记号：保留字，标识符，常量，分界符或操作符(例如 *, ;)
- 模式 *Pattern*
  - 描述某个记号的词素集合的构造规则
- 词素 *Lexeme*
  - 源程序的符号序列，可匹配某个记号的模式
  - 标识符： x, count, name, …

记号-模式-词素（例）

非形式化描述不利于记号的自动识别

TOKEN	INFORMAL DESCRIPTION	SAMPLE LEXEMES
if	characters `i`, `f`	`if`
else	characters `e`, `l`, `s`, `e`	`else`
comparison	`<` or `>` or `<=` or `>=` or `==` or `!=`	`<=`, `!=`
id	letter followed by letters and digits	`pi`, `score`, `D2`
number	any numeric constant	`3.14159`, `0`, `6.02e23`
literal	anything but ", surrounded by "'s	`"core dumped"`

解决方法：形式化描述（正则式）

串的各部分(术语)

串S：banana

前缀*Prefix*:从尾部删除0或多个符号得到的串。ban, banana, ε

后缀*Suffix*:从首部删除0或多个符号得到的串。ana, banana, ε

子串*Substring*:删除某个前缀和某个后缀后得到的串。 nan, ban, ana, banana, ε

子序列*Subsequence*:删除0或多个符号所得的串。bnan, nn, ε

真前缀：不等于原串，也不为ε的前缀

真后缀、真子串：与上类似

试说明字符串abcdefghij，分别有多少个前缀、后缀、真前缀、子串和子序列。

解：串abcdefghij长度为10,分别从其尾部删除0,1,2,…10个符号，得不同前缀，故前缀数为11。

类似地，可知后缀数为11。

在前缀中去除长度为0和10两种情形，得真前缀，故真前缀数为11-2=9。

子串长度大于等于2时，有$C_{10}^{2}=45$种，长度等于1时，有$C_{10}^{1}=10$种，加上空串，共有56个子串。

子序列由该串任意字符组合确定，共有

$$C_{10}^{0}+C_{10}^{1}+C_{10}^{2}+\cdots+C_{10}^{10}=2^{10}=1024$$

语言上的运算

语言是符号串的集合，语言上的运算就是符号串集合上的运算

OPERATION	DEFINITION AND NOTATION
Union of L and M	$L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$
Concatenation of L and M	$LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$
Kleene closure of L	$L^* = \cup_{i=0}^{\infty} L^i$
Positive closure of L	$L^+ = \cup_{i=1}^{\infty} L^i$

正则表达式与语言（正则集）

- 正则表达式（又名正则式、正规表达式、正规式）：基于字母表构造串的一组规则（模板、模式）
  - 基于一个字母表，可以通过不同的正则式来定义不同的语言
- 如果r是正则式, 则L(r)表示r描述的语言（正规集、正则集）

还可以用什么来描述语言？

正则表达式及其语言

- 正则式定义在字母表Σ上：
  - 情形1：ε是正则式，L(ε)={ε}
  - 情形2：若a属于Σ, 则a是正则式，L(a)={a}
  - 若r和s为正则式，且L(r)和L(s)为对应语言．则
    - ✓(r)是正则式，表示语言 L(r)
    - ✓或(构造运算1): (r) | (s) 是正则式，表示语言 L(r) ∪ L(s)
    - ✓连接(构造运算2): (r)(s)是正则式，表示语言 L(r) L(s)
    - ✓闭包(构造运算3): (r)*是正则式，表示语言 (L(r))*

以上运算均左结合，优先级从高到低为：*， 连接，|

如何理解*ab|c*d*

正则表达式与语言(例)

由正则式生成的串的集合称为正则语言。语言中串的数量可能有限，也可能无限，它们就是词法分析时的词素。

正则式	语言
a	{a}
a|b	{a,b}
ab	{ab}
(a|b)(a|b)	{aa,ab,ba,bb}
a*	{ε,a,aa,…, 任意个a的串}
(a|b)*	{ε,a,b,aa,ab,ba,bb,…所有a,b组成的串}

$(a|b|\varepsilon)^+$表示什么？

正则式运算的代数规律

设r,s,t为正则式：

(1) r|s = s|r

(2) r|(s|t) = (r|s)|t

(3) (rs)t = r(st)

(4) r(s|t) = rs|rt

(s|t)r = sr|tr

(5) rε = εr = r

(6) r|r = r

描述以下正则式表示的语言：

- $a(a|b)^*a$

  定义于{a,b}之上的，以a开头和结尾的，长度不少于2的串。

- $a^*ba^*ba^*ba^*$

  定义于{a,b}之上的，含有3个“b”的串。

- $(a|b)^*a(a|b)(a|b)$

  定义在{a,b}之上的，倒数第3个字母为“a”的串。

- $((\varepsilon|a)b^*)^*$

  定义于{a,b}之上的任意串。

选择所描述语言与$(0|1)^*1(0|1)^*$相同的正规式

多选

(A) (01|11)*(0|1)*

(B) (0|1)*(10|11|1)(0|1)*

(C) (1|0)*1(1|0)*

(D) (0|1)*(0|1)(0|1)*

正则表达式的扩展

- “+”：1个或多个
  - $r^* = r^+ | \varepsilon$
  - $r^+ = r\ r^*$
- “?”：0或1个
  - $r? = r\ | \varepsilon$
- [range]：字符的范围

A | B | C | … | Z = [ABC…Z] = [A-Z]

id →[A-Za-z][A-Za-z0-9]*

正则表达式例子

- 以 “tab”开始，或以 “bat”结束的小写字母串

  tab(a|b|…|z)* | (a|b|…|z)* bat

  tab[a-z]* | [a-z]*bat

- 福大邮箱地址

  $[a\text{-}z0\text{-}9]^{+}$@ fzu.edu.cn

- 由a和b组成，且不含子序列abb的串

  b*a*b?a*

- 由a和b组成，且不含子串abb的串

  b*(a|ba)*b? 或 b*(a|ab)*

一些记号的正则式

记号	模式
if	if
then	then
else	else
relop	< | <= | > | >= | = | < >
id	letter (letter | digit)*
num	$digit^{+}(.digit^{+})?(E(+\|-)?digit^{+})?$