第2讲
词法分析-正则表达式

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年09月06日

两个问题！

- 什么是词法分析?
- 如何描述词法?

词法分析 (Lexical Analysis)

- **程序示例：**

```
if (i == j)
    printf("equal!");
else
    num5 = 1;
```

- **程序是以字符串的形式传递给编译器的**

\tif (i == j)\n\t\tprintf("equal!");\n\telse\n\t\tnum5 = 1;

- **目的：将输入字符串识别为有意义的子串**
  - ❖ 子串的种类（Name）
  - ❖ 可帮助解释和理解该子串的属性（Attribute）
  - ❖ 可描述具有相同特征的子串的模式（Pattern）

词法单元 token

词法单元 (Token)

- **由一个记号名和一个可选的属性值（可以为空）组成**
  - token := <token_name, attribute_value>
- **属性记录词法单元的附加属性**
  - 例：标识符id的属性包括词素、类型、第一次出现的位置等
  - 保存在符号表（Symbol table）中，以便编译的各个阶段取用

源程序
position = initial +
rate * 60

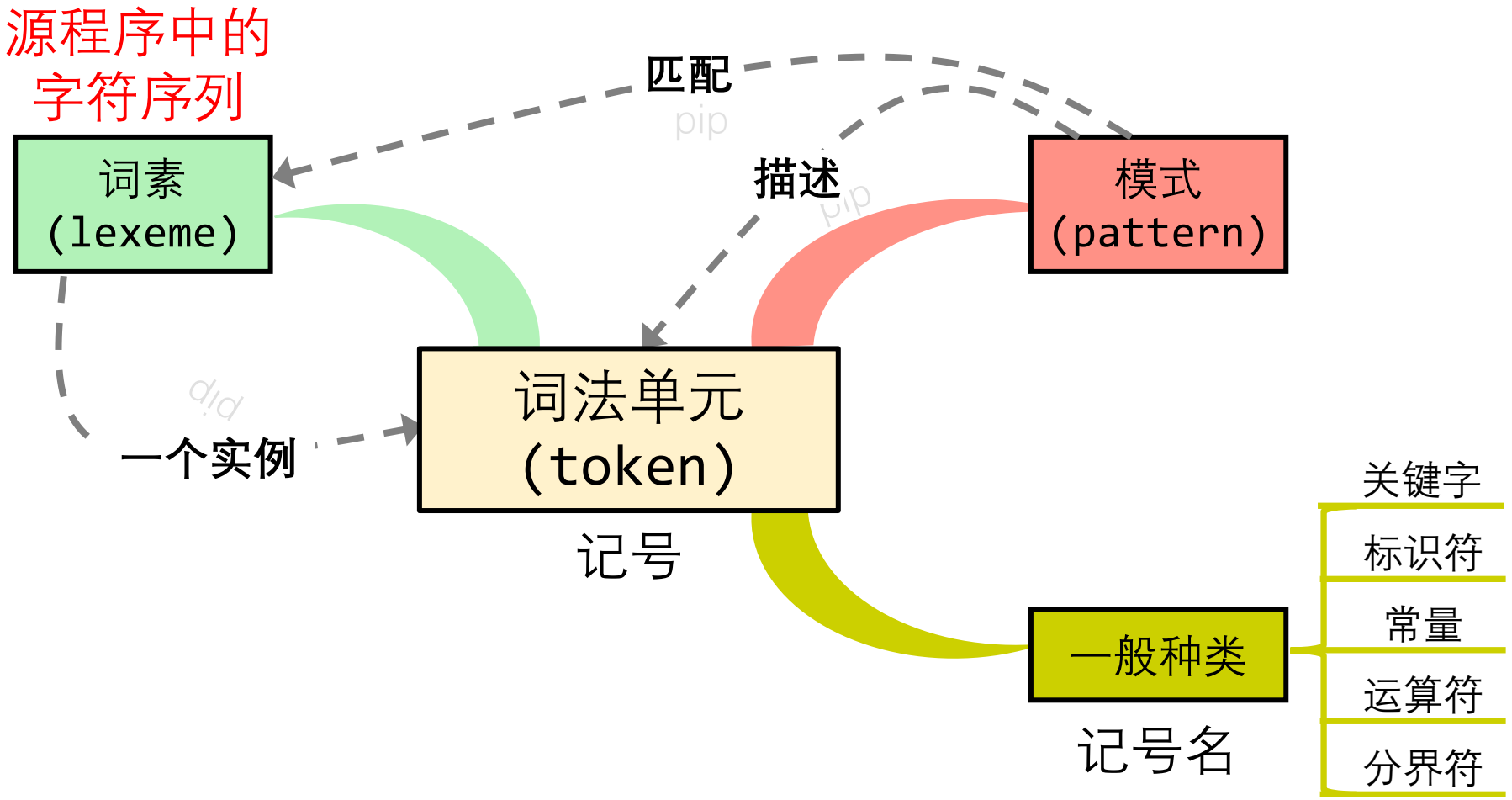
⟨id，指向符号表中position条目的指针⟩
⟨assign _ op⟩
⟨id，指向符号表中initial条目的指针⟩
⟨add_op⟩
⟨id，指向符号表中rate条目的指针⟩
⟨mul_ op⟩
⟨number，整数值60⟩

符 号 表

1	position	· · ·
2	initial	· · ·
3	rate	· · ·

词素（实例）

四个关键术语


源程序中的
字符序列
匹配
描述
词素
(lexeme)
模式
(pattern)
一个实例
词法单元
(token)
记号
一般种类
记号名
关键字
标识符
常量
运算符
分界符

词法单元(记号)、实例与模式

```
if (i == j) printf("equal!");
else num5 = 1;
```

记号名	实例（词素）	模式的非形式描述
if	if	字符i, f
else	else	字符e, l, s, e
relation	==, < , <= , …	== 或 < 或 <= 或 …
id	i, j, num5	由字母开头的字母数字串
number	1, 3.1, 10, 2.8 E12	任何数值常数
literal	“equal!”	引号“和”之间任意不含引号本身的字符串

两个问题！

- 什么是词法分析?
- 如何描述词法?

正整数的描述

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正整数的描述

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

字母表

可以从0-9中任选一个数字
| 表示选择运算符

$$digit \rightarrow 0|1|2|\cdots|9$$

$$digits \rightarrow digit\ digit^*$$

***是闭包运算，表示零次或多次出现**

由数字不断拼接形成（至少有一个数字）
两个元素顺序放置表示拼接操作

正整数的描述

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

digit → 0|1|2|…|9

digits → digit digit*

正则表达式
(Regular Expression)

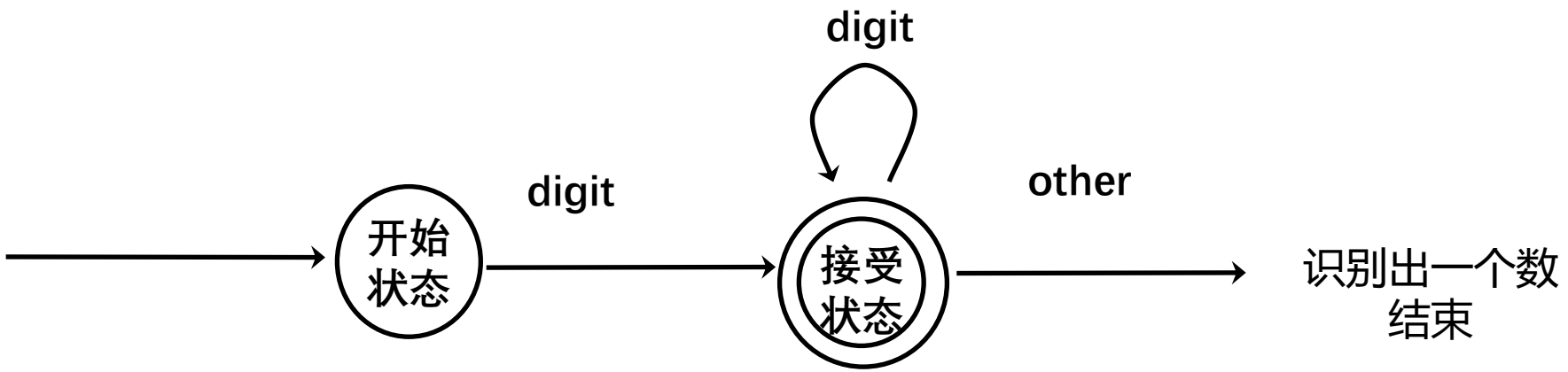
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



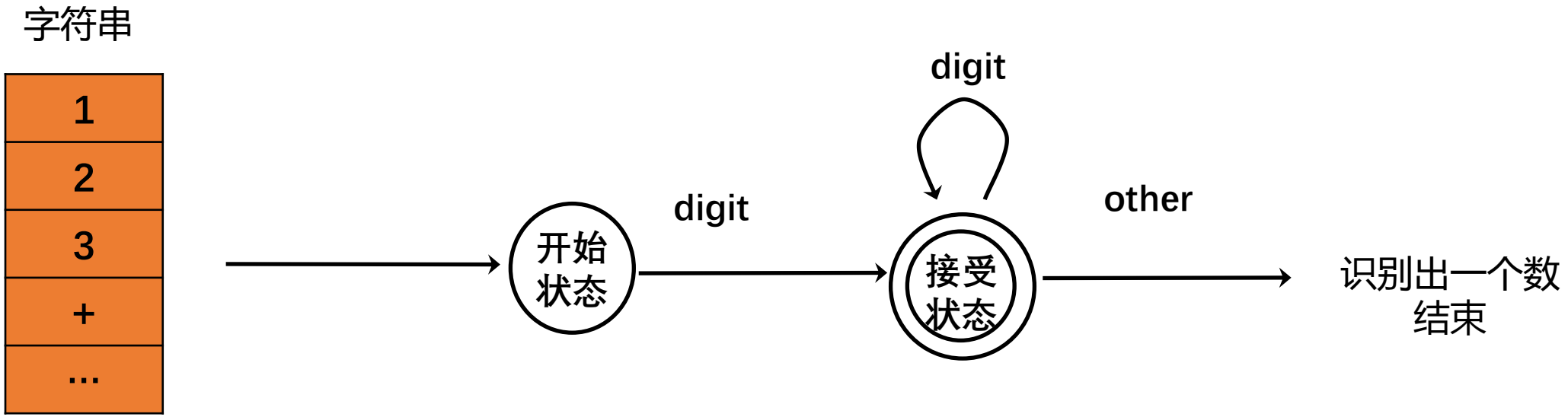
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



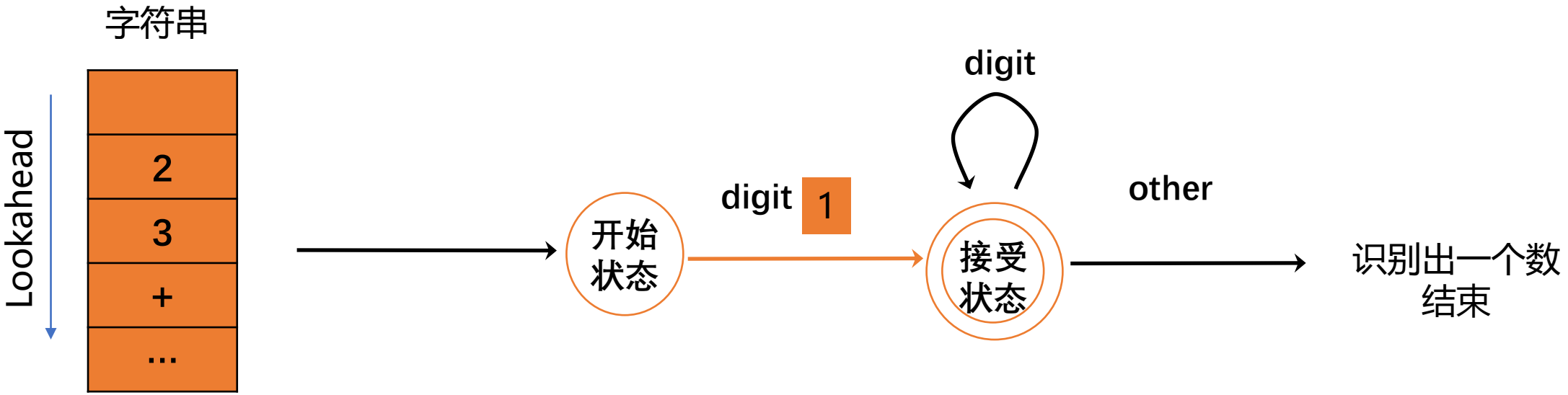
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



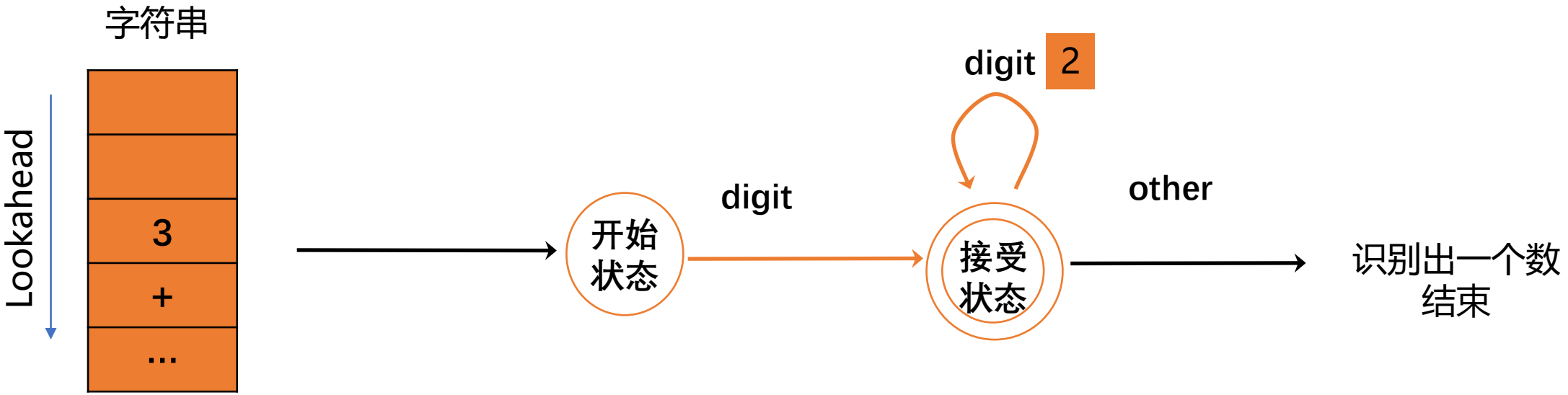
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



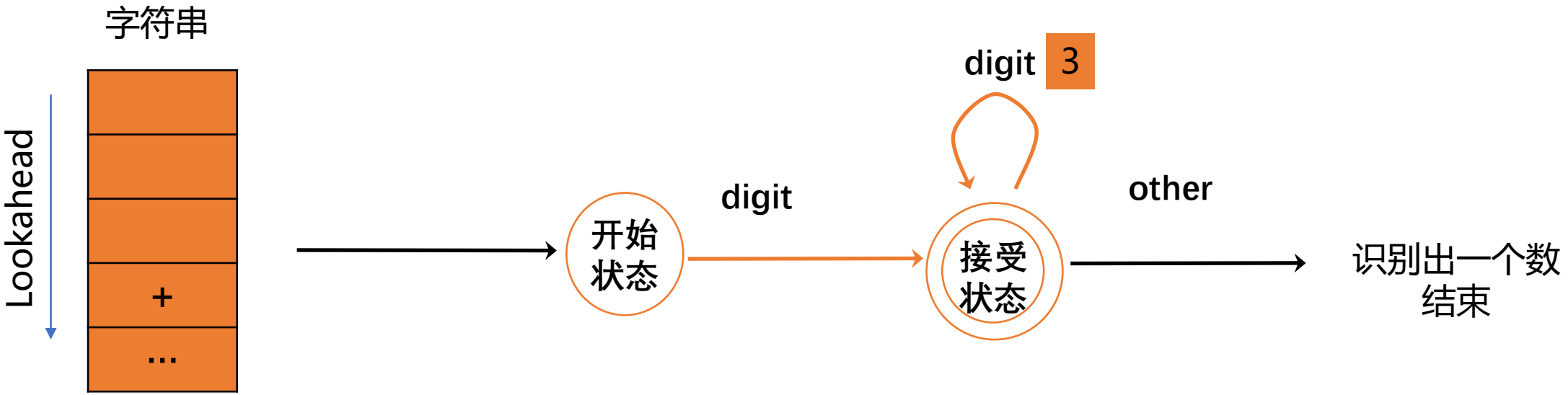
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



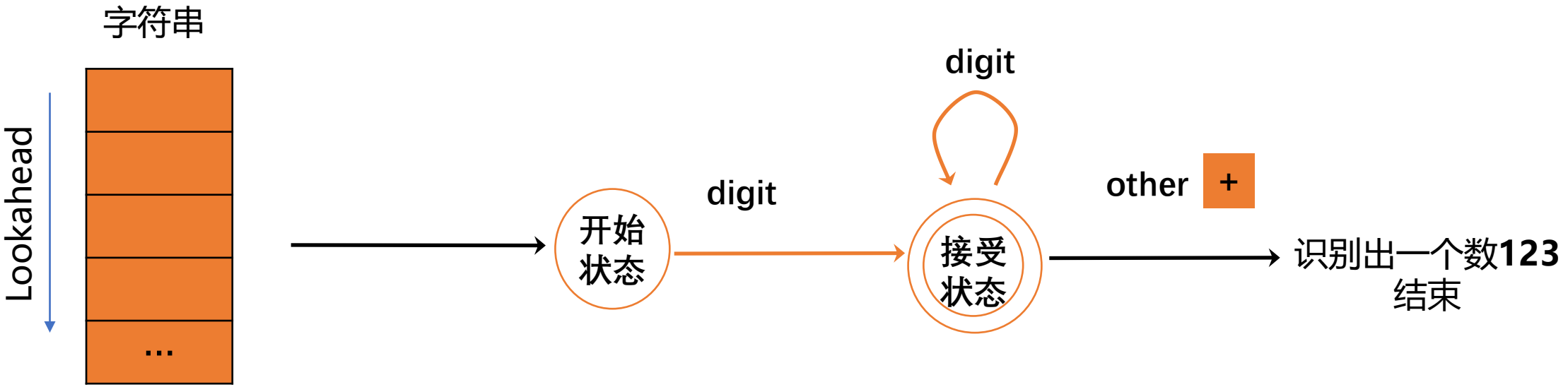
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



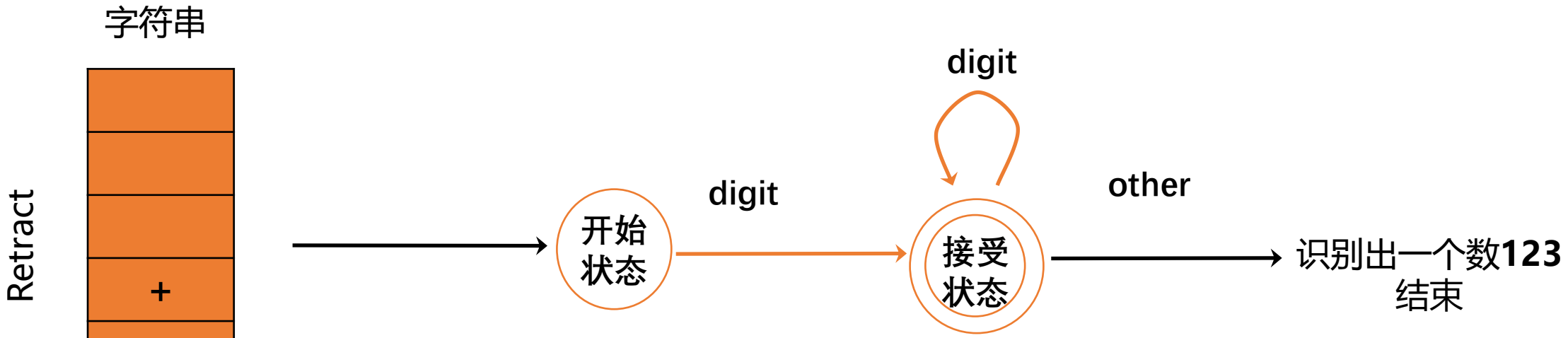
正整数的识别

- **正整数描述了一个集合**
  - 最基本的构成单元：0、1、2、3、…、9
  - 组合形式：10、123、1001、19461、…
    - 可以看做由基本单元不断拼接而形成的串

正则表达式

digit → 0|1|2|…|9

digits → digit digit*



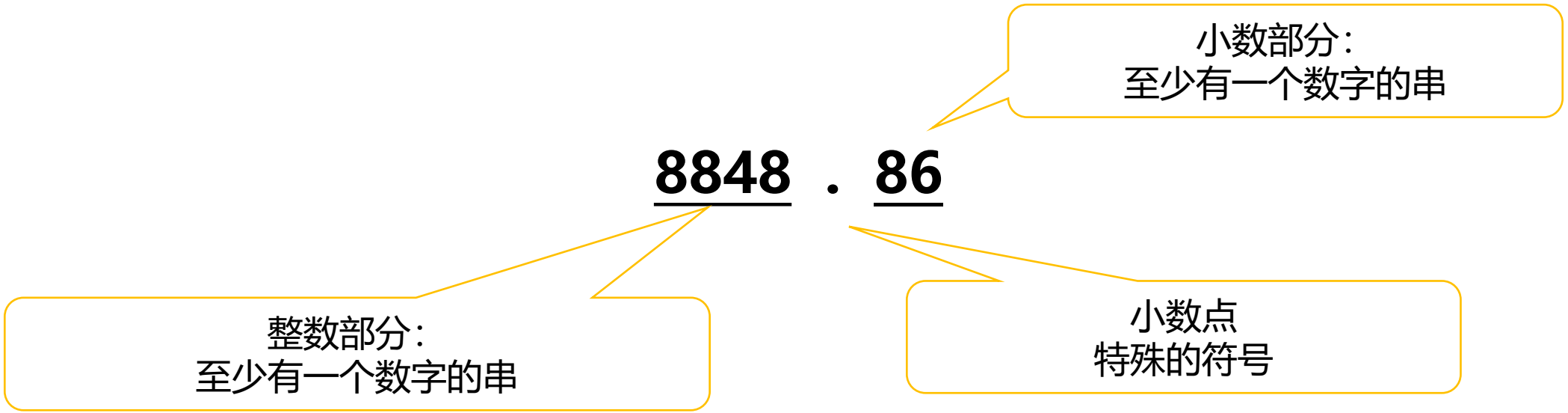
有限自动机
(Finite Automata)

带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86（2020年测定的珠穆朗玛峰高度）

带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86（2020年测定的珠穆朗玛峰高度）



带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86 （2020年测定的珠穆朗玛峰高度）

基本数字 digit → 0|1|2|…|9

整数部分 digits → digit digit*

小数部分 digits → digit digit*

带小数的数字串number→digit digit*.digit digit*

正则表达式
(Regular Expression)

带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86 （2020年测定的珠穆朗玛峰高度）

基本数字 digit → [0-9]

整数部分 digits → $digit^+$

小数部分 digits → $digit^+$

带小数的数字串 number → $digit^+$. $digit^+$

简写形式

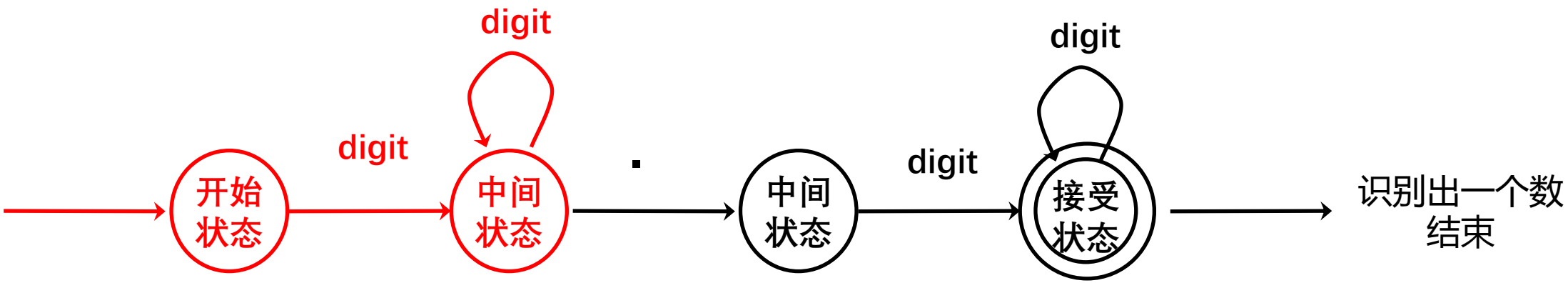
正则表达式
(Regular Expression)

带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86

正则表达式

number → digit⁺ . digit⁺

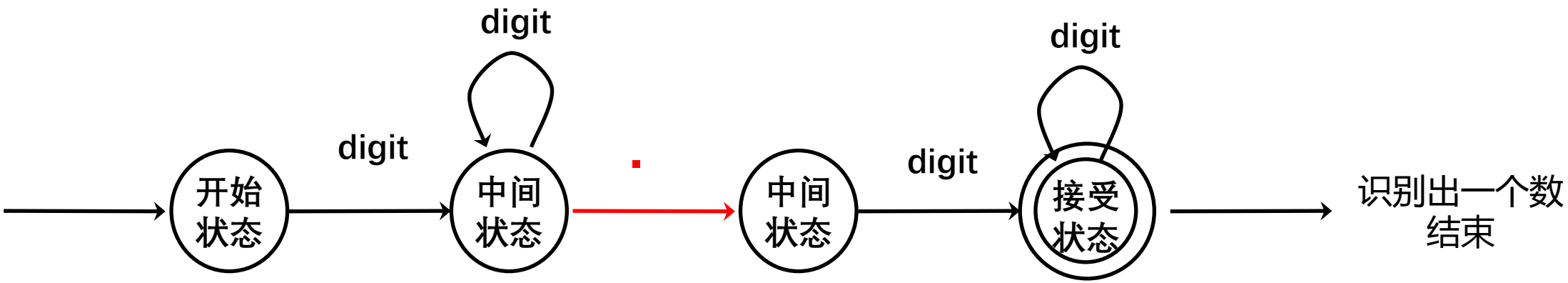


带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86

正则表达式

number → digit⁺ . digit⁺

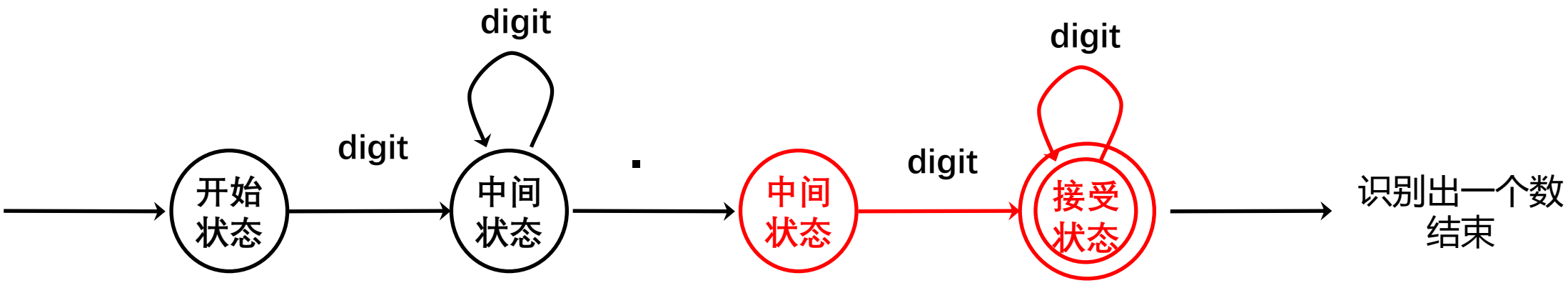


带小数的数如何识别?

- 1.5, 10.28, 237.8, 8848.86

正则表达式

number → digit$^+$. digit$^+$



串和语言

- **术语**
  - 字母表：符号的有限集合，例：$\Sigma = \{0, 1\}$
  - 串：符号的有穷序列，例：*0110*，ε
  - 语言：字母表上的一个串集

    $\{\varepsilon, 0, 00, 000, \ldots\}$，$\{\varepsilon\}$，$\varnothing$
  - 句子：属于语言的串

注意区别：
ε，$\{\varepsilon\}$，$\varnothing$

- **串的运算**
  - 连接（积）：　　xy，$s\varepsilon = \varepsilon s = s$
  - 指数（幂）：　　s^0为ε，s^i为$s^{i-1}s$（$i > 0$）

串和语言

- **语言的运算**
  - 并：$L \cup M = \{s \mid s \in L \text{ 或 } s \in M\}$
  - 连接：$LM = \{st \mid s \in L \text{ 且 } t \in M\}$
  - 幂：L^0是$\{\varepsilon\}$，L^i是$L^{i-1}L$
  - 闭包：$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$
  - 正闭包：$L^+ = L^1 \cup L^2 \cup \ldots$

优先级：
幂〉连接〉并

- **示例**

L: { $A, B, \ldots, Z, a, b, \ldots, z$ }, D: { 0, 1, …, 9 }

$L \cup D$, LD, L^6, L^*, $L(L \cup D)^*$, D^+

正则表达式（Regular Expr）

优先级：
闭包*〉连接〉选择 |

- ∑ = {a, b}
  - ❖ *a* | *b*　　{*a*, *b*}
  - ❖ (*a* | *b*) (*a* | *b*)　　{*aa*, *ab*, *ba*, *bb*}
  - ❖ *aa* | *ab* | *ba* | *bb*　　{*aa*, *ab*, *ba*, *bb*}
  - ❖ a^*　　由字母*a*构成的所有串集
  - ❖ $(a \mid b)^*$　　由*a*和*b*构成的所有串集
- **复杂的例子**

  (00 | 11 | ((01 | 10) (00 | 11) * (01 | 10))) *

  句子：01001101000010000010111001

正则表达式（Regular Expr）

- 正则式用来表示简单的语言

正则式	定义的语言	备注
ε	{ε}	
a	{*a*}	$a \in \Sigma$
(*r*)	*L*(*r*)	*r*是正则式
(*r*) | (*s*)	*L*(*r*)∪*L*(*s*)	*r*和s是正则式
(r)(s)	*L*(*r*)*L*(*s*)	*r*和s是正则式
$(r)^*$	$(L(r))^*$	*r*是正则式

((*a*) $(b)^*$)| (*c*)可以写成ab^*| *c*

优先级：
闭包*〉连接〉选择 |

正则定义的例子

- C语言的标识符是字母、数字和下划线组成的串

$$\text{letter\_} \rightarrow A \mid B \mid \cdots \mid Z \mid a \mid b \mid \cdots \mid z \mid \_$$

$$\text{digit} \rightarrow 0 \mid 1 \mid \cdots \mid 9$$

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

正则定义

- **bottom-up方法**
  - ❖ 对于比较复杂的语言，为了构造简洁的正则式，可先构造简单的正则式，再将这些正则式组合起来，形成一个与该语言匹配的正则序列。

$$d_1 \rightarrow r_1$$

$$d_2 \rightarrow r_2$$

$$\cdots$$

$$d_n \rightarrow r_n$$

  - ❖ 各个d_i的名字都不同，是新符号，not in $\sum$
  - ❖ 每个r_i都是$\sum \cup \{d_1, d_2, \ldots, d_{i-1}\}$上的正则式

正则定义的例子

- 无符号数集合，例1946,11.28,63E8,1.99E–6

正则定义的例子

- 无符号数集合，例1946,11.28,63E8,1.99E–6

digit → 0 | 1 | … | 9

digits → digit digit*

optional_fraction → . digits | ε

optional_exponent → (E (+ | – | ε) digits) | ε

number→digits optional_fraction optional_exponent

正则定义的例子

- 无符号数集合，例1946,11.28,63E8,1.99E–6

digit → 0 | 1 | … | 9 [0-9]

digits → digit digit*

optional_fraction → . digits | ε

optional_exponent → (E (+ | – | ε) digits) | ε

number→digits optional_fraction optional_exponent

- 简化表示

number → $digit^{+}$ (.$digit^{+}$)? (E[+–]? $digit^{+}$)?

注意区分：
？ 和 *

正则定义的例子

while → while

do → do

relop → < | < = | = | < > | > | > =

letter_ → [A-Za-z_]

id → letter_ (letter_ | digit)*

number → digit+ (.digit+)? (E[+–]? digit+)?

delim → blank | tab | newline

ws → delim+

问题：正则式是静态的定义，如何通过正则式动态识别输入串？

一起努力
打造国产基础软硬件体系！

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年09月06日