

《编译原理和技术》

词法分析简介



什么是词法分析?

```
1. main( ) {  
2.   int aaa, bb=10, c=10;  
3.   aaa=bb+c*10;  
4.   Printf("a 的值是%d", a);  
5. }
```



什么是词法分析?

字符串

i
n
t
a
a
a
,
b
b
=
...

词法分析器

```
1. main( ) {  
2.   int aaa, bb=10, c=10;  
3.   aaa=bb+c*10;  
4.   Printf("a 的值是%d", a);  
5. }
```

记号流

int
aaa
bb
=
10
c
=
10



什么是词法分析?

字符串

a
a
a
=
b
+
c
*
10

词法分析器

```
1. main( ) {  
2.   int aaa, bb=10, c=10;  
3.   aaa=bb+c*10;  
4.   Printf("a 的值是%d", a);  
5. }
```

记号流

aaa
=
bb
+
c
*
10



什么是词法分析?

字符串

P
r
i
n
f
(
"
a
的
值
是
...

词法分析器

```
1. main( ) {  
2.   int aaa, bb=10, c=10;  
3.   aaa=bb+c*10;  
4.   Printf("a 的值是%d", a);  
5. }
```

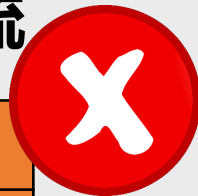
编译器的错误提示:


```
prog.c:4:1: error: "Printf" 函数没有定义  
    Printf( "a 的值是%d" , a);  
    ^
```

任务: 词法分析是程序编译的第一阶段, 将源代码中的字符拼接成为合法的单词, 输出单词的序列 (记号流)。


记号流

Printf





**问题：如何描述编程语言所允许的
合法输入？**





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

$\text{digit} \rightarrow 0|1|2|\cdots|9$

可以从0-9中任选一个数字
| 表示选择运算符

$\text{digits} \rightarrow \text{digit digit}^*$

*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）
两个元素并列放置表示拼接操作



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

$\text{digit} \rightarrow 0|1|2|\cdots|9$

$\text{digits} \rightarrow \text{digit digit}^*$

正则表达式
(Regular Expression)



正整数的识别

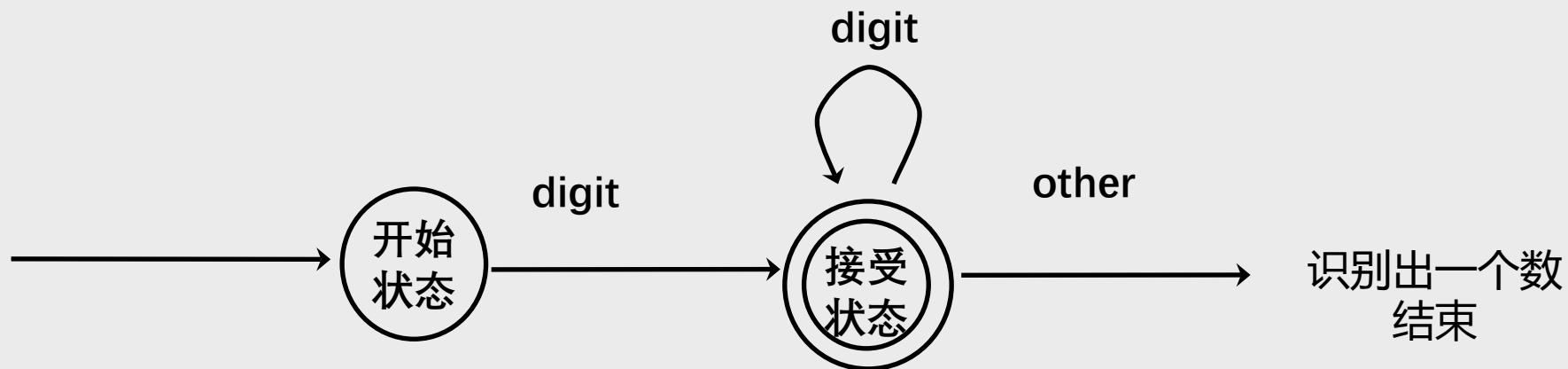
□ 正整数描述了一个集合

- 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 \rightarrow 0|1|2|...|9

digits \rightarrow 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

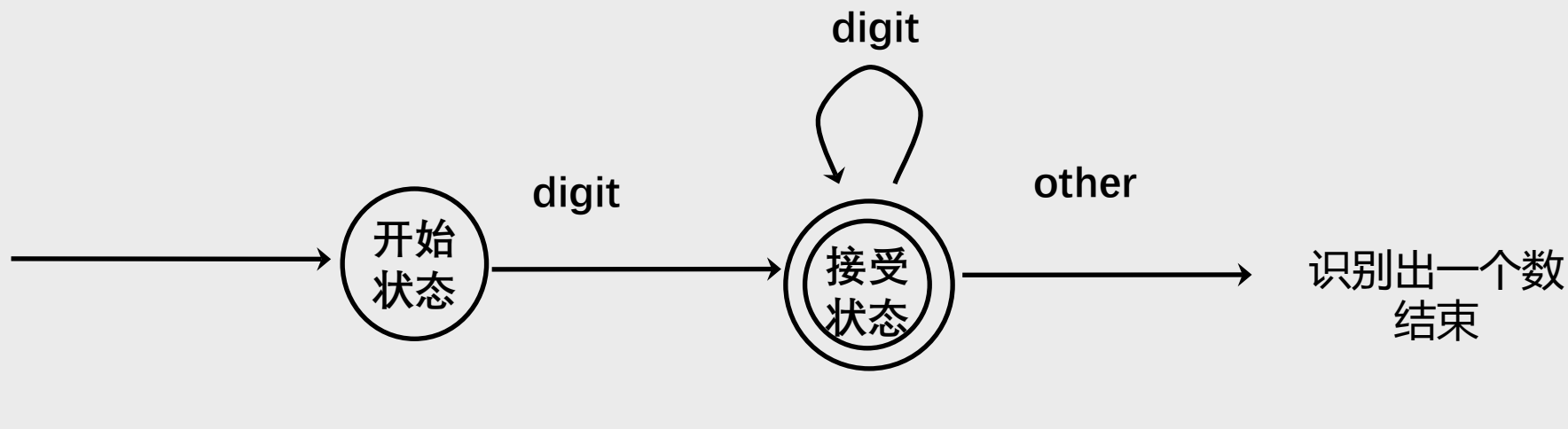
正则表达式

$\text{digit} \rightarrow 0|1|2|\cdots|9$

$\text{digits} \rightarrow \text{digit digit}^*$

字符串

1
2
3
+
...





正整数的识别

□ 正整数描述了一个集合

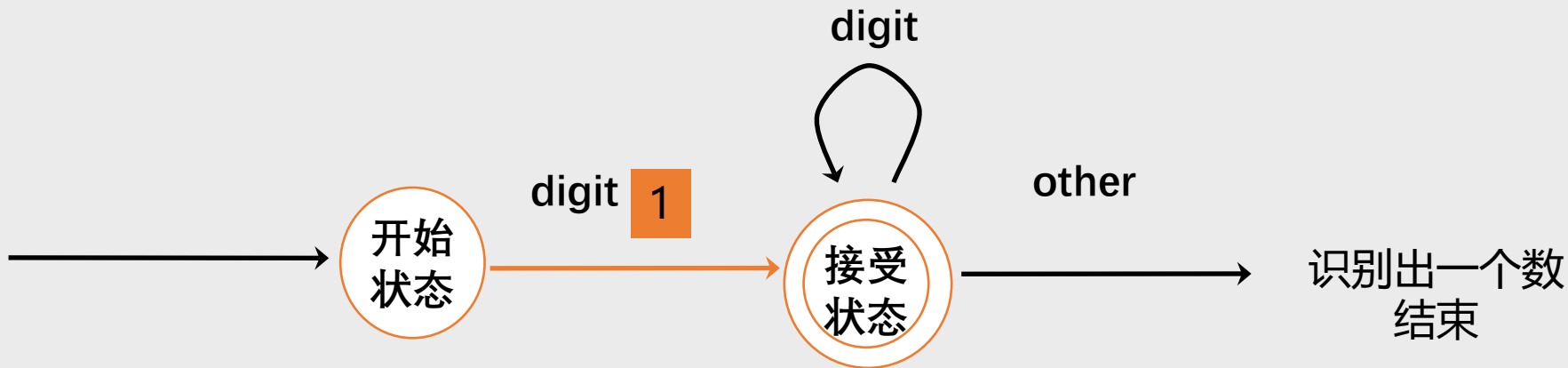
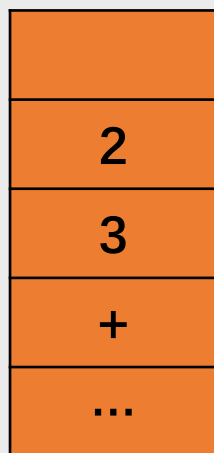
- 最基本的构成单元：0、1、2、3、...、9
- 组合形式：10、123、1001、19461、...
 - ❖ 可以看做由基本单元不断拼接而形成的串

正则表达式

$\text{digit} \rightarrow 0|1|2|\cdots|9$

$\text{digits} \rightarrow \text{digit digit}^*$

字符串





正整数的识别

□ 正整数描述了一个集合

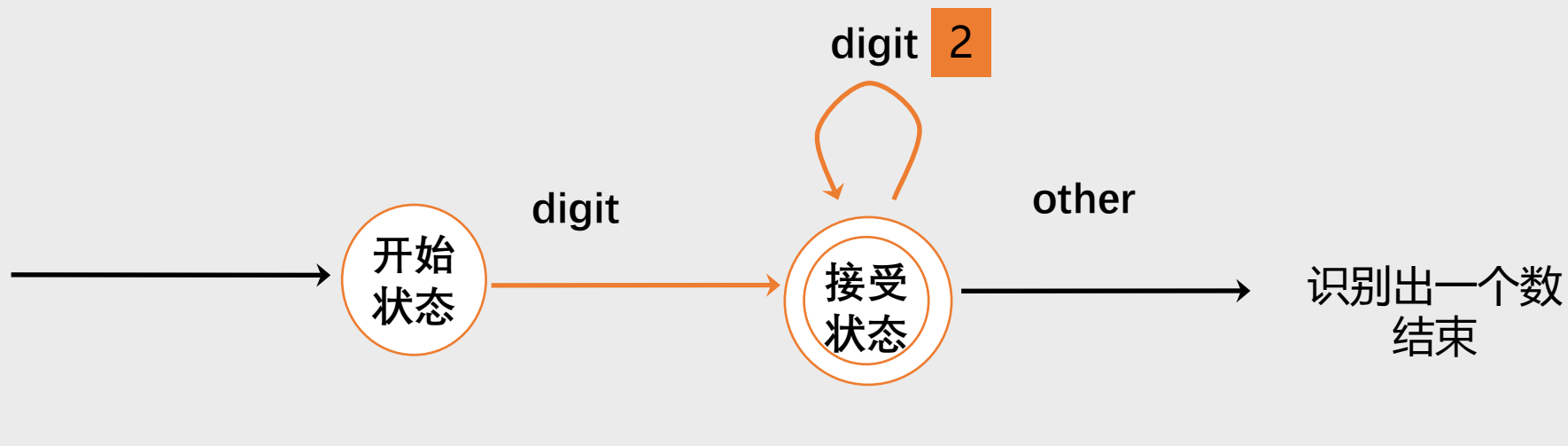
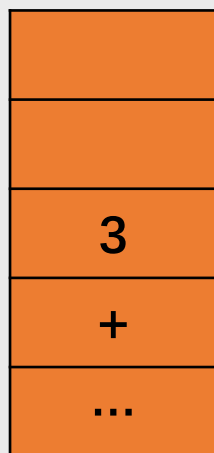
- 最基本的构成单元：0、1、2、3、...、9
- 组合形式：10、123、1001、19461、...
 - ❖ 可以看做由基本单元不断拼接而形成的串

正则表达式

$\text{digit} \rightarrow 0|1|2|\cdots|9$

$\text{digits} \rightarrow \text{digit digit}^*$

字符串





正整数的识别

□ 正整数描述了一个集合

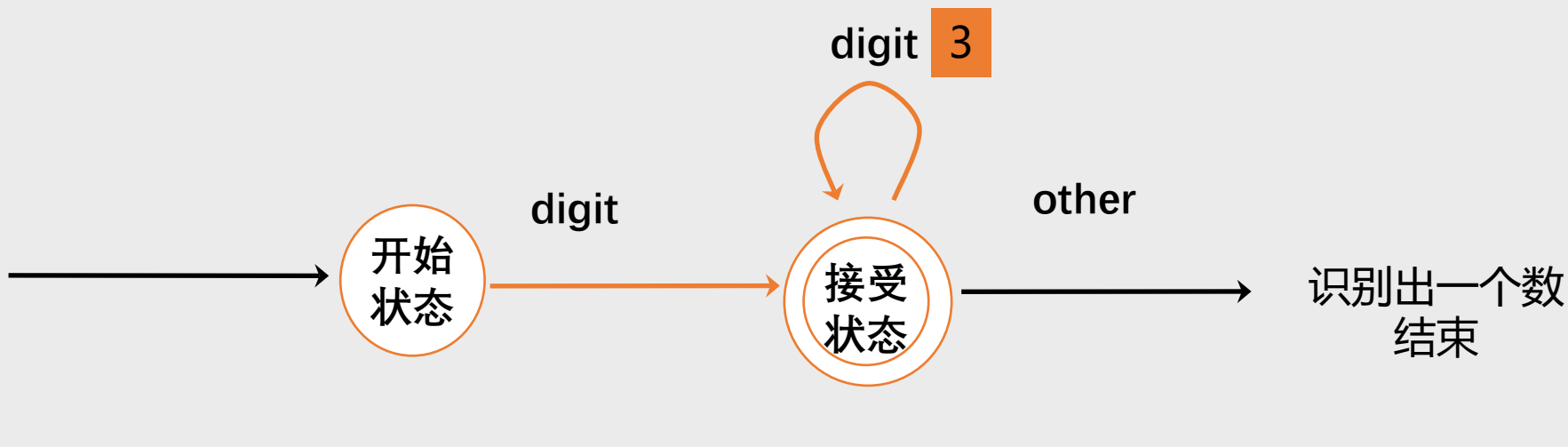
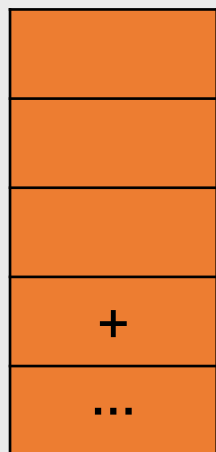
- 最基本的构成单元：0、1、2、3、...、9
- 组合形式：10、123、1001、19461、...
 - ❖ 可以看做由基本单元不断拼接而形成的串

正则表达式

$\text{digit} \rightarrow 0|1|2|\cdots|9$

$\text{digits} \rightarrow \text{digit digit}^*$

字符串





正整数的识别

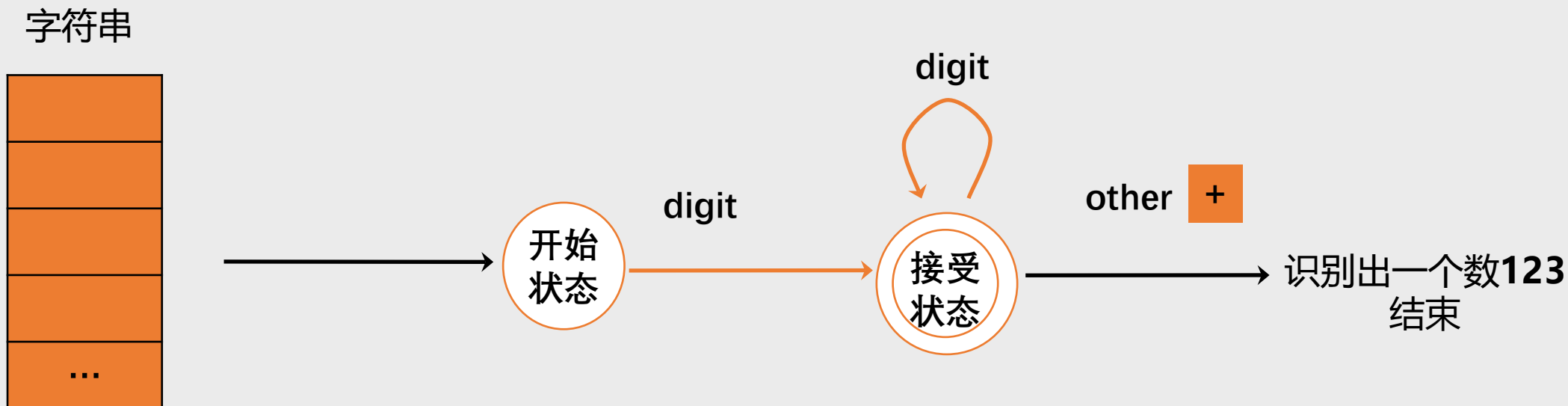
□ 正整数描述了一个集合

- 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 \rightarrow 0|1|2|...|9

digits \rightarrow digit digit*





正整数的识别

□ 正整数描述了一个集合

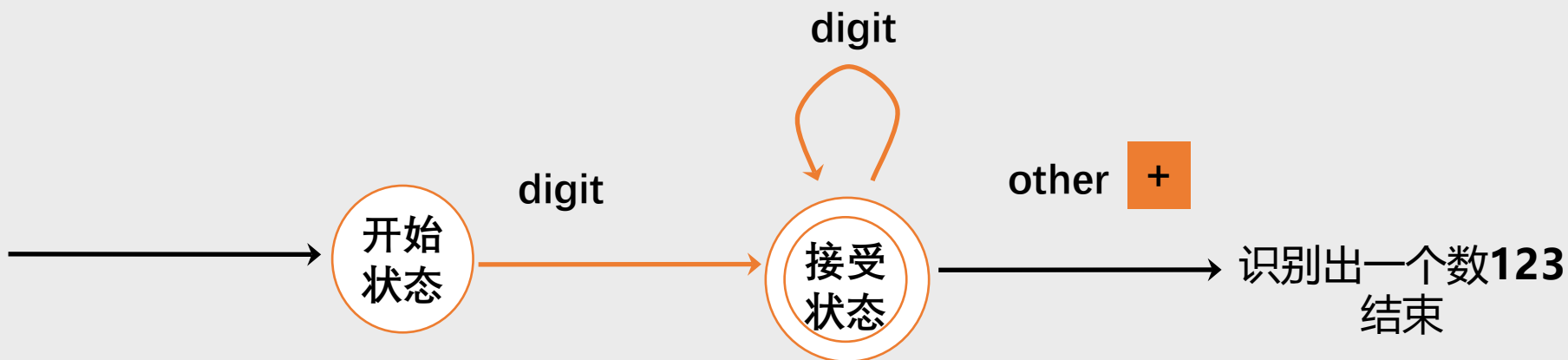
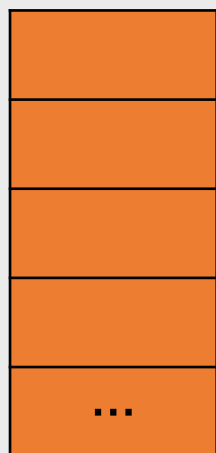
- 最基本的构成单元：0、1、2、3、...、9
- 组合形式：10、123、1001、19461、...
 - ❖ 可以看做由基本单元不断拼接而形成的串

正则表达式

digit \rightarrow 0|1|2|...|9

digits \rightarrow digit digit*

字符串



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)

8848 . 86

整数部分：
至少有一个数字的串

小数部分：
至少有一个数字的串

小数点
特殊的符号



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

整数部分 digits $\rightarrow \text{digit digit}^*$

小数部分 digits $\rightarrow \text{digit digit}^*$

带小数的数字串 number $\rightarrow \text{digit digit}^*.\text{digit digit}^*$

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 \rightarrow [0-9]

整数部分 digits \rightarrow digit⁺

小数部分 digits \rightarrow digit⁺

带小数的数字串 number \rightarrow digit⁺ . digit⁺

简写形式

正则表达式
(Regular Expression)

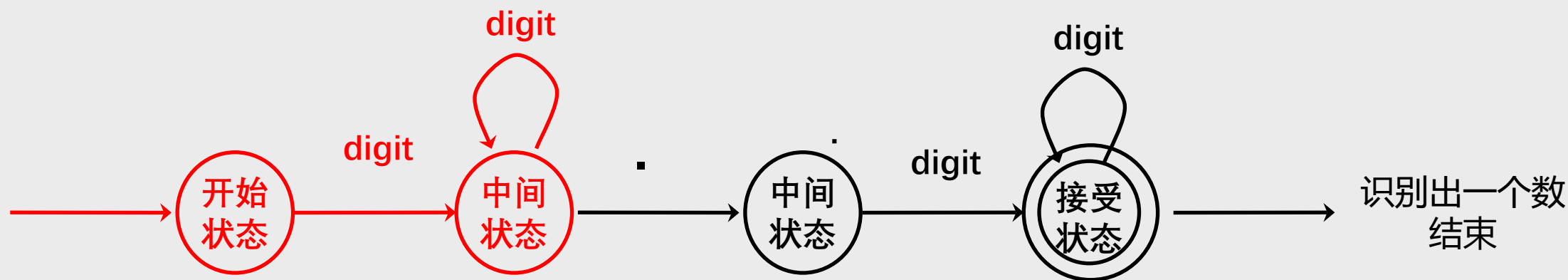


带小数的数如何识别?

□ 1.5, 10.28, 237.8, 8848.86

正则表达式

number \rightarrow **digit⁺** . digit⁺



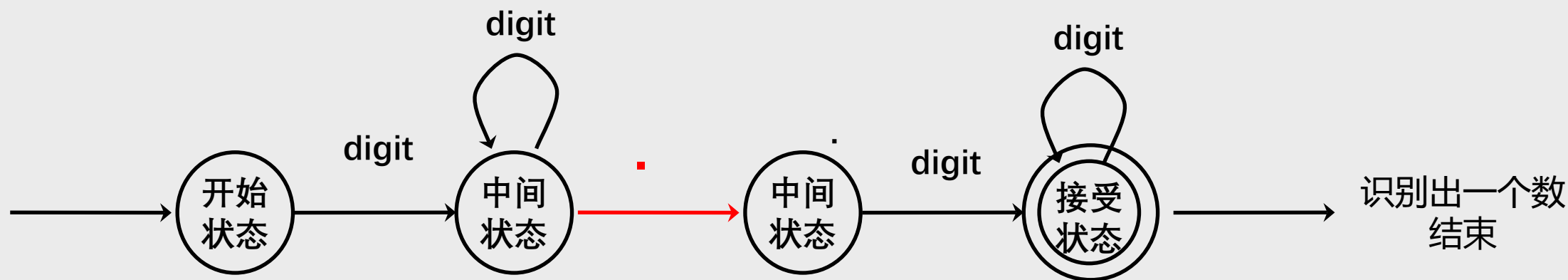


带小数的数如何识别?

□ 1.5, 10.28, 237.8, 8848.86

正则表达式

$\text{number} \rightarrow \text{digit}^+ . \text{digit}^+$



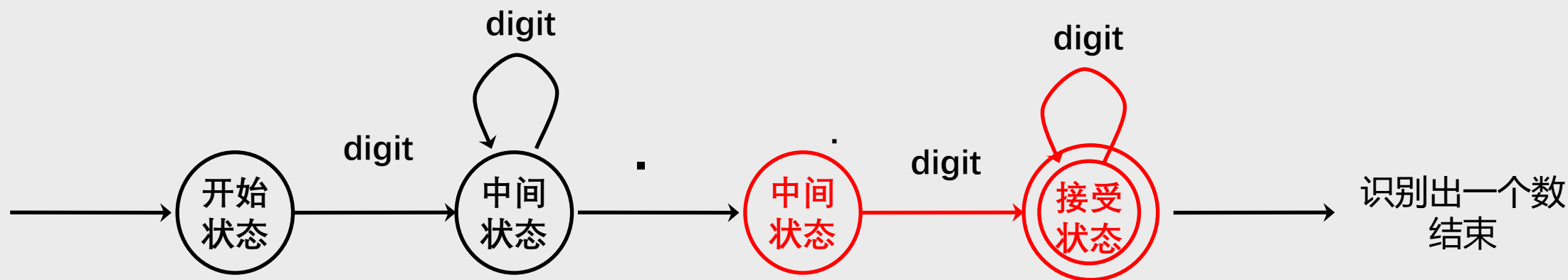


带小数的数如何识别?

□ 1.5, 10.28, 237.8, 8848.86

正则表达式

number \rightarrow digit⁺ . digit⁺





有限自动机的定义

❑ (不确定的) 有限自动机NFA是一个数学模型, 它包括:

- ❖ 有限的状态集合 S
- ❖ 输入符号集合 Σ
- ❖ 转换函数 $move : S \times (\Sigma \cup \{\epsilon\}) \rightarrow P(S)$
- ❖ 状态 s_0 是唯一的开始状态
- ❖ $F \subseteq S$ 是接受状态集合

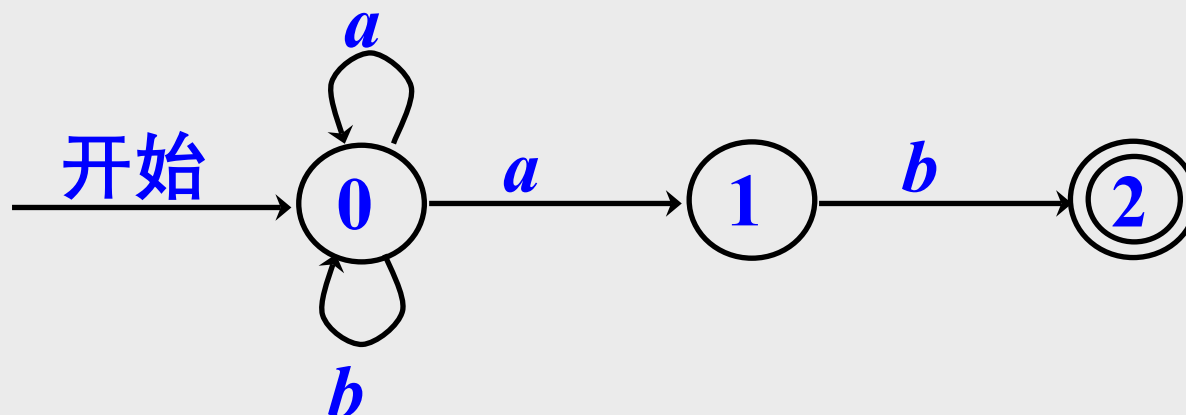


有限自动机的定义

❑ (不确定的) 有限自动机NFA是一个数学模型，它包括：

- ❖ 有限的状态集合 S
- ❖ 输入符号集合 Σ
- ❖ 转换函数 $move : S \times (\Sigma \cup \{\epsilon\}) \rightarrow P(S)$
- ❖ 状态 s_0 是唯一的开始状态
- ❖ $F \subseteq S$ 是接受状态集合

识别语言
 $(a|b)^*ab$
的NFA



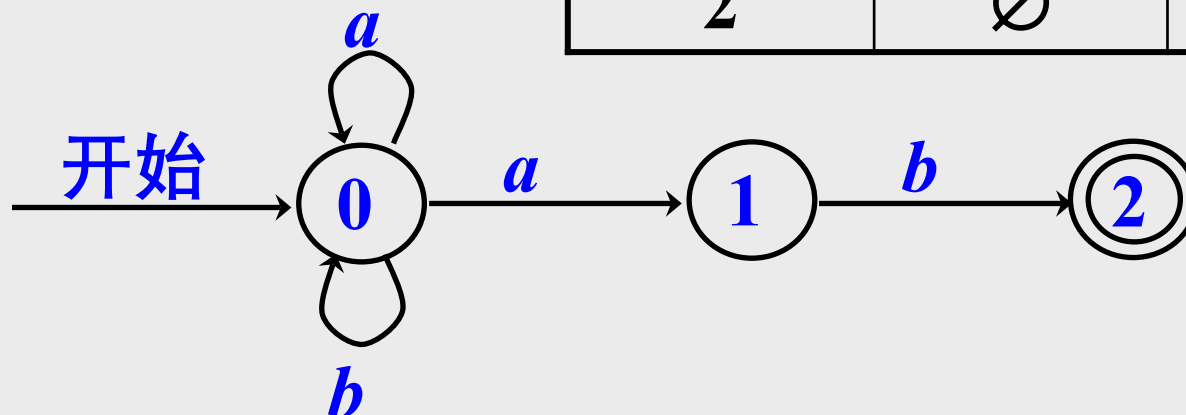


有限自动机的实现

- ❑ 构造状态之间的转换表，在读入字符串的过程中，不停查表，直至到达接受状态
- ❑ 或者，报告非法输入

	输入符号	
	a	b
0	$\{0, 1\}$	$\{0\}$
1	\emptyset	$\{2\}$
2	\emptyset	\emptyset

识别语言
 $(a|b)^*ab$
的NFA





本节小结

❑ 词法分析为源代码分词，且识别词的合法性，包括以下步骤

- Step1：确定描述单词合法性的正则表达式
- Step2：将正则表达式转换为有限自动机
- Step3：生成自动机的状态转换图
- Step4：从左到右依次读入源代码中的字母，查询状态转换图
 - ❖ Step 4.1 前进直至单词被成功识别
 - ❖ Step 4.2 回退，沿着4.1继续尝试别的表达式
 - ❖ Step 4.3 报错，提醒用户输入有问题



拓展与思考

□ 问题一：可否为正则表达式生成有限自动机？

■ 请预习参考书中下一节子集构造法和算法3.23。

□ 问题二：有限自动机如何实现为代码？

■ 请课外阅读[有限自动机的Python实现样例](#)



课后作业

❑ **请完成Lab1中的实验，在词法分析器Flex中为Cminus语言的词法写对应的正则表达式**

- 提交内容：代码+实验文档（markdown）
- 提交方式：直接上传到gitlab的仓库中
- 提交时间：9月30日晚上20:00前，以时间戳为准
- 评分要求：代码测试（80%）+文档质量（20%）

《编译原理和技术》

词法分析

谢谢！