软件设计师
--程序语言基础知识

高级项目经理　任铄
QQ：1530841586

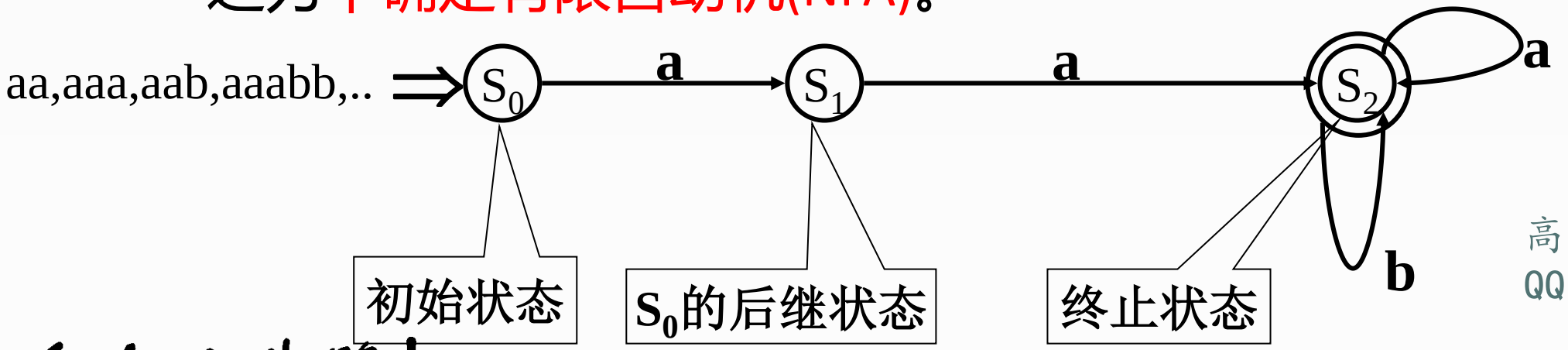
第二章 程序语言基础知识

- 2.1 基础知识
- 2.2 编译系统基本原理
- 2.3 程序语言的控制结构

向上人生路！

- 初始状态
- 终止状态(接收状态):
- 后继状态：有限状态机在读入一个字符时,其状态改变为另一状态，则改变后的状态被称为后继状态。

如果有限状态机每次转换后的状态是唯一的则称之为确定有限状态自动机(DFA)；如果转换后的后继状态不是唯一的则称之为不确定有限自动机(NFA)。



高级项目经理　任铄
QQ：1530841586

向上人生路！

二、确定有限自动机（DFA）

一个确定有限自动机（DFA） M是一个五元式：

M=(S , ∑ , δ , s0 , F)，其中

1、S是一个有限集，它的每个元素称为一个状态

2、∑是一个有穷字母表，它的每个元素称为一个输入字符

3、 δ是一个从S × ∑至S的单值部分映射。 δ（s,a）=s′ 意味着：当现行状态为S，输入a时，将转换到下一状态s′ 我们称s′ 为s的一个后继状态。

4、s0∈S是唯一的初态

5、F⊆S是一个终态集（可空）

高级项目经理 任铄
QQ：1530841586

例:AD=(S, ∑ , δ ,K , F)

其中:

S={S0, S1, S2, S3};

∑={a,b,c}

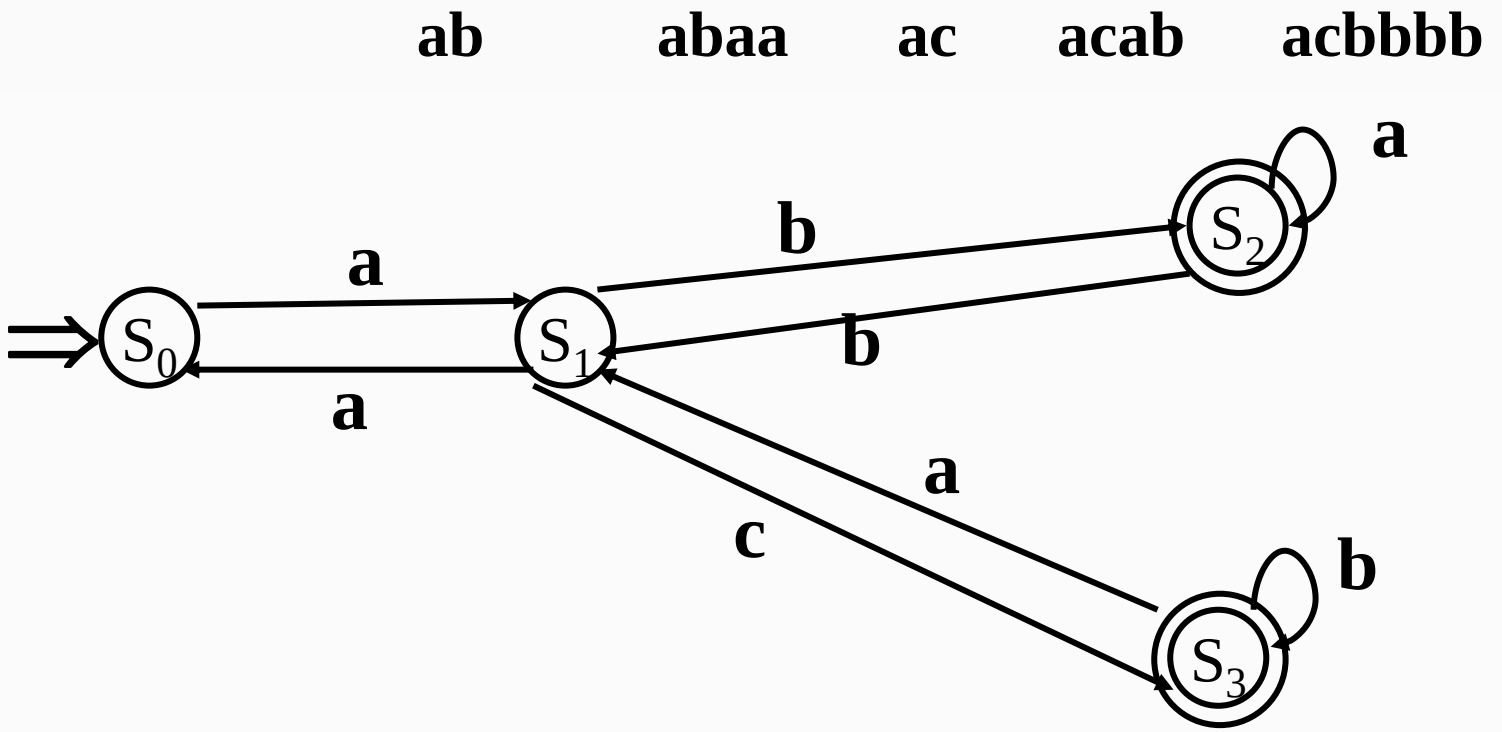
K= S0;

F={S2, S3};

δ 转换函数:

(S0,a)= S1;　(S1,a)= S0;　(S1,b)= S2;　(S1,c)= S3;

(S2,a)= S2;　(S2,b)=S1;　(S3,a)= S1;　(S3,b)= S3;



向上人生路!

②语法分析阶段

语法分析器以单词符号作为输入，分析单词符号串是否形成符合语法规则的语法单位，如表达式、赋值、循环等，

按语法规则分析检查每条语句是否有正确的逻辑结构。

```
int arr[2],b;
b = arr * 10;
```

语法分析的方法：

- 自上而下分析法
- 自下而上分析法

高级项目经理　任铄
QQ：1530841586

③语义分析阶段

语义分析阶段主要是检查源程序是否存在语义错误，并收集类型信息供后面的代码生成阶段使用，只有语法和语义都正确的源程序才能翻译成正确的目标代码。

语义分析的主要工作是进行各类型分析和检查。赋值语句的右端和左端的类型不匹配。表达式的除数是否为零等。

```
int arr[2],b;
b = arr * 10;
```

高级项目经理　任铄
QQ：1530841586

④中间代码生成阶段

中间代码生成阶段的工作是根据语义分析的输出生成中间代码。

- 中间代码是一种简单且含义明确的记号系统，可以有若干种形式，常见的有逆波兰记号、四元式、三元式和树。
- 他们的共同特征是代码的方式与具体的机器无关。

高级项目经理　任铄
QQ：1530841586

⑤代码优化阶段

代码优化阶段是对前阶段产生的中间代码进行变换或进行改造，目的是使生成的目标代码更为高级，即省时间和省空间。

高级项目经理　任铄
QQ：1530841586

⑥目标代码生成阶段

是把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。这是编译的最后阶段，它的工作与硬件系统的结构和指令的含义有关。

高级项目经理　任铄
QQ：1530841586

第二章 程序语言基础知识

- 2.1 基础知识
- 2.2 编译系统基本原理
- 2.3 程序语言的控制结构

向上人生路！

一、表达式

- 前缀表达式：也被称为波兰表示法，其特点是将操作符置于操作数之前，如：- × + 3 4 5 6 。
- 中缀表达式：即我们常用的表示方法，(3 + 4) × 5 - 6 。
- 后缀表达式：又被称为逆波兰法，其特点是将操作符置于操作数之后，3 4 + 5 × 6 - 。

高级项目经理　任铄
QQ：1530841586

1、前缀表达式

从右至左扫描表达式，遇到数字时，将数字压入堆栈，遇到运算符时，弹出栈顶的两个数，用运算符对它们做相应的计算（栈顶元素 op 次顶元素），并将结果入栈；重复上述过程直到表达式最左端，最后运算得出的值即为表达式的结果。

例如前缀表达式“- × + 3 4 5 6”：

高级项目经理　任铄
QQ：1530841586

2、中缀表达式（中缀记法）

中缀表达式是一种通用的算术或逻辑公式表示方法，操作符以中缀形式处于操作数的中间。中缀表达式是人们常用的算术表示方法。

虽然人的大脑很容易理解与分析中缀表达式，但对计算机来说中缀表达式却是很复杂的，因此计算表达式的值时，通常需要先将中缀表达式转换为前缀或后缀表达式，然后再进行求值。

高级项目经理　任铄
QQ：1530841586

3、后缀表达式

与前缀表达式类似，只是顺序是从左至右：

从左至右扫描表达式，遇到数字时，将数字压入堆栈，遇到运算符时，弹出栈顶的两个数，用运算符对它们做相应的计算（次顶元素 op 栈顶元素），并将结果入栈；重复上述过程直到表达式最右端，最后运算得出的值即为表达式的结果。

例如后缀表达式“3 4 + 5 × 6 -”：

● 算术表达式(a–b)*c+d 的后缀式是__(22)__（–、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

（22）A. a b c d – * +　　B. a b – c d * +

C. a b – c * d +　　D. a b c – d * +

向上人生路！

二、操作符的优先级

- 指针最优，单目运算优于双目运算。如正负号。
- 先乘除（模），后加减。
- 先算术运算，后移位运算，最后位运算。

  1 << 3 + 2 & 7等价于 (1 << (3 + 2))&7
- 逻辑运算最后计算。

优先级	运算符	结合性
1	() [] .	从左到右
2	! +(正) -(负) ~ ++ --	从右向左
3	* / %	从左向右
4	+(加) -(减)	从左向右
5	<< >> >>>	从左向右
6	< <= > >= instanceof	从左向右
7	== !=	从左向右
8	&(按位与)	从左向右
9	^	从左向右
10	|	从左向右
11	&&	从左向右
12	||	从左向右
13	?:	从右向左
14	= += -= *= /= %= &= |= ^= ~= <<= >>= >>>=	从右向左

三、语句间的结构

- 顺序语句
- 选择语句
- 循环语句

高级项目经理　任铄
QQ：1530841586

四、过程控制

```
int Function1 (int x,int y)
  {
  .......
   }
```

参数传递的方式

- 传值调用 -- 数据传送是单向的
- 引用调用(地址调用)--数据传送是双向的

高级项目经理　任铄
QQ：1530841586

```
void Exchg1(int x, int y)
{ int tmp;
  tmp=x;
  x=y;
  y=tmp;
  printf("Exchg1:x=%d,y=%d\n",x,y);
}
void Exchg2(int *x, int *y)
{  int tmp;
  tmp=*x;
  *x=*y;
  *y=tmp;
  printf("Exchg2:x=%d,y=%d\n",*x,*y);
}
```

```
void main()
{
  int a=4,b=6;
  Exchg1 (a,b) ;
  printf("a=%d,b=%d\n",a,b);
  Exchg2(&a,&b) ;
  printf("a=%d,b=%d\n",a,b);
}
```

高级项目经理　任铄
QQ：1530841586

可以通过下列渠道沟通联系：

1、 QQ:1530841586

2、 QQ群：164955673

向上人生路！