

第4讲 从多项式计算规则变化看自动计算 —规则与迭代

战 德 臣

哈尔滨工业大学计算学部教学委员会主任
国家教学名师

18686783018, dechen@hit.edu.cn

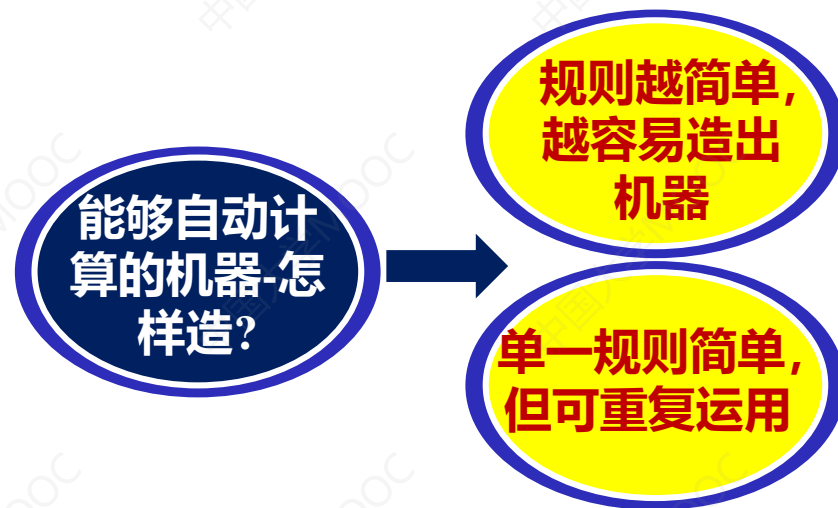
自动计算要解决什么问题

两对象-三机制

两个对象：**数据** 与 **计算规则**

三个机制：**表示-存储-执行**

- ◆ **数据** 的表示：**进位制与编码**
- ◆ **计算规则** 的表示：**程序**
- ◆ 数据与计算规则的 **自动存储**
- ◆ 计算规则/程序的 **自动执行**



计算规则

示例：怎样计算多项式的值-差分法

用简单计算规则（如加减法运算）求解复杂计算问题（如乘方运算和多项式运算）

例如：求 n^2

$$(n+1)^2 = n^2 + \alpha_n + \beta_n$$

n	n^2	一阶差分 $\alpha_n = n^2 - (n-1)^2$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	0		
1	1	1	
2	4	3	2
3	9	5	2
4	16	7	2
5	25	9	2

计算规则

示例：怎样计算多项式的值-差分法

用简单计算规则（如加减法运算）求解复杂计算问题（如乘方运算和多项式运算）

例如：求 $F(x) = x^2 + 2x + 3$

初始值不一样，可计算不同的多项式

$$F(n+1) = F(n) + \alpha_n + \beta_n$$

n	F(n)	一阶差分 $\alpha_n = F(n) - F(n-1)$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	3		
1	6	3	
2	11	5	2
3	18	7	2
4	27	9	2
5	38	11	2

计算规则表达与执行

5

计算规则表达【程序】与计算规则执行【计算机器】

/*类C语言表达的计算规则—程序

Main()

{

int k, n, F[], alpha[], beta[];

input k;

F[0]=3;

F[1]=6;

F[2]=11;

alpha[1] = 3;

for n=2 to k-1

{

alpha[n] = F[n] - F[n-1];

beta[n] = alpha[n] - alpha[n-1];

F[n+1] = F[n] + alpha[n] + beta[n];

}

output F[k];

}

能够保存数据

T: 输入不同的初始
值便可计算不同的一
元二次多项式的值

能够重复进行

能够完成计算

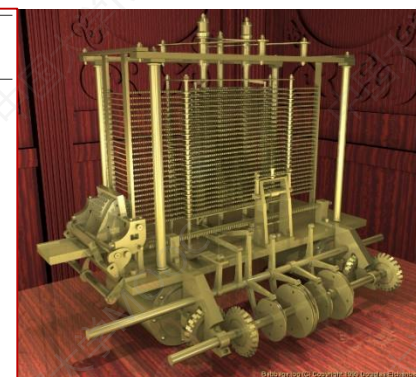
Q: 保存的数据
随K值增大而增
多, 怎么办?

F[0] 保存F(0)的值
F[1] 保存F(1)的值
F[2] 保存F(2)的值
F[3]
F[4]
F[5]
...

变量, 是程序
执行过程中其
值可发生变化
的量

数组变量, 可
以按序号保存
多个值

n	F(n)	一阶差分 $\alpha_n = F(n) - F(n-1)$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	3		
1	6	3	
2	11	5	2
3	18	7	2
4	27	9	2
5	38	11	2



计算规则表达与执行

6

怎样用更少的变量来实现自动计算—降低机器构造的复杂性?

/*类C语言表达的计算规则—程序

Main()

```
{ int k, n, F_nminus1, F_n, F_nplus1,  
  int alpha_nminus1, alpha_n, beta_n;  
  input k;  
  F_nminus1 = 1; F_n = 4; alpha_nminus1 = 1;  
  for n=2 to k-1  
  {  
    alpha_n = F_n - F_nminus1;  
    beta_n = alpha_n - alpha_nminus1;  
    F_nplus1 = F_n + alpha_n + beta_n;  
    F_nminus1 = F_n;  
    F_n = F_nplus1;  
    alpha_nminus1 = alpha_n;  
  }  
  output F_nplus1;  
}
```

轮流计
算新值

为下次计算保
留相关数据

保存的数据不随
K值变化而变化

F_nminus1

类似的, 就是一个变量
名, 可任意命名, 可保
存一个数值

F[0] 保存F(0)的值
F[1] 保存F(1)的值
F[2] 保存F(2)的值
F[3]
F[4]
F[5]
...

只用1个变量
保存当前数值

占用n个变量
保存n个数值

计算规则表达与执行

迭代式方法的执行过程

/*类C语言表达的计算规则—程序

Main()

```
{ int k, n, F_nminus1, F_n, F_nplus1,  
  int alpha_nminus1, alpha_n, beta_n;  
  input k;
```

```
  F_nminus1 = 1; F_n = 4; alpha_nminus1 = 1;
```

```
  for n=2 to k-1
```

```
  {
```

```
    alpha_n = F_n - F_nminus1;
```

```
    beta_n = alpha_n - alpha_nminus1;
```

```
    F_nplus1 = F_n + alpha_n + beta_n;
```

```
    F_nminus1 = F_n;
```

```
    F_n = F_nplus1;
```

```
    alpha_nminus1 = alpha_n;
```

```
  }
```

```
  output F_nplus1;
```

```
}
```

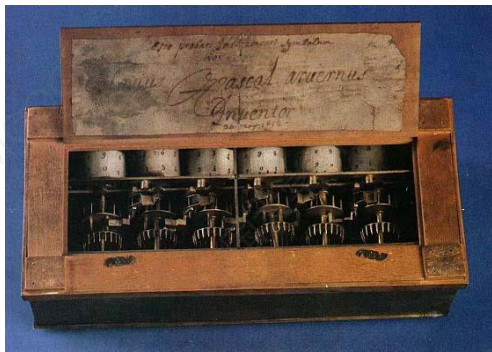
n	F_nplus1	F_n	F_nminus1	alpha_n	alpha_nminus1	beta_n
2	9	4	1	3	1	2
3	16	9	4	5	3	2
4	25	16	9	7	5	2
5	36	25	16	9	7	2
		36	25		9	

自动计算的探索

8

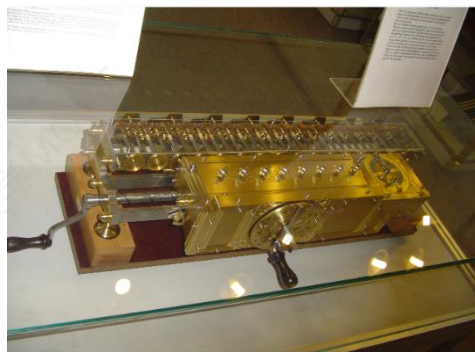
由机械计算机到电子计算机

Pascal机械计算机: **第一台机械计算机-能够自动完成计算**



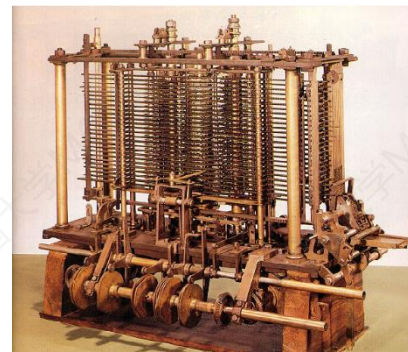
帕斯卡, B.

莱布尼茨机械计算机: **自动计算--固定的计算规则**



莱布尼兹, G. W.

Babbage分析机: **特定程序 -- 可有限变化的计算规则**



巴贝奇

现代计算机: **一般程序 -- 任意可变的计算规则**



约翰·冯·诺伊曼

自动计算的探索

9

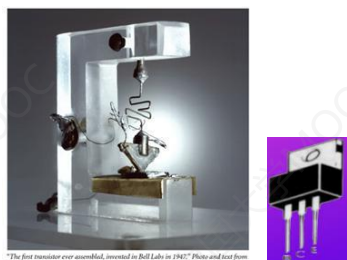
电子计算机的元件探索

电子管: 可自动控制
0和1变化的元件



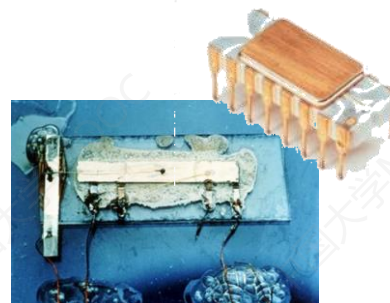
人类第一只电子管
(真空二极管), 1895

晶体管: 可自动控制
0和1变化的元件



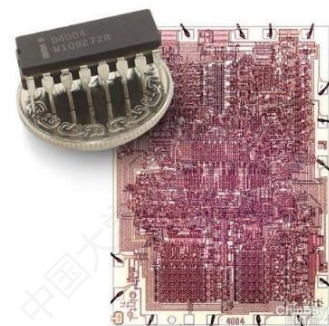
人类第一只晶体管
(点接触晶体管), 1947

集成电路: 可自动实
现一定变换的元件



集成电路的发明, 1959

**超大规模集成电路
(VLSI)**



VLSI出现, 1974



电子管计算机ENIAC,
1946年, 17468只电子管



第一台晶体管计算机
TRADIC, 1953



第三代计算机
IBM360, 1964

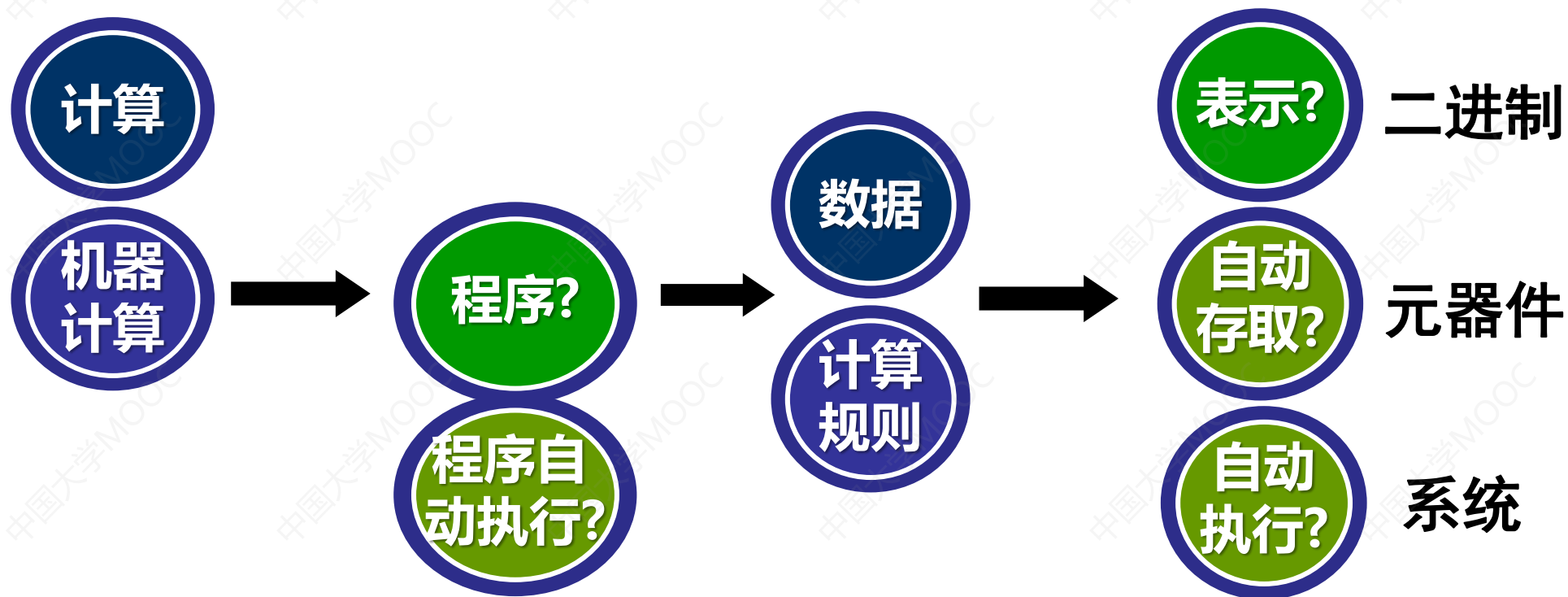


第四代计算机—
个人计算机, 1981

计算机器

10

计算机器的探索思路

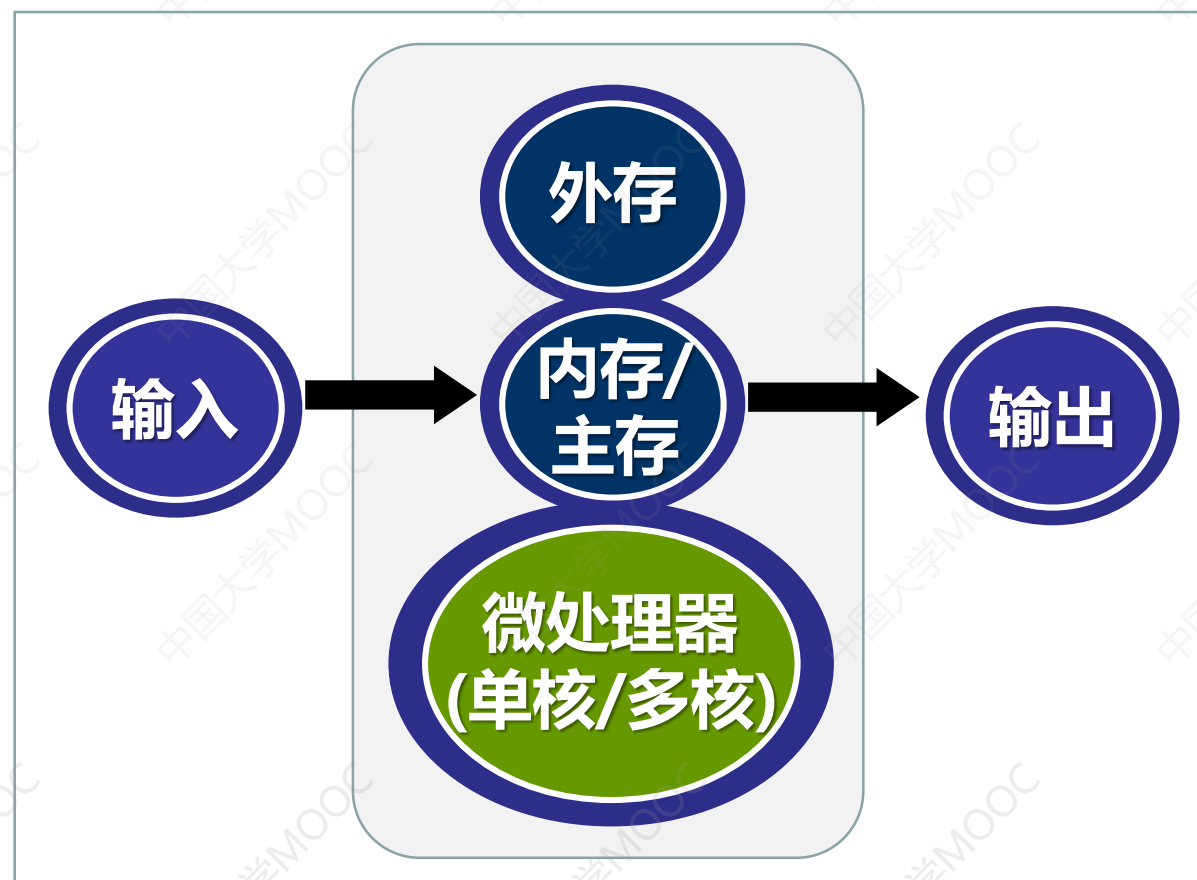
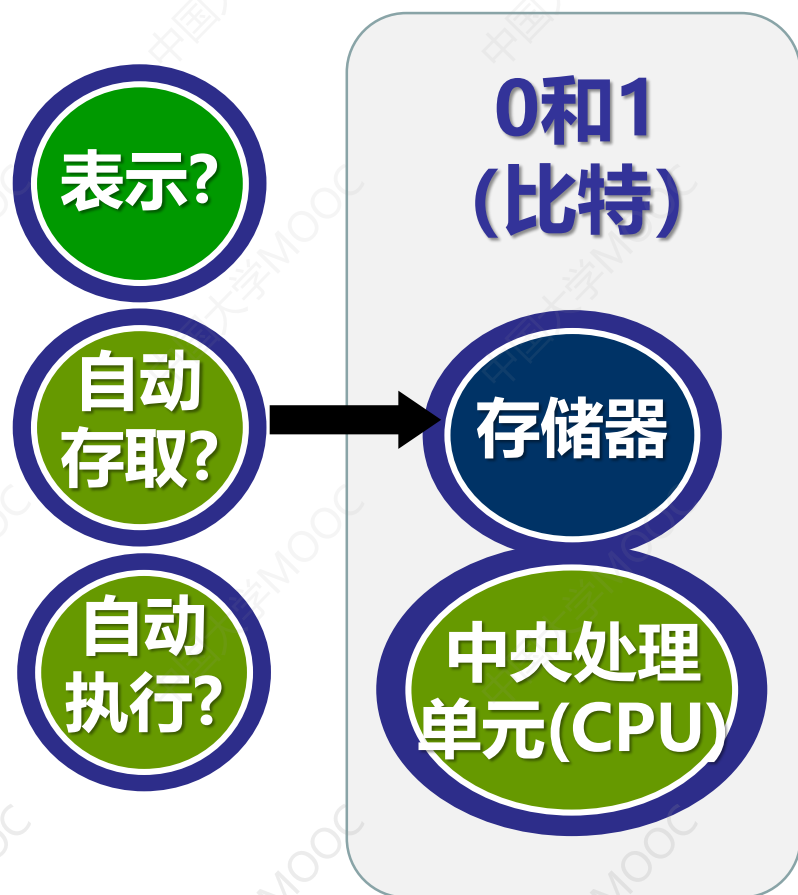


计算机器

11

计算机：能够自动执行程序的机器

个人计算机



计算机器

概念计算机 与 现实中典型的个人计算机



如何降低计算机器制造难度

13

小结

两个对象：**数据** 与 **计算规则**

三个机制：**表示-存储-执行**

- ◆ **数据** 的表示
- ◆ **计算规则** 的表示：**程序**
- ◆ 数据与计算规则的 **自动存储**
- ◆ 计算规则/程序的 **自动执行**

