



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

逻辑数据的运算与处理

- ◆ 3、用布尔代数简化逻辑式子.....
- ◆ 4、条件运算.....
- ◆ 5、其他运算.....
- ◆ 6、常量表达式.....

2.4.5 逻辑运算符

► 用布尔代数简化逻辑式子

- 布尔代数（开关代数、逻辑代数）起源于数学领域，用于集合运算和逻辑运算。

基本定律	$A + 0 = A$	$A \cdot 0 = 0$	$\overline{\overline{A}} = A$
	$A + 1 = 1$	$A \cdot 1 = A$	
	$A + A = A$	$A \cdot A = A$	
	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$	
结合律	$(A + B) + C = A + (B + C)$		$(AB)C = A(BC)$
交换律	$A + B = B + A$		$AB = BA$
分配律	$A(B + C) = AB + AC$		$A + BC = (A + B)(A + C)$
摩根定律	$\overline{A \cdot B \cdot C \cdots} = \overline{A} + \overline{B} + \overline{C} \cdots$		$\overline{A + B + C \cdots} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdots$
吸收律	$A + A \cdot B = A$		
	$A \cdot (A + B) = A$		
	$A + \overline{A} \cdot B = A + B$		
	$(A + B) \cdot (A + C) = A + BC$		

2.4.5 逻辑运算符

► 用布尔代数简化逻辑式子

- (1) 代入规则：任何一个含有变量A的等式，如果将所有出现 A 的位置都代入同一个逻辑函数，则恒等式成立。
- 【例】
- $B(A+C)=BA+BC$ ，现将所有出现A的地方都代入函数 $A+D$ ，则有
- $B[(A+D)+C]=B(A+D)+BC=BA+BD+BC$

2.4.5 逻辑运算符

► 用布尔代数简化逻辑式子

- (2) 反演规则：使用摩根定律来求一个逻辑函数F的非函数的规则：
 - ①将 F 表达式中的与(·)换成或(+), 或(+)换成与(·);
 - ②将原变量换成非变量, 非变量换成原变量;
 - ③将逻辑1换成0, 0换成1。
- 注意：变换中必须保持先与后或的顺序，否则将会出错。
- 【例】

$$F = \overline{\overline{A} \cdot \overline{B} + C \cdot D} = (A+B) \cdot (\overline{C} + \overline{D})$$

2.4.5 逻辑运算符

► 用布尔代数简化逻辑式子

- (3) 对偶规则：某个逻辑恒等式成立，则对偶式也成立，称为对偶规则。F 是一个逻辑表达式，把F中的与(\cdot)换成或($+$)，或($+$)换成与(\cdot)；1换成0，0换成1，所得的新的逻辑函数式叫F的对偶式，记为F'。
- 【例】
- $F = (A + \overline{B}) + (A + C), F' = \overline{A} \overline{B} + A C,$

2.4.5 逻辑运算符

▶ 用布尔代数简化逻辑式子

▶ 【例】

$$\text{▶ } F = \overline{A}BC + \overline{A}B\overline{C} = \overline{A}B(C + \overline{C}) = \overline{A}B$$

2.4.6 条件运算符

表2-8 条件运算符

运算符	功能	目	结合性	用法
? :	条件运算	三目	自右向左	expr1 ? expr2 : expr3

2.4.6 条件运算符

- ▶ (1) 条件运算符的运算对象可以是常量、变量或表达式，可以是任意数据类型。无论类型和结果如何，`expr1`总是按逻辑值对待。

2.4.6 条件运算符

- ▶ (2) 当使用条件运算符的结果时，expr2和expr3常常要统一数据含义，否则运算结果就有二义性。
- ▶ 示例

```
r = a > b ? length : volume  
//若length是长度，volume是体积时，则结果r出现二义性
```

2.4.6 条件运算符

- ▶ 当不使用关系运算符的结果时，expr2和expr3是可以各自独立的。
- ▶ 示例

```
a>b ? L=length : V=volume //根据a>b的真假选择两个赋值之一
```

- ▶ 利用条件运算符可以实现分项处理、分段函数计算等。

2.4.6 条件运算符



【例2.7】

写出分段函数

$$y = \begin{cases} ax + b & x \geq 0 \\ x & x < 0 \end{cases}$$

的C语言表达式。

解：

$$y = x >= 0 ? a * x + b : x ;$$

2.4.9 取长度运算符

表2-12 取长度运算符

运算符	功能	目	结合性	用法
sizeof	取长度运算	单目	自左向右	sizeof expr、sizeof(expr)或sizeof (typename)

sizeof有三种形式：

sizeof (typename)	//取类型typename的长度
sizeof (expr)	//取变量、常量或表达式的长度
sizeof expr	//取变量、常量或表达式的长度

2.4.9 取长度运算符

- ▶ sizeof是在编译时自动确定长度的，运行时sizeof表达式实际是一个无符号整型常量。程序中用sizeof运算，而不是直接用内存长度值，可以提高程序的可移植性。
- ▶ 示例

```
sizeof(char) //结果是char类型的内存大小，为1  
sizeof(unsigned long) //结果是unsigned long类型的内存大小，为4  
sizeof a //结果是变量a的内存大小，也即a的类型的内存大小  
sizeof (a+b) //结果是表达式a+b的内存大小，也即a+b的类型的内存大小  
sizeof (int) //在Visual C++ 6.0中为4，在Turbo C中为2
```

2.4.10 逗号运算符

表2-13 逗号运算符

运算符	功能	目	结合性	用法
,	逗号运算	双目	自左向右	expr1 , expr2

2.4.10 逗号运算符

► 示例

```
int i=3,j=5;  
k=i++,i+1,j++,j+1; //k值为3 (i++的值), 表达式的值为7  
k=(i++,i+1,j++,j+1); //k值为7 (j+1的值), 表达式的值为7
```


2.4.10 逗号运算符



【例2.10】

将两个整型变量a和b的值相互交换。

解：

方法一，借助第三方变量

```
int t;  
t=a, a=b, b=t; //交换a和b的值
```

方法二，不使用第三方变量

```
a=a+b, b=a-b, a=a-b; //交换a和b的值
```

2.4.11 圆括号运算符

表2-14 圆括号运算符

运算符	功能	目	结合性	用法
()	括号和函数调用	单目	自左向右	(expr)

2.4.11 圆括号运算符

- ▶ 一般地，为了清楚地表达求值顺序，应该使用圆括号运算符清晰地表明求值顺序，而不是写合法的但难于阅读的表达式

- ▶ 示例

- ▶ $((((a*x+b)*x+c)*x+d)*x+e \quad // \quad ax^4 + bx^3 + cx^2 + dx + e$

- ▶ $(a+b)/2/x, (a+b)/(2*x) \quad // \quad \frac{a+b}{2x}$ ， $(a+b)/2*x$ 是错误的

2.4.12 常量表达式

- ▶ 仅由常量、const变量和运算符组成的式子称为常量表达式。常量表达式在编译时就确定其值了，因此在程序运行时常量表达式本质上就是一个常量值。
- ▶ 示例

```
const int i=9;  
int m;  
m=i+10; //编译时i+10就已经确定为19，因此运行时这条语句实际为m=19;
```

CP 程序设计