优先级与结合性

March 22, 2017

当表达式中有多个不同优先级的运算符，运算符的优先级决定哪个运算符被优先执行。如 10 + 20 * 30 等价于 10 + (20 * 30)。

结合性用于一个表达式中两个运算符的优先级相同的情形。结合性可能是从左到右或者从右到左。如 * 与 /有相同的结合性，它们的结合性从左到右，故表达式 100 / 10 * 10 等价于 (100 / 10) * 10。

优先级与结合性是运算符的两个特征，用于确定子表达式在没有括号时的运算次序。

1. 结合性仅用于两个或两个以上具有相同优先级的情形。需要指出的是结合性没有定义 单运算符中操作数的运算次序。观察以下程序

```
// Associativity is not used in the below program. Output
// is compiler dependent.
int x = 0;
int f1()
{
  x = 5;
  return x;
}
int f2()
{
  x = 10;
  return x;
}
int main()
{
  int p = f1() + f2();
  printf("%d␣", x);
  return 0;
}
```

   在该程序中，运算符 + 的结合性为从左到右，但这并不意味着 f1() 总在 f2() 之前被调用。

2. 具有相同优先级的所有运算符有相同的结合性。这是必然的，否则编译器将不知道如何在具有相同优先级和不同结合性的表达式中确定计算次序。如 + 和 - 具有相同的结合性。

3. 后缀 ++ 与 前缀 ++ 的优先级与结合性是不同的。后缀 ++ 的优先级高于前缀 ++ ，它们的结合性也是不一样的。后缀 ++ 的结合性是从左到右，前缀 ++ 的结合性是从右到左。

4. 逗号具有最低优先级，使用时需谨慎。

```
#include<stdio.h>
int main()
{
    int a;
    a = 1, 2, 3; // Evaluated as (a = 1), 2, 3
    printf("%d", a);
    return 0;
}
```

5. 对诸如 c > b > a的表达式，C将解释为 (c > b) > a，而在Python中，该表达式将解释为 a < b and b < c。

```c
#include <stdio.h>
int main()
{
  int a = 10, b = 20, c = 30;

  // (c > b > a) is treated as ((c > b) > a), associativity of '>'
  // is left to right. Therefore the value becomes ((30 > 20) > 10)
  // which becomes (1 > 20)
  if (c > b > a)
    printf("TRUE");
  else
    printf("FALSE");
  return 0;
}
```

Table 1: 运算符的优先级与结合性

运算符	描述	结合性
()	圆括号(函数调用)	从左到右
[]	方括号(数组下标)	
.	通过结构体、共同体等获取成员	
->	通过指针获取成员	
++ --	后缀自增/后缀自减	
++ --	前缀自增/前缀自减	
从右到左 + -	正/负	
!	逻辑非/位补	
(type)	强制类型转换	
*	取值运算符	
&	取址运算符	
sizeof		
* / %	乘/除/求余	从左到右
+ -	加/减	从左到右
<< >>	位左移/位右移	从左到右
< <=	小于/小于等于	从左到右
> >=	大于/大于等于	从左到右
== !=	等于/不等于	从左到右
&	位与	从左到右
^	位异或	从左到右
|	位或	从左到右
&&	逻辑与	从左到右
||	逻辑或	从左到右
?:	三元条件	从右到左
=	赋值	从右到左
+= -=	加/减赋值	
*= /=	乘/除赋值	
%= &=	求余/位与赋值	
^= !=	位异或/位或赋值	
<<= >>=	位左移/位右移赋值	
,	逗号(分离表达式)	从左到右