



Expresiones Lógicas y Operadores sobre Bits

Andrés Arcia
Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes



Expresión Lógica

El resultado de la evaluación de la expresión es un valor lógico (cierto o falso). Se construye mediante

- Operadores relacionales

- Operadores lógicos

Operadores relacionales

Operan sobre operandos (constantes, variables, expresiones) de diferentes tipos (numéricos, caracteres) y dan como resultado un valor de tipo lógico (cierto o falso; 1 o 0).

Operación	Operador Alg.	Op. C	Exp. C
Igualdad	=	==	a==b
Desigualdad	≠	!=	a!=b
Menor	<	<	a<b
Menor Igual	≤	<=	a<=b
Mayor	>	>	a>b
Mayor Igual	≥	>=	a>=b


Recuerde que en lenguaje C es muy facil confundir el operador de asignación = (igual) con el de comparación == (igual igual)

Operadores Relacionales

Ejemplos:

a	b	a==b	a!=b	a < b	a<=b	a > b	a>=b
2	3	falso	cierto	cierto	cierto	falso	falso
'a'	'd'	falso	cierto	cierto	cierto	falso	falso
44	44	cierto	falso	falso	cierto	falso	cierto

Precedencia de los Operadores Relacionales



() Subexpresiones encerradas entre paréntesis se evalúan primero (mayor nivel de precedencia).

<, <=, >, >=

==, != (menor nivel de precedencia)

La evaluación de los operadores con la misma prioridad se realiza de izquierda a derecha.

Observación

$A = B$ asigna el valor de la variable B a la variable A

$A == B$ compara el valor de la variable A con el valor de la variable B

Una comparación produce un resultado CIERTO/FALSO. En C++ el valor falso corresponde a 0 (cero). Las formas de representar el cero o la falsedad son:

Como valor entero. `int value = 0;`

Como caracter. `char false_value = '\\0';`

Como palabra reservada. `bool t = false;`

Expresiones Lógicas

Ejemplos

`(a + b) < c`

`votos > electores`

`a != (x*x + b*x + c)`

`(alto - bajo) == 2`

`(horas*3600) >= (minutos*60 + segundos)`

`numero <= 2`

Expresiones Lógicas: Ejemplo 1

Evaluar la siguiente expresión lógica:

$$x \geq (y+z)$$

$$R1 = y + z$$

$$R2 = x \geq R1 \rightarrow \text{Resultado}$$

Para los valores $x = 3$, $y = 4$, $z = 1$ se tiene

$$R1 = 4 + 1 = 5$$

$$R2 = 3 \geq 5 = \text{falso} \rightarrow \text{Resultado}$$

Expresiones Lógicas: Ejemplo 2

Evaluar la siguiente expresión lógica:

$$\text{contador1} + 1 < \text{contador2}$$

$$R1 = \text{contador1} + 1$$

$$R2 = R1 < \text{contador2} \rightarrow \text{Resultado}$$

Para los valores $\text{contador1} = 3$, $\text{contador2} = 1$ se tiene

$$R1 = 3 + 1 = 4$$

$$R2 = 4 < 1 = \text{falso} \rightarrow \text{Resultado}$$

Expresiones Lógicas: Ejemplo 3

Evaluar la siguiente expresión lógica:

$$\text{num} \% 4 == 0$$

$$R1 = \text{num} \% 4$$

$$R2 = R1 == 0 \rightarrow \text{Resultado}$$

Para el valor $\text{num} = 7$ se tiene

$$R1 = 7 \% 4 = 3$$

$$R2 = 3 == 0 = \text{falso} \rightarrow \text{Resultado}$$

Expresiones Lógicas: Ejemplo 4

Evaluar la siguiente expresión lógica:

$$(a + b - 1) < (x * y)$$

$$R1 = a + b$$

$$R2 = R1 - 1$$

$$R3 = x * y$$

$$R4 = R2 < R3 \rightarrow \text{Resultado}$$

Para los valores $a = 31$, $b = -1$, $x = 3$, $y = 2$ se tiene

$$R1 = 31 + (-1) = 30$$

$$R2 = 30 - 1 = 29$$

$$R3 = 3 * 2 = 6$$

$$R4 = 29 < 6 = \text{falso} \rightarrow \text{Resultado}$$

Operadores Lógicos

Operan sobre operandos (constantes, variables, expresiones) de tipo lógico (cierto o falso) y evalúan a un valor de tipo lógico (cierto o falso).

Operación	Op. Algebraico	Op. C	Exp. C
y lógico	\wedge	<code>&&</code>	<code>a && b</code>
o lógico	\vee	<code> </code>	<code>a b</code>
Negación	\neg	<code>!</code>	<code>!a</code>

Operadores Lógicos

Tabla de la Verdad:

a	b	a && b	a b	!a
cierto	cierto	cierto	cierto	falso
cierto	falso	falso	cierto	falso
falso	cierto	falso	cierto	cierto
falso	falso	falso	falso	cierto

Expresiones Lógicas

Ejemplos

`!(p && q) || s`

`p && q && s`

`(a || b) && (c || d)`

`a > 5 && !(b < c) || 3 <= d`

`(op < min) || (op > max)`

`(edad >= 18) && (edad <= 60)`

`!encontrado`

Precedencia de los Operadores Lógicos

() Sub-expresiones encerradas entre paréntesis se evalúan primero (mayor nivel de precedencia).

!

&&

|| (menor nivel de precedencia)

Precedencia y Asociatividad de Operadores

Operadores	Asociatividad
()	de izq. a der.
! ++ -- - (unario)	de der. a izq.
* / %	de izq. a der.
+ - (binario)	de izq. a der.
< <= > >=	de izq. a der.
== !=	de izq. a der.
&&	de izq. a der.
	de izq. a der.

Expresiones Lógicas: Ejemplo 1

Evaluar la siguiente expresión lógica

$$!(x < 5) \ \&\& \ !(y \geq 7)$$

$$R1 = x < 5$$

$$R2 = y \geq 7$$

$$R3 = !R2$$

$$R4 = !R1$$

$$R5 = R4 \ \&\& \ R3 \quad \rightarrow \quad \text{Resultado}$$

Expresiones Lógicas: Ejemplo 1

Evaluar la siguiente expresión lógica para $x = 3$ y $y = 6$

$$!(x < 5) \ \&\& \ !(y \geq 7)$$

$R1 = 3 < 5 \quad \rightarrow \quad \text{cierto}$

$R2 = 6 \geq 7 \quad \rightarrow \quad \text{falso}$

$R3 = \text{!falso} \quad \rightarrow \quad \text{cierto}$

$R4 = \text{!cierto} \quad \rightarrow \quad \text{falso}$

$R5 = \text{cierto} \ \&\& \ \text{falso} = \text{falso}$

Expresiones Lógicas: Ejemplo 2

Evaluar la siguiente expresión lógica

$$\neg((i > 4) \vee (j \leq 6))$$

$$R1 = i > 4$$

$$R2 = j \leq 6$$

$$R3 = R1 \vee R2$$

$$R4 = \neg R3$$

→ Resultado

Expresiones Lógicas: Ejemplo 2

Evaluar la siguiente expresión lógica para $i = 22$ y $j = 3$

$$\neg((i > 4) \vee (j \leq 6))$$

$R1 = 22 > 4 \rightarrow$ cierto

$R2 = 3 \leq 6 \rightarrow$ cierto

$R3 = \text{cierto} \vee \text{cierto} \rightarrow$ cierto

$R4 = \neg \text{cierto} \rightarrow$ falso

Expresiones Lógicas: Ejemplo 3

Evaluar la siguiente expresión lógica

$$!(a + b == c) \parallel (c != 0) \&\& (b - c >= 19)$$

$$R1 = a + b$$

$$R2 = R1 == c$$

$$R3 = !R2$$

$$R4 = c != 0$$

$$R5 = b - c$$

$$R6 = R5 >= 19$$

$$R7 = R3 \&\& R5$$

$$R8 = R6 \parallel R7 \rightarrow \text{Resultado}$$

Expresiones Lógicas: Ejemplo 3

Evaluar la siguiente expresión lógica para $a = 34$, $b = 12$

y

$c = 8$

$!(a + b == c) \parallel (c != 0) \&\& (b - c >= 19)$

$R1 = 34 + 12 = 46$

$R2 = 46 == 8 = \text{falso}$

$R3 = \text{!falso} = \text{cierto}$

$R4 = 8 != 0 = \text{cierto}$

$R5 = 12 - 8 = 4$

$R6 = 4 >= 19 = \text{falso}$

$R7 = \text{cierto} \&\& \text{falso} = \text{falso}$

$R8 = \text{cierto} \parallel \text{falso} = \text{cierto}$

Resumen

Tipos de datos

Tipo	Notación en C
Entero	int - long – long long – char – short – unsigned
Real	float – double – long double
Caracter	char

Tipo Caracter

Este tipo es capaz de almacenar un carácter del conjunto definido por el lenguaje. Casi universalmente un tipo *char* tiene 8 bits, y en consecuencia puede almacenar 256 tipos de caracteres diferentes.

Tipicamente el conjunto de caracteres es una variante del conjunto ISO-646; por ejemplo el conjunto ASCII.

‘A’ == 65, ‘B’ == 66, ..., ‘Z’ == 90, ‘a’ == 97

¿Cuales caracteres están entre [91, 96]?

Tipos Numéricos

El tipo entero viene de tres maneras:

int (plano), signed int (con signo) y unsigned int (sin signo).

Y de tres tamaños:

short int (corto), int (plano), long int (largo) y long long (mas largo)

Literales:

base decimal: 0 2 63

base octal: 00 02 077

base hexad.: 0x0 0x2 0x3f

Además se pueden añadir sufijos para escribir explícitamente literales sin signo: U ó largos: L. Ejemplo: 4u, 3L.

Tipos Numéricos

Punto Flotante:

Por omisión los punto flotante son del tipo *double*.

1.23 .23 0.23 1. 1.0 1.2e10 1.12e-15

(Note que no existen espacios dentro de cada número, por ejemplo, 1.12 e – 15 no es válido).

Si se quiere escribir un punto flotante del tipo *float*:

3.14159265f 2.997925F

Resumen

Definición y declaración de constantes

Notación algorítmica	Notación en C++
nombre: valor_constante	#define identificador valor
PI: 3.14159	#define PI 3.14159
	const tipo_de_dato identificador = valor
	const float PI = 3.14159;

Resumen

Declaración de variables

Notación Algorítmica	Notación en C++
lista_de_variables: tipo_de_dato	tipo_de_dato lista_de_variables;
a, b, c : entero	int a, b, c;
x: real x = 2.34	float x = 2.34;

Resumen

Hay distintas maneras de observar la salida de una variable de tipo entero:

Int i=425;

printf(“%d %o %x %X”,i,i,i,i);

salida:

425 651 1a9 1A9

Resumen

Tipo Entero

Modificadores	d, i	o, u, x, X
hh	signed char	unsigned char
h	short	unsigned short
l (ele)	long	unsigned long
ll (ele ele)	long long	unsigned long long

Tipo Decimal

Modificador	a, A, e, E, f, F
L	long double

Resumen

Para el tipo decimal:

eE: El argumento double es redondeado y convertido al formanto `[-]d.ddde+-dd`.

fF: El argumento double es redondeado y convertido a la notación decimal `[-]ddd.ddd`

aA: El argumento double es covertido a la notación hexadecimal: `[-]0xh.hhhp[+-]d`. La `p` se usa para separar la mantiza del exponente.

Resumen

Tambien para observar un tipo decimal:

```
float f=12.234;
```

```
printf(“%7.2f  %7.2e\n”,f,f);
```

```
12.23  1.223e+01
```


Operadores sobre Bits

C provee operadores sobre Bits exclusivamente para los tipos enteros (char, unsigned, long, etc.)

Estos son:

& (AND), | (OR), ^ (XOR), << (Corrimiento Izq), >> (Corrimiento Der), ~ (Negacion)

Operadores sobre Bits

Por ejemplo:

0000 0001 & 1111 1111 = 0000 0001

Que es equivalente a:

```
int a=0x01, b=0xff;
```

```
printf("%i",a&b);
```

```
// resulta 1
```

```
printf("%i",alb);
```

```
// resulta 255
```

Operadores sobre Bits

```
printf(“%i”,~a);
```

//resulta -2 (complemento a 1)

Otro ejemplo:

```
char a=0xaa, b=0x55;
```

```
printf(“%i”,a^b);
```

// resulta 255

```
printf(“%i”,a>>4);
```

// resulta en 0xa

Ejercicios

Evaluar cada una de las siguientes expresiones

- a) $(b + (b * b - 4 * a * c)) / (2 * a) \leq 25$
- b) $(x \geq \text{MINIMO}) \ \&\& \ (x \leq \text{MAXIMO})$
- c) $(\text{contador} < \text{MAXIMO}) \ \&\& \ \text{!encontrado}$
- d) $(\text{mes1} < \text{mes2}) \ \text{||} \ (\text{mes1} == \text{mes2}) \ \&\& \ (\text{anyo1} \leq \text{anyo2})$
- e) $\text{anyo} \% 4 == 0 \ \&\& \ \text{anyo} \% 100 != 0 \ \text{||} \ \text{anyo} \% 400 == 0$
- f) $(A - 1) < B \ \text{||} \ A != (B - C) \ \text{||} \ A \geq C$

Ejercicios

Cuáles de los siguientes identificadores no son válidos. Justifique su respuesta.

- a) S_neto
- b) 1A
- c) NotaDefinitiva
- d) _definitiva
- e) masculino_o_femenino
- f) contador2
- g) masculino*femenino
- h) real
- i) float

Ejercicios

Defina las siguientes constantes:

- a) Una constante entera EOF con valor -1.
- b) una constante entera MAXIMO con valor 256.
- c) Dos constantes enteras CIERTO y FALSO con valores 1 y 0, respectivamente.
- d) Una constante real EPSILON con valor 0.0001.

Ejercicios

Declare las siguientes variables:

- a) Dos variables carácter car1 y car2 con valores iniciales 'A' y 'B' respectivamente.
- b) Dos variables enteras i, j.
- c) Una variable real x.
- d) Una variable bandera de tipo lógico con valor inicial cierto.

Ejercicios

Suponiendo declaradas y asignadas las variables

int i=8, j=5;

double x=0.005, y=-0.01;

Indique los valores de las siguientes expresiones:

a) $(3 * i - 2 * j) \% (2 * i - j)$

b) $2 * ((i / 5) + (4 * (j - 3)) \% (i + j - 2))$

c) $-(i + j) / 2$

d) $i++$;

e) $j--$;

f) $(x > y) \&\& (i > 0) \|\ (j < 5)$

g) $(x > y) \|\ (i > 0) \&\& (j < 5)$

Ejercicios

Dadas las siguientes definiciones de constantes

```
#define MIN 1
```

```
#define MAX 5
```

```
#define NUM1 5
```

```
#define NUM2 8
```

```
#define NUM3 -1
```

y la siguiente declaración de variable

```
int num;
```

Escriba las expresiones lógicas que sean verdaderas si

- a) El valor de num se encuentra en el rango definido por las constantes MIN y MAX.
- b) El valor de num es uno de los tres valores constantes NUM1, NUM2 o NUM3.