Boolean Expressions

Lecture 9
Sections 2.11, 4.1, 4.7

Robb T. Koether

Hampden-Sydney College

Fri, Sep 13, 2019

Outline

Boolean Variables and Operators

- A boolean variable may take on one of only two boolean values
  - true
  - false
- There are four standard boolean operators
  - and
  - or
  - not
  - exclusive or (xor)
- A boolean expression is an expression which takes on a boolean value (whether or not its components are boolean).
  - $x > 2$
  - $x \leq 0$ or $x \geq 1$

Logical “And”

- If p and q are boolean expressions, then the expression

  “p and q”

  is true if and only if p is true and q is true.

p	q	p and q
T	T	
T	F	
F	T	
F	F	

Logical “And”

- If p and q are boolean expressions, then the expression

  “p and q”

  is true if and only if p is true and q is true.

p	q	p and q
T	T	T
T	F	F
F	T	F
F	F	F

Logical “Or”

- If p and q are boolean expressions, then the expression

  “p or q”

  is true if and only if p is true or q is true.

p	q	p or q
T	T	
T	F	
F	T	
F	F	

Logical “Or”

- If p and q are boolean expressions, then the expression

  “p or q”

  is true if and only if p is true or q is true.

p	q	p or q
T	T	T
T	F	T
F	T	T
F	F	F

Logical “Not”

- If p is a boolean expression, then the expression

  “not p”

  is true if and only if p is false, i.e., if p is not true.

p	not p
T	
F	

Logical “Not”

- If p is a boolean expression, then the expression

  “not p”

  is true if and only if p is false, i.e., if p is not true.

p	not p
T	F
F	T

Logical “xor”

- If p and q are boolean expressions, then the expression

  “p xor q”

  (exclusive or) is true if p or q is true, but not both.

p	q	p xor q
T	T	
T	F	
F	T	
F	F	

Logical “xor”

- If p and q are boolean expressions, then the expression

  “p xor q”

  (exclusive or) is true if p or q is true, but not both.

p	q	p xor q
T	T	F
T	F	T
F	T	T
F	F	F

Truth Tables

- A truth table for a Boolean expression is a table that shows every possible combination of boolean values of the variables, together with the boolean values of the expression.
- If there are n variables, then there are 2^n combinations of boolean values.

Example: Truth Table

- Truth Table for "p and not (q or r)."

p	q	r	q or r	not (q or r)	p and not (q or r)
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Example: Truth Table

- Truth Table for "p and not (q or r)."

p	q	r	q or r	not (q or r)	p and not (q or r)
T	T	T	T		
T	T	F	T		
T	F	T	T		
T	F	F	F		
F	T	T	T		
F	T	F	T		
F	F	T	T		
F	F	F	F		

Example: Truth Table

- Truth Table for "p and not (q or r)."

p	q	r	q or r	not (q or r)	p and not (q or r)
T	T	T	T	F	
T	T	F	T	F	
T	F	T	T	F	
T	F	F	F	T	
F	T	T	T	F	
F	T	F	T	F	
F	F	T	T	F	
F	F	F	F	T	

Example: Truth Table

- Truth Table for "p and not (q or r)."

p	q	r	q or r	not (q or r)	p and not (q or r)
T	T	T	T	F	F
T	T	F	T	F	F
T	F	T	T	F	F
T	F	F	F	T	T
F	T	T	T	F	F
F	T	F	T	F	F
F	F	T	T	F	F
F	F	F	F	T	F

Outline

The `bool` Data Type

- In C++, there is the **`bool`** data type.
- A **`bool`** object can take on one of only two **`bool`** values.
  - **`true`**
  - **`false`**
- The **`bool`** type is in the integer family.
  - **`true`** is stored as `1`.
  - **`false`** is stored as `0`.
- **`bool`** objects occupy one byte of memory, even though they need only one bit.

The Boolean Operators

- There are three (not four) logical operators in C++.
  - The “and” operator is `&&`
  - The “or” operator is `||`
  - The “not” operator is `!`

Examples

Boolean	C++
(not p) or q	`!p || q`
not (p or q)	`!(p || q)`
(not p) and q	`!p && q`
not (p and q)	`!(p && q)`
p and q or r	`p && q || r`

Relational Operators

- Relational operators are operators that compare objects.
- Equality Operators
  - The “equal to” operator is ==.
  - The “not equal to” operator is !=.
- Order Operators
  - The “greater than” operators is >.
  - The “less than” operator is <.
  - The “greater than or equal to” operator is >=.
  - The “less than or equal to” operator is <=.

Boolean Expressions and Relational Operators

- Typically, boolean expressions are created by using relational operators to compare numerical or other quantities.
- Examples
  - Integer: `count != 0`
  - Floating-point: `x < 123.4`
  - Character: `c >= 'A' && c <= 'Z'`
  - String: `answer == "yes"`
  - Mixed: `count > 0 && sum <= 100.0`
- The operands may be of various types, but the result is always **`bool`**.

Relational Operators

- The equality operators == and != should be defined on all data types since they always make sense.
- The order operators <, >, <=, and >= should be defined only on data types for which they make sense.

Relational Operators

- For which types do the order operators make sense?
  - **`short`**, **`int`**, and **`long`**?
  - **`float`** and **`double`**?
  - **`char`**?
  - `string`?
  - **`bool`**?

Outline

Precedence Rules

- Precedence order from highest to lowest.
  - Post-increment and post-decrement `++`, `--`
  - Logical “not” `!`
  - Unary operators `+`, `-`
  - Pre-increment and pre-decrement `++`, `--`
  - Multiplicative operators `*`, `/`, `%`
  - Additive operators `+`, `-`
  - Insertion and extraction `<<`, `>>`
  - Relational ordering operators `<`, `>`, `<=`, `>=`
  - Relational equality operators `==`, `!=`
  - Logical “and” operator `&&`
  - Logical “or” operator `||`
  - Assignment operators `=`, `+=`, `-=`, `*=`, `/=`, `%=`

Compound Boolean Expressions

Examples

```
x == -y || z != 0                    // ok
x < y && y < z                       // ok
x = b == 0 || a / b == c && !p       // Confusing
```

Improved Examples

```
(x == -y) || (z != 0)
(x < y) && (y < z)
x = ((b == 0) || ((a / b == c) && !p))
```

Compound Boolean Expressions

Examples

```
0 <= x <= 1             // Wrong
0 <= x && x <= 1        // Right
```

- We cannot “chain together” inequalities

A Special Case

- Note that `<<` has a higher precedence than the relational operators `==`, `!=`, `<`, `>`, `<=`, and `>=` and the logical operators `&&` or `||`.
- Therefore, the statement

```
cout << a == 0 << endl;
```

  is illegal because it is interpreted as

```
(cout << a) == (0 << endl);
```

- It must be written

```
cout << (a == 0) << endl;
```

Outline

Example

- Example
  - `BoolOperators.cpp`

Outline

Assignment

Assignment

- Read Sections 2.11, 4.1, 4.7.