CMPSCI 250: Introduction to Computation

Lecture #2: Propositions and Boolean Operators
25 January 2012

Propositions and Boolean Operators

- What is a Proposition?
- Java Booleans
- Boolean Operators and Compound Propositions
- AND, OR, NOT, and XOR
- Implication and Equivalence
- Tautologies

What is a Proposition?

- A **proposition** is a statement that is either true or false.

- In mathematics we want to reason about statements like “x = 5” or “these triangles are congruent” without knowing whether they are true or false. We say things like “if x = 5, then x^2 = 25” or “a natural number is the same thing as a non-negative integer”.

- In computing we reason with **assertions** about a program, like “if this method terminates, the value of `i` is positive”. Ultimately we’d like to say “if the input is as specified, then the output is as specified”, meaning “the program is correct”.

- What isn’t a proposition? Questions, commands, statements without meaning, or paradoxes like “This statement is false”.

Java Booleans

- Java has a `boolean` primitive data type, and a `boolean` must have either the value `true` or the value `false`.

- We use booleans in the conditions for `if` or `while` statements -- if we write "`if (x > 4) y = 5;`", then the statement `y = 5` will be executed only if the boolean value `x > 4` evaluates to `true`.

- The operators ==, `!=`, >, >=, <, and <= create `boolean` values from values of other types. We often write methods that return `boolean` values, or use existing `boolean` methods like `equals`.

- You may think of a "proposition" as any statement that *could* in principle be modeled by a boolean variable. Of course we will have propositions that talk about physical or mathematical objects as well as data in a computer.

Boolean Operators and Compound Propositions

- A **compound proposition** is a proposition that is made up from other propositions, called **atomic propositions**, using **boolean operators**.

- If I say "you must have MATH 132, and either CMPSCI 187 or ECE 242", we can define three atomic propositions and write this as a compound proposition. If x is "you have MATH 132", y is "you have CMPSCI 187", and z is "you have ECE 242", then my statement can be rephrased as "x, and either y or z". Symbolically, we will write this as "$x \wedge (y \vee z)$".

- If x, y, and z are any three booleans, the truth of $x \wedge (y \vee z)$ depends on which of x, y, and z are true. In Java, if `x`, `y`, and `z` are boolean variables, we can write the expression `x && (y || z)`, and this represents $x \wedge (y \vee z)$.

- The rules for working with booleans and boolean operators are called the **propositional calculus** and we will start learning them now.

AND, OR, NOT, and XOR

- If x and y are any two propositions, their **conjunction** $x \wedge y$ is the proposition that is true if and only if *both* x and y are true. We read it “x and y”. The Java operators `&` and `&&` both compute the value of a conjunction -- we usually use `&&` which only evaluates the second argument if it is needed.
- The **disjunction** of x and y is written $x \vee y$, read “x or y”, and is true if *either* x or y is true, *or both*. In Java the disjunction is computed with `|` or `||`.
- The **negation** of x is written $\neg x$, read “not x”, and is true when x is false and false when x is true. In Java the negation operator is `!`.
- The **exclusive or** of x and y is written $x \oplus y$, read “x exclusive or y” or “x or y, but not both”, and is true if one of x and y is true and the other false. In Java we can write `x ^ y` to compute the exclusive or of x and y.

Implication and Equivalence

- The last two boolean operator we will use are **implication** and **equivalence**. These are important in mathematics because they express two relationships between propositions that we frequently want to prove.

- The implication $x \rightarrow y$ is read “if x, then y” or “x implies y”. It is true if either x is false or y is true. Equivalently, it is true *unless* x is true and y is false. This may or may not correspond with your understanding of the English phrasing, but we need to fix this interpretation as a rule.

- Normally in mathematics we want to make some **assumptions** and prove that some **consequences** must be true if the assumptions are true. This is an implication.

- In the book we have Bertrand Russell’s argument that with the assumption “0 = 1”, we can prove the proposition “I am Elvis Presley”.

Equivalence

- Two boolean values are **equivalent** if they are both true or both false. If x and y are propositions, $x \leftrightarrow y$ is the proposition that x and y are equivalent. We can write this this in Java as `x == y`.

- We are often interested in the equivalence of two compound propositions with the same atomic propositions. For example, "$x \rightarrow y$" and "$\neg x \vee y$" are equivalent. How do we know this? They are each true in three of the four possible cases -- they are false only if x is true and y is false. They have the same **truth tables**, as we will soon see.

- As in Java, we have rules for precedence of operations. Negation is first, then the operators $\wedge$, $\vee$, and $\oplus$, then the operators $\rightarrow$ and $\leftrightarrow$. So we can write the equivalence above as the single compound proposition $(x \rightarrow y) \leftrightarrow \neg x \vee y$.

Tautologies

- This compound proposition $(x \rightarrow y) \leftrightarrow \neg x \vee y$ is true in all four possible situations of truth values for x and y, so it is *always true.* We call such a compound proposition a **tautology**. In the next lecture we will learn a systematic method to show that a compound proposition is a tautology, by checking all the possible combinations of values of its atomic propositions.

- Next week we will see how to use particular tautologies as rules, chaining them together to verify larger tautologies without having to check all the possible cases. If there are many atomic propositions, this may be the only feasible way to verify the tautology.

- Our most common tasks with booleans in mathematics turn out to be verifying that particular implications or equivalences are tautologies. Verifying $x \rightarrow y$ means that if we assume x, we may conclude y. Verifying $x \leftrightarrow y$ means that x and y are in effect the same compound proposition.