Logic & Logical Thinking

Paul S. Wang, Sofpower.com

May 27, 2023

Digital computers are logic machines. They use *bits* to store information. A bit is nothing but a switch with two states, on and off. A bit can represent a digit, `1` or `0`. Or a bit can represent a truth value, *true* or *false*. We will describe the close relationship between logic and computers.

Digital circuits in modern computers are built with logic gates that perform computations on truth values. Boolean algebra deals with computations on truth values. Logic conditions and implications are used in software to control program execution. These information can help us understand digital technology and, at the same time, provide a background for logical thinking.

The ability to think and reason logically is important in general but critical for computational thinkers. Thus, also covered here is how to improve and sharpen our own ways for logical thinking.

Materials here can help you think logically and make you a better computational thinker as well. This post is part of our *Computational Thinking* (CT) blog where you can find many other interesting and useful articles.

Boolean Algebra

Logic operations are at the base of digital computers. Using a bit to represent 0 and 1 and treating 1 as *true* and 0 as *false*, all the basic logic operations `AND`, `OR`, `NOT`, `XOR`, `NAND`, `NOR`, and `XNOR` can be performed by logic gates. These operations form the foundation of all other computations in a computer, as we can see in our other article "*NAND Rules The World*".

The word *algebra* comes from Arabic al-jebr meaning "reunion of broken parts". Elementary algebra, the kind we learn in middle school, deals with real numbers and symbols. The symbols stand for variables and unspecified numbers. Boolean algebra, introduced by George Boole in 1854, deals with truth values, *true* and *false* or 1 and 0, instead of numbers. Variables in Boolean algebra may take on either of the two values.

Boolean algebra has the following *basic* operations and operators.

- *Conjunction*—Denoted `A` $\wedge$ `B`, `A AND B`, `A & B`, or `A` $\bullet$ `B`; the value of `A AND B` is *true* only if both `A` and `B` are *true*
- *Disjunction*—Denoted `A` $\vee$ `B`, `A OR B`, `A || B`, or `A + B`; the value of `A OR B` is *true* if at least one of `A` and `B` is *true*
- *Negation*—Denoted ¬`A`, `NOT A`, `!A`, or $\overline{\texttt{A}}$; the value of `NOT A` is *true* if `A` is *false* and is *false* otherwise

Boolean algebra deals with expressions involving these operators, their properties, and manipulations. As such, it is very useful in the study and design of digital circuits.

Expressions and Laws

Here, in our introduction to Boolean algebra, we will use the values `0` and `1`, and the operators $\bullet$, $+$, and $^{-}$.

Let a, b, and c be Boolean variables. The following *laws* hold in Boolean algebra.

- Simplification laws: $a \bullet a = a,\ a + a = a,\ \overline{(\overline{a})} = a,\ 0 + a = a,\ 0 \bullet a = 0,$ $1 + a = 1,\ 1 \bullet a = a$
- Communicative laws: $a \bullet b = b \bullet a,\ a + b = b + a$
- Associative laws: $a \bullet (b \bullet c) = (a \bullet b) \bullet c,\ a + (b + c) = (a + b) + c$
- Distributive laws: $a \bullet (b + c) = (a \bullet b) + (a \bullet c),\ a + (b \bullet c) = (a + b) \bullet (a + c)$
- Absorption laws: $a \bullet (a + b) = a,\ a + (a \bullet b) = a$ (The variable b is absorbed as if it is not there.)
- Negation laws: $a \bullet \overline{a} = 0,\ a + \overline{a} = 1$
- De Morgan's laws: $\overline{a \bullet b} = \overline{a} + \overline{b},\ \overline{a + b} = \overline{a} \bullet \overline{b}$

The three logical operations `AND`, `OR`, and `NOT` are basic in that they can be used to produce any truth table with up to 2 inputs and one output. For example,

`a XOR b` $= (a + b) \bullet \overline{(a \bullet b)}$

Decision Making

When drawing a flowchart or specifying an algorithm, we often need to have test conditions. Depending on the yes/no answer of a test, a procedure may take a different path through its steps (Figure 1). In logic and in programming, a function that produces a result which is either *true* or *false* is known as a *predicate*. Programming languages usually provide *relational operators* as predefined predicates. Table 1 lists relational operators in JavaScript that compare numerical quantities and produce a truth value.

Figure 1: Decision Making

Table 1: JavaScript Relational Operators

Operator	Meaning
`==`	Equal to
`!=`	Not equal to
`<`	Less than
`>`	Greater than
`<=`	Less than or equal to
`>=`	Greater than or equal to

Usually, a bit pattern with all 0s is treated as *false* and any other value is treated as *true*. This makes sense because 0 is *false* and anything that is not 0 is *true*. An immediate result of this convention is that any function that returns a value can be treated as a predicate. Programming languages also provide *logical operators* to perform Boolean operations on truth values. Table 2 lists JavaScript logical operators. Like most other programming languages, JavaScript adopted relational and logical operators from C/C++.

Table 2: Logical Operators

Operator	Meaning
`&&`	`AND`
`||`	`OR`
`!`	`NOT`

As an example, let's use the preceding notations to define a predicate `rhNormal` which takes an input relative humidity reading and determines if it is in the normal comfortable range (between 50% to 60%) for people.

Algorithm `rhNormal`:
Input: Integer percentage `rh`
Output: Returns `0` (*false*) or nonzero (*true*)

1. If `(rh > 60 || rh < 50)` then return `0`
2. Return `1`

Alternatively, we can use the following.

Algorithm `rhNormal`:
Input: Integer percentage `rh`
Output: Returns `0` (false) or nonzero (*true*)

1. If `(rh >= 50 && rh <= 60)` then return `1`
2. Return `0`

Automatic control often means keeping values of certain parameters within allowable ranges. A humidity control system may call `rhNormal` periodically and decide to start/stop the increasing or decreasing the humidity. Such computer controls are commonplace. You'll find them in automobile cruise control systems, anti-lock brake systems, GPS navigation systems, autopilot for airplanes, and so on.

Conditions and Implications

As you can imagine, when devising an algorithm, making the right decisions on which next step to take is critical. Typically, we use

`if` *predicate* `then` $action_1$ `else` $action_2$

to indicate such decisions. If the *predicate* evaluates to *true* $action_1$ is taken. Otherwise, $action_2$ is taken. The `else` part is usually optional in the notation.

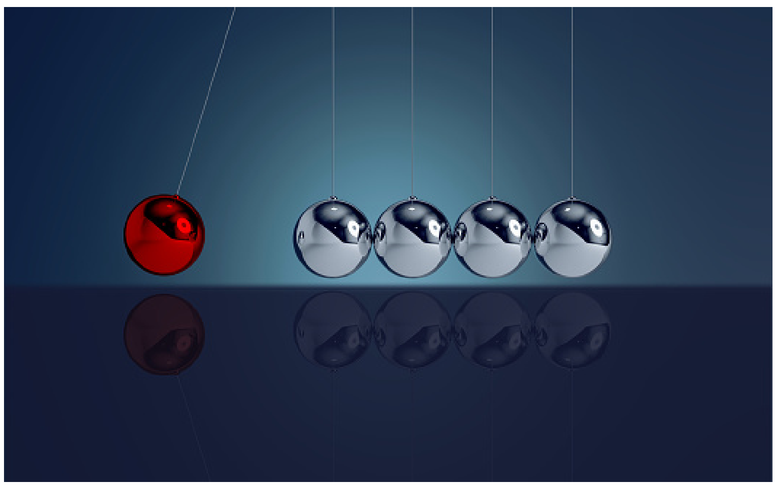

Figure 2: Implication

Correctness of an algorithm depends on using the right *implications* Figure 2). An implication is a logical statement commonly given in these forms.

- p implies q, or $p \implies q$
- if p than q
- q if p

where p is a *premise* and q is a *conclusion*. Let's look at an algorithm that compares two input numbers `x` and `y` and returns one of the following

1. A positive number if `x` is larger than `y`
2. A negative number if `x` is smaller than `y`

3. A zero if `x` is equal to `y`

Algorithm `numberCompare`:
Input: Number `x`, number `y`
Output: Returns `1`, `0`, or `-1`

1. If `x > y`, then return `1`
2. If `x < y`, then return `-1`
3. Return `0`

In algorithm `numberCompare`, note the implications

"control flow reaching Step 2" $\implies$ "`x <= y`"

"control flow reaching Step 3" $\implies$ "`x == y`"

Now think about why `numberCompare` can be implemented simply as "`return x - y`".

Necessary vs. Sufficient Conditions

Given the implication $p \implies q$ (p being *true* causes q to be *true*) then the following statements are true.

- p is a *sufficient condition* for q, namely p being *true* guarantees that q is *true*.
- q is a *necessary condition* for p, namely q must be *true* for p to be *true*. Also, if q is *false* then p is *false* as well. Thus, the implication $p \implies q$ is logically the same as the implication $\overline{q} \implies \overline{p}$.
- If p is *false*, the implication says nothing about q.
- If q is *true*, the implication says nothing about p.

For example, the implication "If x is a woman then x is a person" certainly does not mean "If x is a person then x is a woman". Nonetheless, if x is not a person then x can not be a woman.

Similarly, "It is a river $\implies$ water flows in it" does not mean if water flows in it then it is a river. In fact, it could be a water hose or a drain pipe. But, if water does not flow in it then it is not a river.

And "If n is a multiple of 8 $\implies$ n is an even number" does not mean if n is even then it is divisible by 8. And "If a person is over 30 years old then the person is an adult" does not mean an adult is over 30.

Finally, "A good computer programmer thinks logically" does not mean that anyone who thinks logically is a good programmer. The person must have other training as well. Yet, it is definitely the case that without logical thinking one cannot be a good programmer. The same can be said of good computational thinkers.

To summarize, a sufficient condition may not be necessary and a necessary condition may not be sufficient.

However, if we have both implications $p \implies q$ and $q \implies p$, then q is a *necessary and sufficient* condition for p. Likewise, p is a *necessary and sufficient* condition for q. Alternatively, we say p if and only if q or simply $p \iff q$. In such a case, p and q are both *true* or both *false*. For example, a person may vote in a United States election if and only if the person is a United States citizen, at least 18 years old, and not a convicted felon.

Don't hesitate to study the materials over again. Make logic your own natural mental tool and it will help immensely in whatever you do. However, once logic is natural to you, don't assume others are the same. In fact, it is a good bet to assume otherwise. Your being logical certainly does not imply that everyone else is. Because we need to work with others to achieve many tasks, guarding against falling victim to less than logical thinking on the part of others would be wise indeed.

Logical Thinking

Logical thinking is a cognitive process that involves reasoning and problem-solving in a systematic and rational manner. It is the ability to analyze and evaluate information, identify patterns, and draw logical conclusions based on evidence and facts.

But this is easier said than done. People can easily become emotional in many situations, including being optimistic, pessimistic, wishful, fearful, personal, and so forth. These feelings can negatively impact objective and evidence-based reasoning.

Logical thinking is an essential skill for decision-making, critical thinking, and problem-solving in various fields such as science, engineering, mathematics, and computer science. It is a skill computational thinkers should acquire and sharpen.

Key points and methods of logical thinking include:

Figure 3: Attention to Details

- Identifying and defining the problem or situation
- Gathering and analyzing information and data paying attention to the 5W and 1H (what, why, when, where, who, and how
- Looking at the whole picture as well as paying attention to details (Figure 3)
- Anticipating potential problems and difficulties by asking “what if” questions
- Identifying patterns and relationships
- Formulating hypotheses and making predictions

- Testing and evaluating hypotheses using evidence and facts
- Drawing conclusions and making decisions based on the results.

Cognitive Biases

A *cognitive bias* is a systematic error in thinking that can distort our perception, judgment, and decision-making. Such biases are mental shortcuts that our brains use to simplify complex information and make quick decisions, but they can also lead to errors in judgment or decision-making.

Cognitive biases can affect our perception, attitude, behavior, and reasoning and negatively impact logical thinking. Let's list some common cognitive biases:

- **Herd mentality bias**: Also known as the bandwagon effect, it is the tendency to follow the crowd or majority. Of course, for any proposition, getting more votes does not prove it right or wrong.
- **Confirmation bias**: The tendency to seek out information that confirms our pre-existing beliefs and ignore information that contradicts them.
- **Availability heuristic**: The tendency to rely on easily available or memorable examples when making decisions or judgments. For example fear of flying but not of driving.
- **Anchoring bias**: The tendency to rely too heavily on the first piece of information encountered when making decisions. For example, if you first see a bicycle that costs $1,200-–then see a second one that costs $100–you're prone to see the second bicycle as cheap or not well built.
- **Hindsight bias**: The tendency to believe that an event was predictable or easily explainable after it has occurred.
- **Overconfidence bias**: The tendency to overestimate our abilities, knowledge, or accuracy of beliefs.
- **Gambler's fallacy**: The belief that past events can influence the probability of future events, even when they are independent. For example, thinking that a coin is less likely to turn up heads after it had turned up heads five or ten times in a row.

- **Self-serving bias**: The tendency to attribute positive outcomes to our own abilities and negative outcomes to external factors.

- **Halo effect**: The tendency to form a positive overall impression of a person or thing based on one specific trait or characteristic. For example judging a book by its cover, or a person by the car he or she drives.

- **Negativity bias**: The tendency to focus more on negative information than positive information.

- **Bandwagon effect**: The tendency to adopt certain beliefs or behaviors because many others are doing so.

  These are just a few examples of the many cognitive biases that can influence our thinking and decision-making. It's important to be aware of these biases and to consciously work to overcome them in order to think more rationally and make better decisions.

Logical Fallacies

Logical fallacies are errors in reasoning or logical deduction that can lead to incorrect or unsupported conclusions. Logical thinking involves identifying and avoiding fallacies, which can help one arrive at more accurate and well-supported conclusions.

Here is a list of some common logical fallacies:

- **Ad hominem fallacy**: Attacking the character or motives of a person making an argument, rather than addressing the substance of the argument itself. This is the most commonly found logic mistake among ordinary people. Meantime the reverse, "appeal to authority" is also a common fallacy (Figure 4).

- **Straw man fallacy**: Misrepresenting or exaggerating someone else's position or argument in order to make it easier to attack. The distorted position is known as a *straw man* which is set up to be easily destroyed (Figure 5).

- **False dichotomy fallacy**: Presenting only two options as if they are the only possibilities, when in fact there are more.

Figure 4: Ad hominem (personal attack fallacy)

- **Slippery slope fallacy**: Suggesting that a particular action will inevitably lead to a series of negative consequences, without providing sufficient evidence to support this claim.
- **Appeal to authority fallacy**: Relying on the opinion of an authority figure as evidence for an argument, without providing sufficient evidence to support their claim.
- **Hasty generalization fallacy**: Drawing a general conclusion based on insufficient or unrepresentative evidence.
- **Post hoc fallacy**: Assuming that one event caused another event simply because it occurred before it.
- **Circular reasoning fallacy**: Using the conclusion of an argument as evidence to support the premises of that same argument.
- **False cause fallacy**: Assuming that because two events occur together, one must have caused the other.
- **Begging the question fallacy**: Assuming the truth of the very thing that one is trying to prove in the argument.
- **Appeal to emotion fallacy**: Relying on emotions or feelings to support an argument, rather than providing evidence or logical reasoning.

Figure 5: A Strawman

- **Bandwagon fallacy**: Suggesting that something is true or good simply because many people believe or support it.
- **Ad ignorantiam fallacy**: Arguing that something is true simply because it has not been proven false, or vice versa.
- **False analogy fallacy**: Drawing a comparison between two things that are not truly comparable in order to make an argument.
- **Red herring fallacy**: Introducing irrelevant information or arguments into a discussion in order to distract from the main point.

You Can Do It!

Computers are logic machines and it follows logic and instructions without bias. We can sharpen our logical thinking by seeing the rigorous ways computers and algorithms apply logic. Fact is that logical thinking is a vital skill for success in many technical and non-technical areas, including and especially in our daily lives. It is critical for decision-making, problem-solving, critical thinking, effective communication, and creativity. It is a fundamental skill that can benefit everyone both personally and professionally. Of course, logical thinking is especially important for computational thinkers.

For some, acquiring logical thinking can be challenging because of overcoming cognitive biases and avoiding logic fallacies. Here are some sugges-

Figure 6: Yes You Can!

tions for sharpening your logical thinking.

- **Increase experience and practice**: Seek out opportunities to engage in logical thinking, for example joining a debate team or planning an event such as a birthday party or wedding reception. Drawing flowcharts is a good activity. Also solving puzzles and playing strategy games can help (Figure 7).
- **Correct cognitive biases**: Learn to recognize your cognitive biases and consciously work to overcome them.
- **Arrest emotional reasoning**: Catch yourself when emotions interfere with rational and logical thinking. By learning to recognize when emotions are influencing one's thinking, an individual can consciously work to separate their emotions from their logical reasoning.
- **Keep an open mind**: Consider alternative perspectives and think critically. By learning from others with different backgrounds, experiences, and viewpoints, one can improve the situation.
- **Think critically**: Think clearly and pay attention to the logic in reasoning. Have a hard-rto-convinc mindset toward statements or arguments by others. Find logic fallacies in their reasoning so you avoid them in yours.

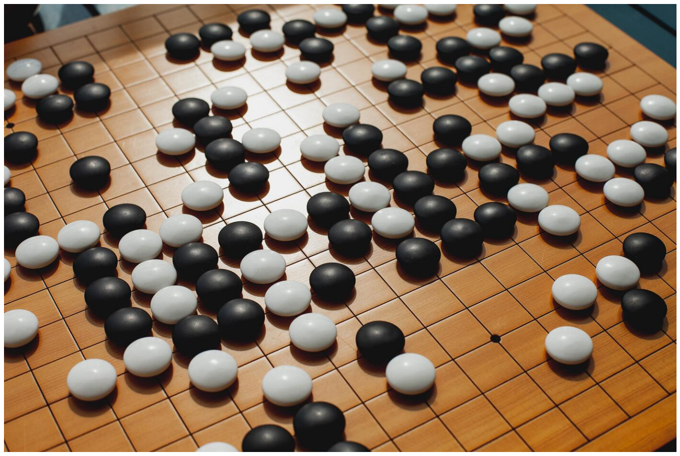

Figure 7: The Game GO

With an understanding of logic, logical reasoning, potential cognitive biases, and logical fallacies, you can appreciate the importance of logical thinking. And with increasing exposure and more practice, you we can become good at logical thinking which is fundamental for becoming a computational thinker.