What is science?

According to the definition: *Science* is understood as a provable and fact-based system of the objective relationships between *nature*, *society*, and *human thought*.

However, science is not only a set of theories stored in textbooks. It is a dynamic process of exploration and innovation that directly shapes technology, industry, and daily life. Science is the reason we can design sustainable energy systems, create AI-powered applications, or even send rockets into space.

Science is not just a collection of knowledge, but an ongoing **discovery process**. It aims to discover new information, facts, and answers about our world or the universe, while also solving real-world engineering problems.

Science, among our historically established forms of social consciousness, is distinguished and emphasized by the following characteristics.

- It uses high-level concepts and logical tools to formulate or express broad, general, or universal **principles**, laws, or axioms. (e.g. gravity, **axioms**, Maxwell's equations)
- It can describe the objective **conditions** under which these principles or laws will take effect.
- It provides logical methods and computational tools that enable us to calculate, simulate, and predict **results** under specific circumstances.

From a modern engineering viewpoint, these features mean that science is not abstract—it is applied, testable. Every time an engineer uses mathematical models to optimize a production line, simulates a digital twin of a factory, or analyzes big data to predict outcomes, they are applying these scientific principles.

Using the three general aspects — **principles**, **conditions**, and **results** — we can categorize every *scientific problem* into the following groups.

Inductive Sciences

Induction: the physical conditions are known, just like the results, and we are seeking for the general principle. this is the classical type of experimental physics problem.

flowchart TD E((Results)) F((Conditions)) T((Principles)) E-->T F-->T

Explanation: *Induction* is probably the most important logical method used by scientists to draft new *theories* or *principles*.

Induction is a generalizing method, which means that we seek a universal or general law from a given set of data with fixed conditions. A well-known example of this method is the Mendelian laws of inheritance or Kepler's law of planetary motion

The biggest problem with this method is whether we have (or have yet to) carry out *sufficient observations* to arrive at a general conclusion.

In natural sciences, we are always dealing with partial induction. The more experiments we do, the more confident we will become and the better our chances of understanding the connections.

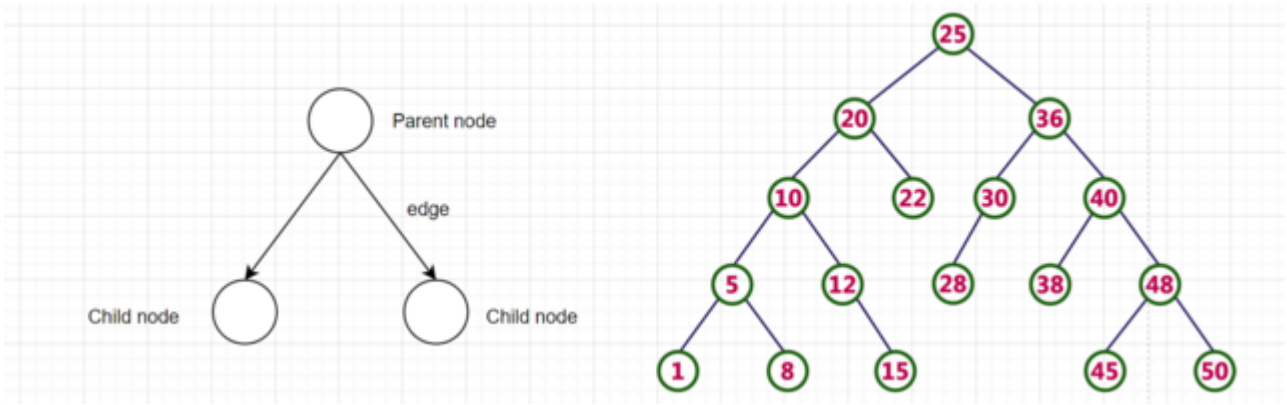
Our confidence is based on the premise that nature itself behaves consistently.

The so-called complete induction, which is used in mathematical problems, will bypass any kind of these problems.

Remark: Legislative processes are based on an inductive method that analyzes social problems and their causes and makes new laws as a conclusion.

Example 1: In **information technology**, mathematical induction can be applied to many areas, including algorithm analysis and data structures. One typical example is proving the correctness of algorithms or the properties of data structures like trees. Here's an example from **binary trees**:

Problem: Prove by induction that a **binary tree** with \(n\) nodes has exactly \(n-1\) edges. A *binary tree* is a hierarchical data structure where each node has at most two children (left and right). An *edge* is the connection between a *parent node* and a *child node*.



Steps for Mathematical Induction:

1. **Base Case** \((n = 1)\): For a binary tree with just one node (the root), there are no edges because there are no child nodes. Therefore, the number of edges is: $$ 0 = 1 - 1 $$
2. **Inductive Hypothesis**: Assume that for any binary tree with \(k\) nodes, the number of edges is \(k-1\).
3. **Inductive Step**: We must prove that if the statement holds for a binary tree with \(k\) nodes, then it also holds for a binary tree with \(k+1\) nodes.
   Suppose we add one more node to the binary tree, bringing the total number of nodes to \(k+1\). When we add this node, we also add exactly one edge connecting the new node to an existing node in the tree (either as a left or right child of a parent node).

   By the inductive hypothesis, the tree with \(k\) nodes has \((k - 1)\) edges. Adding one more node introduces one additional edge, so the number of edges in the tree with \((k + 1)\) nodes is: $$ (k-1) + 1 = k $$ This matches the formula for the number of edges in a tree with \((k +

1)\) nodes, which should be \((k-1) + 1 = k\).

Example 2: Sum of consecutive natural numbers. **Claim**:

$$ 1 + 2 + \cdots + n = \frac{n(n+1)}{2}. $$

Base case (n=1):

$$ 1 = \frac{1 \cdot 2}{2} = 1, $$

so the statement holds.

Inductive step:

Assume the formula is true for \(n = k\):

$$ 1 + 2 + \cdots + k = \frac{k(k+1)}{2}. $$

Then for \(n = k+1\):

$$ 1 + 2 + \cdots + k + (k+1) = \frac{k(k+1)}{2} + (k+1). $$

Simplify:

$$ \frac{k(k+1)}{2} + (k+1) = \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2}. $$

Thus, the formula also holds for \(n = k+1 \). Thus, by mathematical induction, the formula holds for all \(n\in\mathbb{N}\)

https://en.wikipedia.org/wiki/Mathematical_induction

Deductive Sciences

Deduction: the general principles and conditions are known and we seek several expected results. This is a typical example of theoretical physics.

Explanation:

Deduction must solve the initial, boundary or edge requirements of various differential equations.

This deductive method is the core of the so-called pure mathematics, where the theories are built from deductive results explicitly derived from axioms (just like in Euclidean geometry).

Then they take the results as principles if there a logically valid inference chain which has been derived from the axioms.

These deductive results will provide solid proofs that inductive methods can never achieve (assuming that the axioms are consistent).

These deductive results will provide so solid proofs that can never be achieved by inductive methods. (assuming that the axioms are consistent)

flowchart TD E((Results)) F((Conditions)) T((Principles)) T-->E F-->E

However, Deductive Logic cannot confirm whether a statement in the chain was true (or not).

Logic can only state that the results will be true if the premises are true (and consistent) and the arguments are logically correct.

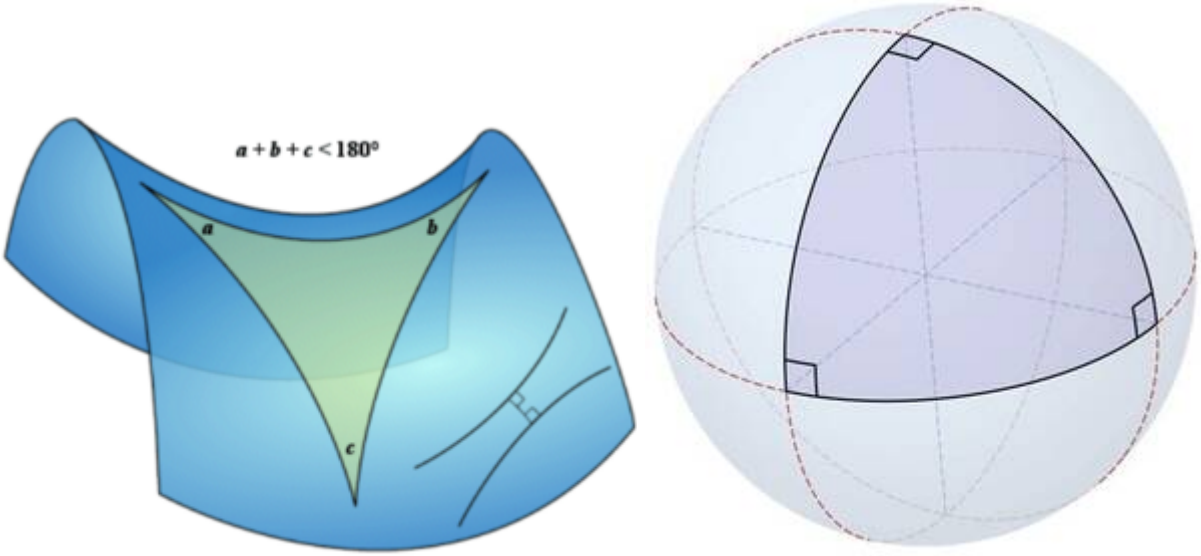
Example:

János Bólyai (1802–1860) – a famous Hungarian mathematician – wrote this famous sentence to his father:

'*I have created a new and different world from scratch*.' He reached the conclusion that by changing the Fifth Axiom of the Euclidean principles, a new world could be created.

Fifth Axiom (Parallel Postulate): Given a straight line and a point not on the line, there is *exactly one straight line* that can be drawn through the point that is parallel to the given line.

People in his era were not really convinced by his theories, but today we have already known that our world is one of those which are based on different Euclidean geometrical principles.



The quote from Niels Bohr, *"In the Institute, we only have one piece of experimental equipment: a ping-pong table"* is a good example of reductive reasoning in science. It suggests that groundbreaking discoveries can be achieved not through extensive experimental setups but rather through the adjustment of theoretical laws and conditions. Bohr highlights the power of thought experiments and abstract reasoning, emphasising that manipulating underlying principles can lead to new insights without always needing physical experimentation.

Reductive Sciences

The main principles and the results are known and we are seeking the **appropriate conditions** which can realize our goals.

Explanation: These types of tasks are typical examples of technical sciences. Unfortunately, the solution cannot be inverted from the end results; therefore, there can be an infinite number of terms that can get us to the known results. In this case, we have to accept a few possibilities (or, more usually, only one). We usually get to this term in **heuristic ways**.

We can face another interpretation of reduction in classifying elementary scientific problems (the so-called 'Trinity' of sciences).

In this case our main task is to reduce the number of possible solutions in a reasonable way.

flowchart TD E((Result)) F((Condition)) T((Principles)) T-->F E-->F

Example: Database Query Optimization

When working with databases, especially large-scale systems, an important task is to optimize database queries to ensure they run as efficiently as possible. The main goal is already clear: execute a query in the shortest time possible while minimizing resource consumption (CPU, memory, disk usage). However, there are many possible ways to structure a query, and each structure might result in different performance levels depending on the database engine, indexing, and hardware setup.

Here's how the concept of **reductive science** applies in this case:

1. **Main Principles Known:**
    1. The query must retrieve specific data, based on given conditions (e.g., filtering, joining tables, sorting).
    2. The performance depends on factors like indexing, table size, query structure, and features of the database engine.
    3. The result of the query must remain the same regardless of the optimization.
2. **Seeking Appropriate Conditions:** The task is to optimize the query by structuring it to minimize response time and resource usage.
    1. **There's no single "perfect" solution** for optimizing the query, because different query structures can yield the same result, but with varying degrees of performance based on the specific context (e.g., hardware, data distribution, and load).
    2. Additionally, the same query might perform differently on different database engines (e.g., MySQL vs. PostgreSQL), and therefore there can be an infinite number of ways to structure a query that achieves the same end result.
3. **How to reduce the number of conditions?** The database administrator (DBA) or developer uses **heuristic methods** like:
    1. Query profiling tools (e.g., EXPLAIN in SQL) to examine how different query structures perform.
    2. Applying **best practices** like indexing the right columns, minimizing nested queries, and using joins effectively.
    3. By profiling and tweaking different versions of the query, the developer reduces the number of possible query structures to a few that perform optimally in the given context.

The *reductive approach* in database query optimization involves narrowing down many possible solutions (query structures) to a few practical ones. The solution can't simply be inverted from the final result (i.e., retrieving the data); instead, developers use heuristics, profiling, and experience to eliminate inefficient options and find the most effective query structure for their specific environment.

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:techcomm:information_-_basics:sciences

Last update: **2026/09/21 12:25**