The New Turing Omnibus

A. K. Dewdney

December 20, 2020

Contents

1 Alagorithms: Cooking Up Programs

A program specifies in the **exact syntax** of some programming language the computation one expects a computer to perform. The syntax is precise and unforgiving. The slightest error in the program as written may cause the computation to be in error or may halt altogether. The reason for this situation seems paradoxical on the surface: It is relatively easy to design a system that converts rigid syntax to computations; it is much harder to design a system that tolerates mistakes or accepts a broader range of program descriptions.

2 Finite automata: The Black Box

It occasionally happens in industrial, military or educational settings that one is presented with a piece of electronic hardware whose exact function is uncertain or unknown. One way of discovering how the device works is to take it apart, piece by piece, and deduce its function by analysing the components and their interconnections. This is not always possible, however, nor is it always necessary. Given that the mystery machine has both input and output facilities, it may be possible to discover what it does without ever taking it apart. Since its appearance gives no clue about its function, we call it a *black box*.

3 Systems of Logic: Boolean Bases

In an age of computers and automation, almost every electronic device one can name incorporates at least one boolean function. For example, many current models of automobile will emit a high-pitched whine, buzz or other disturbing noise until their drivers fasten their seat belts. Such a device realises a boolean function of two variables.

4 Simulation: The Monte Carlo method

In the quest to understand the many systems that comprise the modern world we turn increasingly to computer simulation. Whether the system is natural or artificial, frequently one or more of its components have such complex behaviour that the only feasible approach to approximating such behaviour is to assume that it is random.

5 Gödel's Theorem: Limits on Logic

In the early 1930's, Kurt Gödel, a German mathematician, attempted to show that predicate calculus was complete - that one can obtain mechanically (in principle, at least) a proof of any true formula expressed in that calculus. His failure to do this was crowned by the discovery that the task was impossible.

6   Can machines think?

Turing addressed the question "Can machines think?" in his 1950 paper *Computing machinery and intelligence.*

7   But what is LaTeX good for?

We're using this LaTeX document to demonstrate some of its key strengths that you will find useful during and after University:

1. LaTeX can quickly create pdf files

2. LaTeX uses professional typesetting

3. LaTeX documents can be more legible, clear, and visually appealing to the reader than those created with word processing software