

Flowcharting and Designing

P. Reany

December 6, 2025

Truth, like oil, will in time rise to surface.

— Charlie Chan

1 Introduction: A Couple Memes Denied

A *meme*¹ is a notion that eventually gets spread through a population by its perceived merit. There are two notions of cognitive significance in computer science, in particular, that I want to emphasize that are popular by virtue of their effectiveness in problem solving, namely, the *top-down design with stepwise refinement*² and the use of flowcharting. These are basic tools of every computer science student, at least when I took the introductory classes decades ago.

Unfortunately, these cognitive notions in computer science have shown themselves difficult to cross the Computer Science boundary into related disciplines, especially into mathematics.

Flowcharting

Let's begin with flowcharting. Sure, every science uses flowcharts these days. But that's not my point. My point is that flowcharting is not yet rigorously taught to students as a general purpose, everyday technique of problem solving in science and mathematics. For examples: Why aren't geometry students required to present their geometry proofs in flowchart form on occasion to prove that they can do so? Why aren't chemistry students required to demonstrate in a flowchart the procedure for a titration experiment? These are algorithmic uses of flowcharts, and fairly straightforward to understand.

¹This term was introduced by Richard Dawkins in 1976.

²I presented this meme in some detail in the last paper of this series.

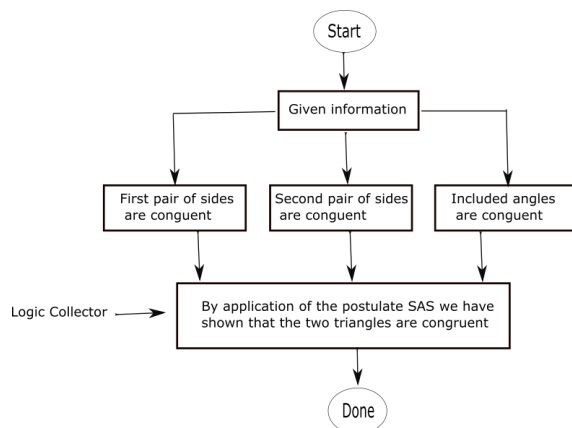


Figure E1. This graphic reveals the non-linearity of some geometry proofs by laying bare the parallelism in the subproofs involved.

For a case in point, consider a typical homework problem in high school geometry: If one is to use the SAS postulate to prove that two triangles are congruent (reference Figure E1), say, one has three subproofs to prove, i.e., that three corresponding parts are congruent. Generally speaking, each of these subproofs is independent of the others and a flowchart can make that obvious by diagramming them in parallel, not in series. But in a paragraph proof or in a two-column proof, this parallelism is obscured.

Okay, so we can use flowcharts for simple homework ‘showthat’ type problems, but what else? Of course, we can use math algorithms for repeated use, taking in various inputs and yielding various outputs, such as an algorithm to find the roots of an arbitrary quadratic in one variable. To see what people have done in algebra flowcharting, try entering the phrase ‘algebra in a flowchart’ into a search engine and look at Images.

Of the two memes, the one for flowcharting has fared a bit better in math and science, but not nearly as well as I image that it should go. Of course, flowcharts have been used for decades in math/science to represent algorithms and complex procedures. One of the earliest flowcharts I can remember, I saw in a 1970s chemistry textbook diagramming a complex analyte precipitate process. But what I find lacking is the encouragement of students to use flowcharts in their **everyday** problem solving.

Top-Down Design

Now to the other meme of Top-Down Design with Stepwise Refinements: I went over this in some detail in the last paper of this series, so I won’t belabor that here. But I do wish to repeat just enough of it to get the main idea across. Say our word problem to solve is to find the time it would take two printers, working at different rates, R_1 and R_2 , to complete a specific print job.

Scheme has taught us to search for totals and parts. Okay, there is a total of **one job** being done by two contributing printers: Printer 1 and Printer 2. We know that every total is **equal** to the sum of its parts. So, let's introduce the shorthand 'part of job done by' \rightarrow PJDB. Then our highest-level equation is

$$1 \text{ job} = (\text{PJDB Printer 1}) + (\text{PJDB Printer 2}). \quad (1)$$

Then, by stepwise refinements, this last equation can be whittled down to standard algebra in the form

$$1 = R_1 T_1 + R_2 T_2. \quad (2)$$

My personal Journey into all this

I took my first computer class at university in the early 1980s (while finishing my undergraduate degree in mathematics), in which I learned of both memes. By 1985, I was adapting flowcharting to solve high school geometry problems (for tutoring purposes). Today, this graphical technique has found very limited usage in high school mathematics. Figure E2 reveals my personal timeline of utilizing these two memes in novel ways (at least to me) in my own math/science studies.

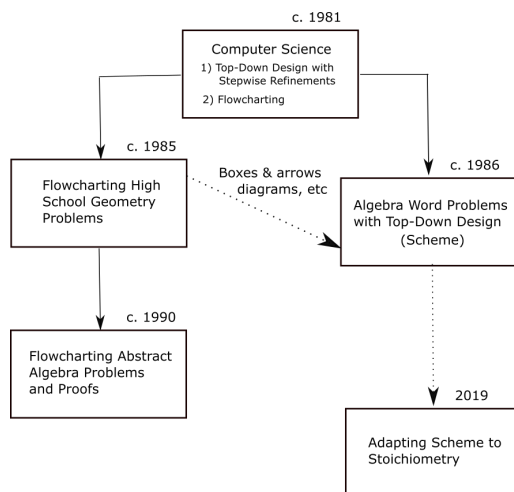


Figure E2. This graphic reveals the timeline of two memes (top-down design, flowcharting) that I carried from computer science into my math/science studies.

Yes, there's a lot of good flowcharting for high school geometry (for the few that use it); however, the use of flowcharting for secondary mathematics and advanced mathematics has scarcely moved beyond that. Just enter 'flowchart geometry' or 'math theorems in a flowchart' into a search engine to see what other people have done with it.

More Flowcharting: Real Proofs

In a theorem flowchart, one sees the typical necessity, in all but the easiest problems, to employ decision statements (nodes, usually diamond shaped) in the diagram. I refer to these nodes as ‘logic splitters’. These statements are of the tautological form $P \vee \neg P$ (read ‘ P or not P ’), where P is a logical proposition (that is, it’s either true or false). Because such a statement is always true, it can be inserted anywhere without messing up the logic, but it remains for the flowchart developer to demonstrate its usefulness in a given problem.

Placing logic splitters into a flowchart of an established algorithm where it arises naturally is a no-brainer. For an example of where a logic splitter would arise naturally in a given algorithm, consider the situation where one is determining the roots of a quadratic equation in one variable and D is the discriminant, as in Figure E3.

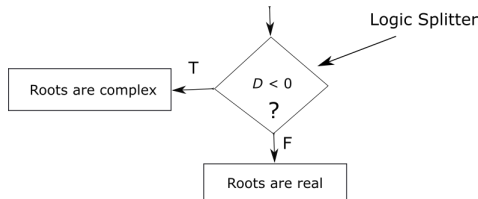


Figure E3. Obviously, either the discriminant D is negative or it’s not.

Now, for an example of a theorem in a flowchart where the logic splitter is not obvious. Let’s do a flowchart proof of the Fundamental Theorem of Arithmetic by the second form of mathematical induction (I won’t include a proof of the uniqueness part of the theorem).

This theorem states that every natural number n greater than or equal to 2 is factorable into the form

$$n = \prod_{i=1}^k p_i, \quad (3)$$

where each p_i is a prime and the factors need not be distinct. To anticipate the limiting case of the ‘factorization’ of n when n is itself a prime, we’ll ‘factor’ it as $n = p_1$. We’ll employ a proof by induction, with our inductive hypothesis that *every* natural number greater than 1 and less than n is factorable into a product of primes. The flowchart proof of the Fundamental Theorem of Arithmetic is in Figure E4. Some obvious stepwise justifications have been left to the reader to provide.

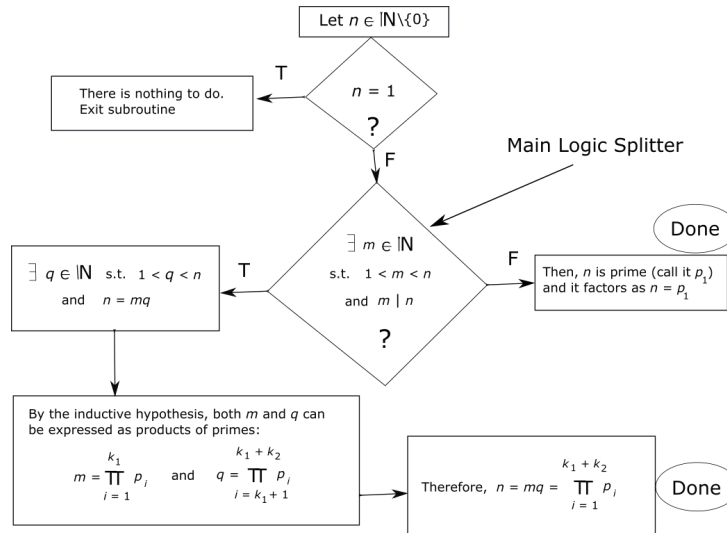


Figure E4. Flowchart for the Fundamental Theorem of Arithmetic [existence part], using the second form of mathematical induction. Note: $k = k_1 + k_2$.

By the way, I did not need to start the proof at step $n = 1$, but I included it for those who think that induction proofs require it.

Now that we have a real-life example to consider, let's make some observations on the benefits of flowcharting theorem proofs, starting with the logic splitter.³

The logic splitter in a theorem-proof flowchart plays the role in a proof that a plot device⁴ plays in a story. Both story and proof have beginnings, middles, and ends, and both often need help in the middle to push things along.

It's been said that one advantage of doing a proof by contradiction is that in that approach one adds information to the system of givens to help find a useful conclusion (in that case, a contradiction). The logic splitter is similar, but much more general, for one can add in any proposition P so long as one tests both P and $\neg P$, each going in a different direction out of the logic splitter.

For example, say you're looking to prove the existence of an element a of set S that satisfies condition Q , i.e., that $Q(a)$ is true. Then set your proposition P to ' $\forall x \in S, Q(x)$ is false'. The negation of a 'for all' is a 'there exists'. In this case, $\neg P$ implies ' $\exists a \in S$ s.t. $Q(a)$ is true'. And, voila: we have a pro forma means to shoehorn into the logic flow of a proof a test for the existence of a particular element of a set!

Beyond the psychological barriers to the potential meme of flowcharting theorems, there are also practical reasons to resist them. Even if the idea of

³I go into a lot more detail in my paper *Group Theory and the Logic of Proof*.

⁴A *plot device* in a story helps move the plot forward, toward the end of the story.

flowcharting theorem proofs were to catch on, publishers would still be reluctant to employ them liberally. First, because they take up more space than an equivalent paragraph proof (spaghetti code⁵). Second, because they are labor intensive to construct. And third, because they can be a pain to find their best placement on a printed page. But, in spite of these concerns, I think they're well worth it.

This idea that a flowchart proof takes up too much space on the page is more than just considerations of the overall length of the book or journal article it resides in. There is this irrational feeling some people get called (in the publishing world) "Fear of the white space." There is a time in a presentation when less is better because it's easier for the mind to comprehend a smaller unit of information. We should resist the urge to fill up the white space of a flowchart with maximal information. Such diagrams can be hard to understand and are often referred to as being 'too busy'.

And one more reference to computer programming: Indentation of computer code is usually promoted by computer experts because it makes the code easier to read and to understand. Yet, indentation of code can generate a lot of white space on the page.

Okay, am I suggesting, for example, that Andrew Wiles's 129-page proof of Fermat's Last Theorem should be recast into flowchart form? Maybe so – sections of it, anyway. But one thing I know: I remember seeing Wiles presenting his explanation of the long process he used to find his proof, and he did so with the help of a flowchart on a chalkboard.

Historical Perspective on Flowcharting

I know what some of you must be thinking: Flowcharting has been around for a long time, so why blame this fictitious semipermeable boundary between Computer Science and Mathematics? Well, yes, flowcharting in one form or another has been around for decades. According to the website

[https://www.studymode.com/essays/
History-Of-Flow-Chart-1846619.html](https://www.studymode.com/essays/History-Of-Flow-Chart-1846619.html)

The first structured method for documenting process flow, the "flow process chart", was introduced by Frank Gilbreth to members of the American Society of Mechanical Engineers (ASME) in 1921 in the presentation "Process Charts — First Steps in Finding the One Best Way".

In fact, I'd say that flowcharting has been around for as long as engineers have made schematics of the flow of fluids in a complicated piping system. But I am interested in generalized flowcharts, where abstract things 'flow' through

⁵*Spaghetti code* is a term used in computer science to put down code that's hard to follow by its unvirtuous lack of 'structure' to reveal its logic flow.

the diagram, such as project goals, authority structures, logic. Thus, in PERT diagrams,⁶ or work-flow diagrams, we deal with activities that have to be performed by prescribed dates, by assign persons responsible for the activities, and account for all contingencies along the way.

I can think of other kinds of flowcharts, such as those revealing ‘subordinating’ structure among members of a set or group, like ranks in the military or varieties of plants (kingdom, phylum,..., genus, specie), or authority relationships in a company. These examples seem to fall under the definition of *taxonomy*, which does sometimes use flowcharts, or something like it. For example, consider the taxonomy of $n \times n$ matrices found at

<https://networkscience.wordpress.com/2012/05/04/taxonomy-of-matrices/>

See Figure E5.

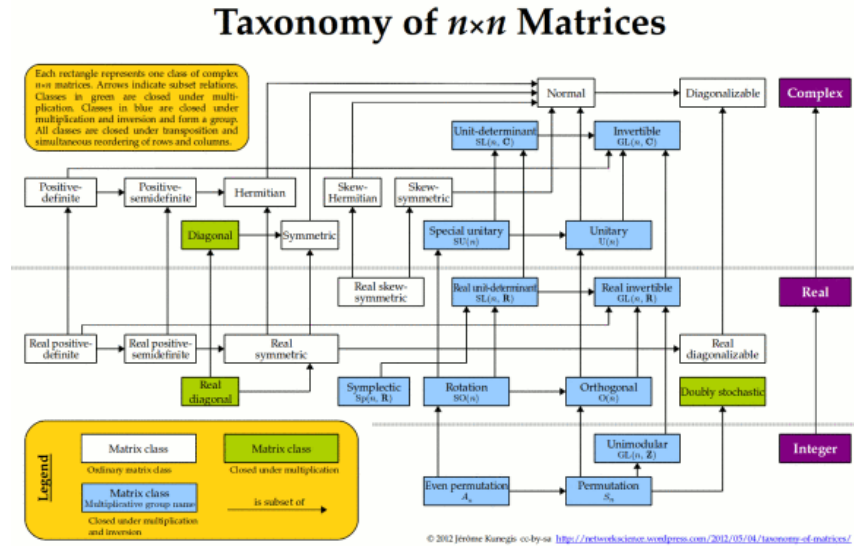


Figure E5. This graphic is used by permission under Creative Commons Licensing Agreement.

Quite recently, I saw a photograph of a 400-year-old book opened to a particular page, on which was a genealogy (in the form of ovals and line segments) that filled up one whole page.

A Flowchart from Physics

For some time now, I have had on my bedroom wall a flowchart of the proof that one can use Maxwell’s equations to prove that in a region of space free

⁶PERT stands for ‘Program Evaluation Review Technique’.

of charges or currents that the \mathbf{E} field behaves as a wave, which propagates at speed $1/\sqrt{\epsilon_0\mu_0}$. I find such a flowchart, not only compact, but also beautiful.

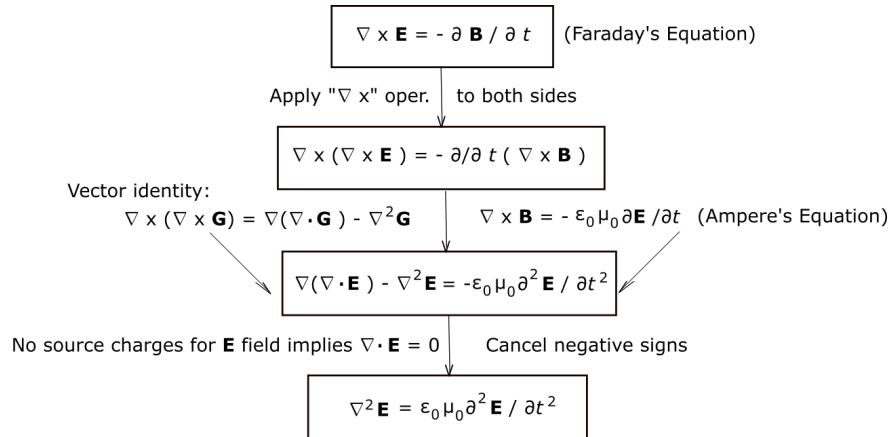


Figure E6. Here we have a flowchart proof that, in the absence of charges and currents, each component of the \mathbf{E} field satisfies a wave equation.

Although it's true that the flowchart in Figure E6 has a similarity to a two-column proof, I contend that my flowchart version is superior, mainly because the LHSs and RHSs of the equations can be both manipulated and annotated independently of each other, all without loss of clarity.

Definition of a Flowchart

So, what do we mean by a 'flowchart'? According to the website

<https://er.yuvayana.org/flowchart-history-definition-benefits-limitation/>

Flow charts are easy-to-understand diagrammatic representation to showing how steps in a process fit together.

According to the Wikipedia

A flowchart is a type of diagram that represents an algorithm, workflow or process. Flowchart can also be defined as a diagrammatic representation of an algorithm (step by step approach to solve a task).

Now, is a graphic a flowchart if it's not labeled as such? Well, yes, it can be. In fact, flowcharts are quite ubiquitous, hidden in plain sight. For example, a Feynman diagram is a flowchart. The 'arrow chasing' diagrams of category theory are flowcharts. And so on. Perhaps it all depends on how one defines a flowchart. Now it's my turn.

I define a *flowchart* as a nonempty collection of a finite number of nodes connected by line segments (edges), where a nonempty subset of these nodes are *starting nodes* and another nonempty distinct subset of the nodes are *finishing nodes*. Using the language of graph theory, such a structure is called a (generalized) *graph*. Every starting node must have at least one pathway to at least one finishing node. And every finishing node must have at least one inverse path to a starting node. The number of times a given node is traversed must be finite. These generalized graphs may have loops whereby nodes may be traversed more than once (e.g., loops in algorithms). They may also have loops from a given node to itself (e.g., identity morphisms on objects in category theory).

The line segments represent relationships between the pairs of nodes they connect. Typically, there is only one relationship per graph, but more relationships are allowed so long as the various relationships are visually distinguishable and consistently applied. Case in point: A typical genealogy flows from top to bottom. Following the arrows between two vertical nodes means ‘whose offspring is’, whereas, the horizontal relationships mean ‘is the spouse of’. Some genealogies use a neutral horizontal bar as a logic collector, which serves to collect all offspring from a pair of spouses (i.e., their parents), from which line segments run down to the offspring.

Whether the line segments have arrows or not, there is a flow from the start nodes to the finish nodes. Therefore, these relationships are not all symmetric. Usually, all these relationships are of one type: ‘the next step to be performed is’, ‘successor is’, ‘is followed by’, ‘logically implies’, ‘whose offspring is’, ‘to be accomplished by such-and-such a time’, etc. Referring again to a top-down genealogy, vertical relationships are unsymmetric, whereas horizontal relationships are (or at least can be written as) symmetric.

Is the Notion of a Flowchart Old Fashioned or Outdated?

Well, some say that it is. After all, these days we have such ‘modern’ things as data-flow diagrams, action diagrams, sequence diagrams, etc. However, my short answer is, No, of course not. I have defined the flowchart so generally that all these are subsumed by the notion of flowchart. One definition to rule them all, you might say.

On the other hand, there are graphical representations of things and/or relationships that do *not* require start nodes or finish nodes, and they would not be flowcharts, but they’re still very useful.

For example, what immediately comes to my mind is the logical equivalence of the Axiom of Choice, the Well-Ordering Principle, and Zorn’s Lemma (Figure E7).

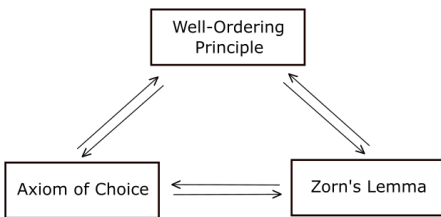


Figure E7. No starting or finishing nodes are needed in this diagram. The relationship assigned to each arrow is 'logically implies'. The relationship assigned to the double arrow is 'logically equivalent to'.

By the way, as a consequence of the logical equivalence of the three theorems, it follows that if one of them is true/false, then they're all true/false. But our human intuition probably fails us here. As the Wikipedia article on Zorn's Lemma puts it:

"The Axiom of Choice is obviously true, the well-ordering principle obviously false, and who can tell about Zorn's lemma?"

-- Jerry L. Bona

Greater Horizons

I trust that I have made my case that I'm not promoting flowcharting to the exclusion of all other forms of data visualization. Indeed, the subject of data visualization has become a vast discipline on its own, called *Infographics*, which Wikipedia defines as

Infographics (a clipped compound of "information" and "graphics") are graphic visual representations of information, data or knowledge intended to present information quickly and clearly.

Of recent importance in education is the *concept map*, which Wikipedia defines as

A concept map or conceptual diagram is a diagram that depicts suggested relationships between concepts. It is a graphical tool that instructional designers, engineers, technical writers, and others use to organize and structure knowledge.

Most of the example images of concept maps I saw on the Web easily fit my definition of a flowchart. But I can think of other infographics that are not flowcharts, such as a graphic demonstrating the hydrologic cycle, which has no starting or finishing points.

So, I'm really very much in favor of expanding our horizons in the use of graphical aids, not arbitrarily restricting them. To that point, I'll make a final plea for people to expand their use of flowcharts, especially in problem solving.

Consider three problems. First, from computer science: Write a program to test the primality of a given integer. Second, from plumbing: Write a procedure to fix a leaking pipe under the sink. And third, prove Cauchy's theorem in group theory, that if p , a prime, divides the order of a finite group G , then G has an element of order p . What are the commonalities here?

So, what do we mean by "problem solving" in the first place? *Problem solving* is a rational process of going step-wise from a well-defined starting state to a well-defined finishing state. And that's exactly what flowcharts can easily model.

Conclusion

Although flowcharts have academic uses well beyond problem solving, my emphasis in this appendix has been to argue for a much more robust application of flowcharts to everyday problem solving in science and mathematics, for professionals and for students. Visualize and conquer.