STEM In the AI Age.

For Freshers in any STEM subject

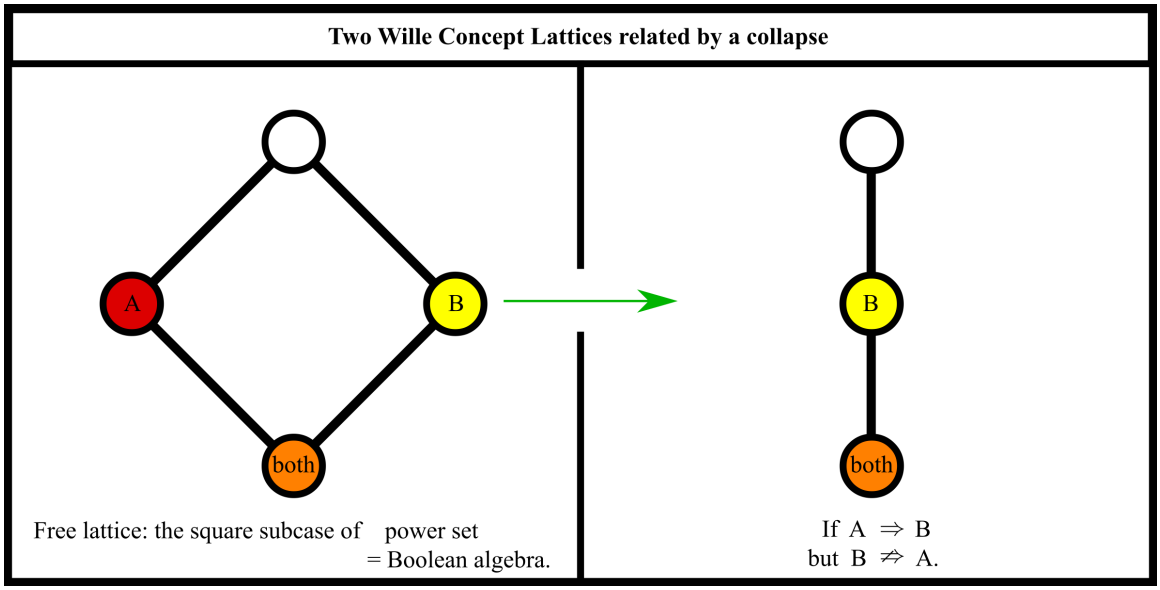
Edward Anderson[1]

[6] Socratic Thinking.

[6] Graphs and Orders.

[8] Formal Languages, with emphasis on Syntax.

[4] The Wille Concept Lattice.



Cite as E. Anderson, “STEM In the AI Age. For Freshers in any STEM subject", Online Encyclopaedia of Applied Graph and Order Theory, institute-theory-stem.org/oeagot-ai-stem-0/ (2026).

[1]Dr.E.Anderson.Maths.Physics *at* protonmail.com Institute for the Theory of STEM.

Date stamp: v1: 09 Nov 2026. Copyright of Dr E. Anderson.

1 Format and Motivation

Intended Audience This course is for Freshers in *any* STEM subject. Or for self-study by whoever has attended school up to the age of 18 and is interested in some part of STEM.

Duration The above numbers are not references. They are, rather, "hours of instruction". So it takes 24 hours to see this material. It will typically take longer for you to understand it and become fluent with it. At a university, this might have been 1/4 of your work-load for a term.

The Socratic Thinking is to be done in an active practical manner. With two of its units spread out along the tail of the course in support of the last two topics. Everything else has a seminars and exercises format.

Motivation AI has totally changed the nature of Mathematics and Computer Science. And will very likely shortly be changing the nature of every other STEM subject. This course represents substantial advances in navigating AI. For one's transition from school life to university life. A first tech job. Or a season of self-study.

2 Course outline

2.1 Socratic Thinking

1) Asking questions is valuable in an intellectual space. We foster confidence in those who do not immediately ask questions. Questions in parallel versus questions in series, in relation to bredth versus depth of enquiry. One's first pertinent question, and one's first pertinent follow-up question. Aiming to increase one's proportion of pertinent questions.

Prompting an LLM. Filtering its Suggested Sequels.

2) The traditional technique of Socratic Thinking. Socratic questions, Socratic probing and Socratic questioning of questions.

3) Socratic probing, and Socratic questioning of questions, as methods of prompting LLMs. And as a method of checking for the plausibility of answers that LLMs generate. Other types of check on this.

4) Advantages of Lateral Thinking in performing a Google search. LLM counterparts. Conditioning prompts to an LLM so as to block out other interpretations of your prompt.

5) Conditioning prompts to an LLM so as to generate replies that are easily parsable by Humans. See Sec 2.3 for details.

6) Detecting conceptual incompleteness in an LLM-driven line of enquiry using Wille's Concept Lattice (Sec 2.4) How this supercedes simpler methods of conceptual analysis.

Comment A further idea here is to Socratically question items elsewhere in this Course Outline. These provide some interesting default topics. So that everybody has at least some topics to Socratically question, whether with each other, or between a Student and an LLM. The objective of this Socratic Thinking section is for everybody involved to emerge with at least 12 hours of practical experience in Socratic Thinking. This confers robustness to whatever else you do. Be it the rest of this course, or your other intellectual pursuits, or your own interactions with LLMs or further types of AI.

2.2 Graphs and Orders

1) Simple graphs [2], with smallest looped and multi-edge counterexamples. The simple graphs on up to 4 vertices. Degrees. Euler's Handshake Lemma. Examples of graphs throughout STEM.

2) Interactive exercises on finding the simple graphs on 5 vertices. On the minimum number of vertex removals it takes to split a connected graph into 2 pieces. And its edge removal version counterpart. And on the trees on up to 8 vertices. Rooted and directed trees. Small data-structure trees, and sort and search operations on these [3].

3) Total orders, alias chains. Versus partial orders [4]. Posets are partially-ordered sets. Hasse diagrams. Posets of combinations, divisors and subsets of a finite set. The last of which includes the power sets, which coincide with the Boolean algebras in this finite setting.

4) Interactive exercise on finding all the posets on up to 4 vertices. A few of the simplest kinds of map on graphs and then on posets.

5) Lattices in the sense of Order Theory. Which posets on up to 4 vertices are furthermore lattices. A small selection of adjectives about lattices, concentrating on attributes of Boolean algebras.

6) Interactive exercise on finding all the lattices on 5 vertices. Integer partitions. Young diagram notation. The Young lattice of partitions. Going far enough for this to be irregular. In contrast with the arenas of combinations, subsets, and divisors lattices' regularities.

2.3 Formal Languages and Syntax

1) Objects and maps versus objects and arenas. And then maps and mapping spaces. Given some kind of object, the arena is the Mathematical space formed by the totality of those objects. Arenas moreover do not usually belong to the same traditional branch of STEM theory as the underlying objects. With Combinatorial examples and exceptions. That this is a type of Modern Applied Topology, whose first two large children are Big Data and LLMs.

2) Alphabets as arenas of letters. Strings as arenas of combinations of letters. Words as a vocabulary's restriction of strings. 0 or 1 , or No and Yes, or False and True as one of the smallest alphabets. Adding the blank to an alphabet. Collections of strings versus their restriction to phrases.

3) Syntax grammars as a means of performing this last restriction. Semantics as a map from words to a reality or an abstraction. Semantic ambiguity. Human Languages versus Computer Languages. Relative significance of Syntax and Semantics in these, in Scientific communications and elsewhere.

4) Chomsky's mathematization of grammar, with ensuing classification. Outline of Chomsky's philosophy of language.

5) Parsing. Cognitive overload. Sentences containing no more than 20 words. Sentences containing no more than 5 technical concepts. Limiting sentences' concept graphs' edge density. Collapse of Chomsky's grammar types under such combined parsing rules. Scale-invariant structure of a Humanly parsable Scientific communication.

6) Example of a parsable grammar protocol. Its corresponding list of allowed concept sentence graphs. Cutting a sentence up into equivalent shorter sentences. With an example given and several examples set. Using a parsable grammar protocol to condition a thread of LLM prompts.

* * *

7) Exercise testing some free LLMs' adherence to parsable grammar protocol. Using a parsable grammar protocol to manually redraft a text.

Comment This unit amounts teaching yourself to write with some of the clarity benefits of Dirac (Physics). Of Jean-Pierre Serre (Mathematics) Of Leslie Lamport (Computer Science). Or of Claude Shannon (Engineering and Information Theory). And to obtain answers from your LLM in a similar style: readily comprehensible to you, with scaling to whoever you share it with.

8) Ways in which LLMs are each of and Chomskyan anti-Chomskyan. That a priori LLMs are black boxes. Some results and hypotheses about how LLMs work. Pointers to qualitative discussion of persistence and cellular sheaves as further types of Modern Applied Topology.

Comment These pointers provide a more complete view of what Modern Applied Topology consists of [5]. For all that the current course is Pre-Topological. Compare with Pre-Calculus versus Calculus. With Topology two steps ahead of Calculus, via rigorous Analysis and then abstraction of this to Topology. Pre-Topology allows for simple Topological notions: connectedness and separation in the current course. But not for theorems using Topological properties or Topological machinery.

2.4 Wille's Concept Lattice

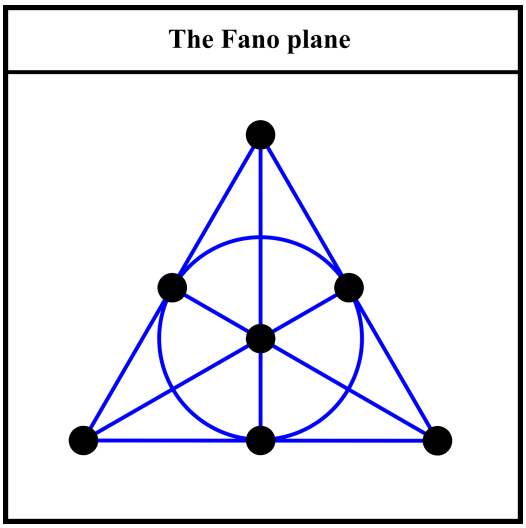
1) Lists of types and attributes. Table formulation. Wille concept lattice formulation [4]. Examples.

2) Free concepts, returning power set lattices. How mangled and incomplete conceptualizations can be spotted by their failing to return a lattice. Nuts and bolts of working out a small Wille concept lattice.

3) Using Wille concept lattices to properly scout out conceptual diversity in prompting an LLM. Getting freely-available Gemini [7] to plot Wille concept lattices for the notions discussed in a prompt thread. Including obtaining Hasse diagrams of Wille concept lattices as output from Gemini. Conditioning prompts to get Gemini to draw better graph and poset pictures, including in particular styles that are visually useful to Humans.

4) Dependencies causing collapse away from power set lattices. Dependencies causing irregularity. With examples of collapse and irregularity. With an example to work out by hand, and more examples for students to tease apart with freely available Gemini. Objects that dominate a Wille concept lattice, with automated LLM detection of dominant objects.

Pointer to how dominant objects arise along multiple conceptually distinct routes of thought. And how some of the dominant objects are additionally beautiful, as exemplified by the Fano plane [6].



References

[1] N. Chomsky, *Syntactic Structures* (de Gruyter 2002; the original dates back to 1957).

[2] R.J. Wilson, *Introduction to Graph Theory* (Longman, Edinburgh 2010; the original dates to 1972).

[3] D.E. Knuth, *The Art of Computer Programming* **1** *Fundamental Algorithms*; **3** *Sorting and Searching*. (Addison-Wesley, Reading MA. 1998).

[4] Chapters 1 to 3 of B.A. Davey and Hilary A. Priestley, *Introduction to Lattices and Order* (C.U.P, Cambridge 2002).

[5] R. Ghrist, *Elementary Applied Topology* (2014).

[6] E. Anderson, "The Fano Configuration, Plane and Graph", Online Encyclopaedia of Applied Graph and Order Theory, institute-theory-stem.org/fano/ (2026).

[7] Gemini by Alphabet as found on the Google search page. At the point of writing, this often means Gemini 1.5 Pro or Gemini 3.5 Flash.