Machine Learning in Symbolic Computation - A Skeptical Perspective

Michael Kohlhase
Computer Science
FAU Erlangen-Nürnberg
https://kwarc.info/kohlhase

Symbolic Computation and Machine Learning, 6. July 2026

1 Introduction

Warning: I will not . . .

 I will not tell you how to do SC better with ML! 

Warning: I will not . . .

 I will not tell you how to do SC better with ML! 

instead

I will try to help you think about what is happening with the AI/LLM hype

Warning: I will not . . .

 I will not tell you how to do SC better with ML! 

instead

I will try to help you think about what is happening with the AI/LLM hype

after all,

I promised you a skepticist view

Warning: I will not . . .

 I will not tell you how to do SC better with ML! 

instead

I will try to help you think about what is happening with the AI/LLM hype

after all,

I promised you a skepticist view
even though that becomes harder and harder to maintain!

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)
- I abandoned "real math" after my Master's Degree $\leadsto$ meta-mathematics $\widehat{=}$ ATP

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)
- I abandoned "real math" after my Master's Degree $\leadsto$ meta-mathematics $\widehat{=}$ ATP
- I abandoned ATP after my Dissertation $\leadsto$ NL Semantics & Math Knowledge Representation

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)
- I abandoned "real math" after my Master's Degree $\rightsquigarrow$ meta-mathematics $\widehat{=}$ ATP
- I abandoned ATP after my Dissertation $\rightsquigarrow$ NL Semantics & Math Knowledge Representation
- Current research Interests
  - Meta-Meta Mathematics $\widehat{=}$ Meta-Logical Frameworks $\widehat{=}$ Logic Engineering.
  - Flexiformalization $\widehat{=}$ Formalization (of Math) to flexible levels of formality.
  - Mathematical Knowledge Management (MKM) $\widehat{=}$ Corpus Meta-Mathematics
  - Applications of flexiformal MKM outside Math
    - Education $\rightsquigarrow$ formalizing Domains, Competencies and Didactics (and automating them)
    - (the monotonic part of) Legal Reasoning (partially automating register court decisions)
    - Mathematical/technical language $\rightsquigarrow$ machine-supported flexiformalization.

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)
- I abandoned "real math" after my Master's Degree $\rightsquigarrow$ meta-mathematics $\widehat{=}$ ATP
- I abandoned ATP after my Dissertation $\rightsquigarrow$ NL Semantics & Math Knowledge Representation
- Current research Interests
  - Meta-Meta Mathematics $\widehat{=}$ Meta-Logical Frameworks $\widehat{=}$ Logic Engineering.
  - Flexiformalization $\widehat{=}$ Formalization (of Math) to flexible levels of formality.
  - Mathematical Knowledge Management (MKM) $\widehat{=}$ Corpus Meta-Mathematics
  - Applications of flexiformal MKM outside Math
    - Education $\rightsquigarrow$ formalizing Domains, Competencies and Didactics (and automating them)
    - (the monotonic part of) Legal Reasoning (partially automating register court decisions)
    - Mathematical/technical language $\rightsquigarrow$ machine-supported flexiformalization.
- Extenuating circumstances that shape my perspective
  - I organize the AI Master Program at FAU (600 students, 10.000 Applications/year)
  - I am head of the FAU Competence Center for Research Data and Information.
  - As a member of the AI Council at FAU, we design a course Framework for an AI intro of all 40.000 students.

Full Disclosure on my Position/Perspective

- I am a Mathematician by genetics and training. (i.e. where my heart is and my intuitions are from)
- I abandoned "real math" after my Master's Degree $\rightsquigarrow$ meta-mathematics $\widehat{=}$ ATP
- I abandoned ATP after my Dissertation $\rightsquigarrow$ NL Semantics & Math Knowledge Representation
- Current research Interests
  - Meta-Meta Mathematics $\widehat{=}$ Meta-Logical Frameworks $\widehat{=}$ Logic Engineering.
  - Flexiformalization $\widehat{=}$ Formalization (of Math) to flexible levels of formality.
  - Mathematical Knowledge Management (MKM) $\widehat{=}$ Corpus Meta-Mathematics
  - Applications of flexiformal MKM outside Math
    - Education $\rightsquigarrow$ formalizing Domains, Competencies and Didactics (and automating them)
    - (the monotonic part of) Legal Reasoning (partially automating register court decisions)
    - Mathematical/technical language $\rightsquigarrow$ machine-supported flexiformalization.
- Extenuating circumstances that shape my perspective
  - I organize the AI Master Program at FAU (600 students, 10.000 Applications/year)
  - I am head of the FAU Competence Center for Research Data and Information.
  - As a member of the AI Council at FAU, we design a course Framework for an AI intro of all 40.000 students.
- I assume that SC is an euphemism for Math Computation/Software, and take that as an excuse to talk about what I like most: Math software ... and ML/LLM influence on that.

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"
  - Abhishek Chugh: "*I take one day of ML/LLM detox a week, so that I do not become obsolete as a software engineer.*"

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"
  - Abhishek Chugh: "*I take one day of ML/LLM detox a week, so that I do not become obsolete as a software engineer.*"
  - Dennis Müller: "*Whenever I ask Claude Code, the answer convinces and even amazes me at first, but when I look deeper, each and every one contains at least deep, non-obvious, and ususally crippling flaw*".

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"
  - Abhishek Chugh: "*I take one day of ML/LLM detox a week, so that I do not become obsolete as a software engineer.*"
  - Dennis Müller: "*Whenever I ask Claude Code, the answer convinces and even amazes me at first, but when I look deeper, each and every one contains at least deep, non-obvious, and ususally crippling flaw*".
  - Jana Kohlhase: "*I asked Claude Sonnet to fix some red tests, and it did; and then I saw that it had rewritten the test harness to ignore failing tests. When I forbade that, it just extended all tests with `return true`.*"
  - My AI students (on average): "*Me, I am modern, I use ChatGPT for my homework submissions*".

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"
  - Abhishek Chugh: "*I take one day of ML/LLM detox a week, so that I do not become obsolete as a software engineer.*"
  - Dennis Müller: "*Whenever I ask Claude Code, the answer convinces and even amazes me at first, but when I look deeper, each and every one contains at least deep, non-obvious, and ususally crippling flaw*".
  - Jana Kohlhase: "*I asked Claude Sonnet to fix some red tests, and it did; and then I saw that it had rewritten the test harness to ignore failing tests. When I forbade that, it just extended all tests with* `return true`."
  - My AI students (on average): "*Me, I am modern, I use ChatGPT for my homework submissions*".
  - My Humanities Colleagues: "*I simply ask ChatGPT to find out whether an essay was prompted*".

Influences/Judgments on ML/LLM from Colleagues

- I still have to make my mind about ML/LLM in Symbolic Computing and Computational Logic
- Asking others does not really help:
  - Andreas Maier: "*Just give me Compute, the rest my LLM-based agents can do!*".
  - Florian Rabe: "*For programming, I see Claude Code as equivalent to a good Master's student, where you build up a very good specification in the prompt (supervision)*".
  - John Harrison: "*All but the hardest HOL Light proofs can nowadays be generated by LLMs*". (3^{rd} hand)
  - Deyan Ginev: "*After working to port LaTeXML from Perl to Rust for 10 years, I bought a Claude Code subscription and prompted the rest in 90 days*"
  - Abhishek Chugh: "*Front-end programming is dead (or will be in 6 months): all programming will move to dark code factories.*"
  - Abhishek Chugh: "*I take one day of ML/LLM detox a week, so that I do not become obsolete as a software engineer.*"
  - Dennis Müller: "*Whenever I ask Claude Code, the answer convinces and even amazes me at first, but when I look deeper, each and every one contains at least deep, non-obvious, and ususally crippling flaw*".
  - Jana Kohlhase: "*I asked Claude Sonnet to fix some red tests, and it did; and then I saw that it had rewritten the test harness to ignore failing tests. When I forbade that, it just extended all tests with* `return true`*.*"
  - My AI students (on average): "*Me, I am modern, I use ChatGPT for my homework submissions*".
  - My Humanities Colleagues: "*I simply ask ChatGPT to find out whether an essay was prompted*".
- I remain skeptical about what part of my research/teaching I even want to delegate to LLMs.
- **Example 1.1.** Writing papers/reports myself helps me better understand the ideas.

2 I have not timed my presentation yet . . . ,

$\rightsquigarrow$ starting with the conclusion . . . ,

$\rightsquigarrow$ I will able to conclude on time!

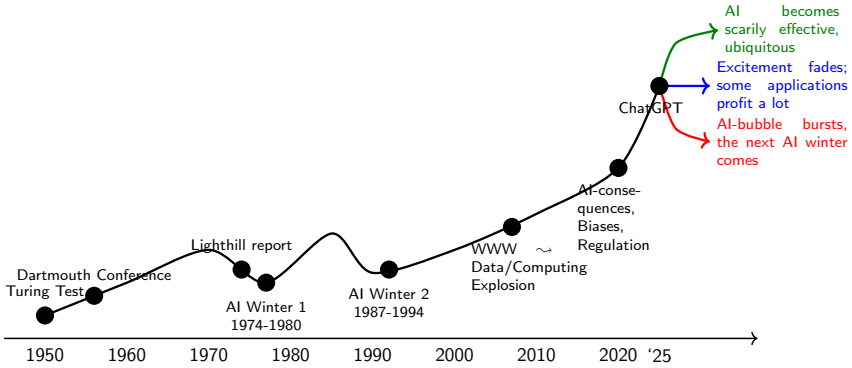
$\rightsquigarrow$ Please feel free to ask questions (we are already done)

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)



Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).
- **My Conclusion:** Instead of just playing along with the ML/LLM hype (gadget lust), we should think about consequences of whole-sale adoption of ML/LLM
  - what are long-term consequences of cognitive laziness (addiction, dependency, loss of competencies)
  - how do we have to change education given that assessment often fails. (gradable competency proxies easily solved by LLMs)
  - can we save the scientific publication system? (rescind "publish or perish"?)

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).
- **My Conclusion:** Instead of just playing along with the ML/LLM hype (gadget lust), we should think about consequences of whole-sale adoption of ML/LLM
  - what are long-term consequences of cognitive laziness (addiction, dependency, loss of competencies)
  - how do we have to change education given that assessment often fails. (gradable competency proxies easily solved by LLMs)
  - can we save the scientific publication system? (rescind "publish or perish"?)
- **To end on a positive note:** In Math and Symbolic Computation we are still relatively safe, as we have relatively robust quality criteria. (invest into formalization!)

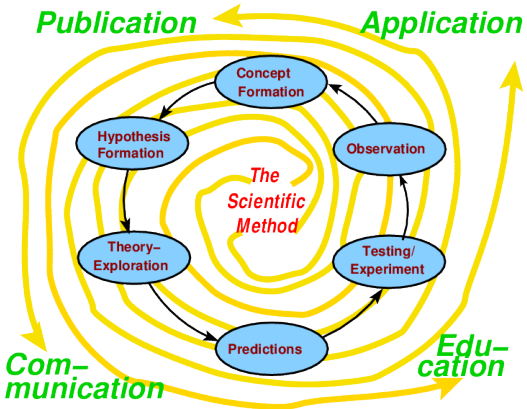
3 A Holistic Model for "Doing Math"; see [Car+21; KR26]

The "Scientific Method" interpreted as MKM (Buchberger's Spiral View)

The Scientific Method in the Creativity Spiral

after [Buchberger 95]

- **Goal:** Supported every step by software systems!
  (Towards an infrastructure for E-Science!)
- **Idea:** Concentrate on communication of the objects and knowledge involved.
  (leave the applications to specialists)
- **Observation:** Communication/system integration needs a uniform meaning space.

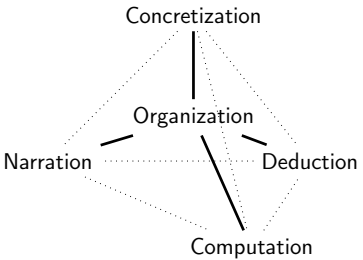


Knowledge Representation is only Part of "Doing Math"

- **Definition 3.1.** One of the key insights is that the mathematics ecosystem involves a body of knowledge externalized in an ontology that provides organization and combines the following four aspects:
  - Deduction: exploring theories, formulating conjectures, and constructing proofs
  - Computation: simplifying mathematical objects, re contextualizing conjectures...
  - Concretization: collecting concrete examples/models, applying mathematical knowledge to real-world problems and situations.
  - Narration: devising both informal and formal languages for expressing mathematical ideas, visualizing mathematical data, presenting mathematical developments, organizing and interconnecting mathematical knowledge

"Doing Math": as a Tetrapod

- We call the endeavour of creating a computer-supported mathematical ecosystem "Project tetrapod" as it needs to stand on four legs.



- **Collaborators:** KWARC@FAU, McMaster, Ljubljana

Tetrapod Example: The Fibonacci Numbers

- We present the five aspects of math using the Fibonacci numbers

Tetrapod Example: The Fibonacci Numbers

- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)

```
theory natarith =
  nat : type
  plus : nat −> nat −> nat
  ...
```

```
theory fibonacci =
  include ?natarith
  fib : nat −> nat | # F (1)
  fib_base = ⊢ fib 0 = 1 ∧ fib 1 = 1
  fib_step = ⊢ fib (n+2) = fib (n+1) + fib n
```

Tetrapod Example: The Fibonacci Numbers

- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)
- **Concretization:** We enumerate the values in a concrete data type (a list of integers)

$$fib = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, \ldots]$$

Tetrapod Example: The Fibonacci Numbers

- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)
- **Concretization:** We enumerate the values in a concrete data type (a list of integers)
- **Narration:** We talk/write about the Fibonacci numbers, e.g. in English. (addressed to humans)

  Fibonacci numbers are named after Italian mathematician Leonardo of Pisa, later known as Fibonacci. In his 1202 book Liber Abaci, Fibonacci introduced the sequence to Western European mathematics, although the sequence had been described earlier in Indian mathematics, as early as 200 BC in work by Pingala on enumerating possible patterns of Sanskrit poetry formed from syllables of two lengths. [Wikipedia]

Tetrapod Example: The Fibonacci Numbers

- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)
- **Concretization:** We enumerate the values in a concrete data type (a list of integers)
- **Narration:** We talk/write about the Fibonacci numbers, e.g. in English. (addressed to humans)
- **Computation:** E.g. a python implementation

```
def fib():
    """␣Generates␣the␣Fibonacci␣numbers,␣starting␣with␣0␣"""
    x, y = 0, 1
    while 1:
        yield x
        x, y = y, x+y
```

Tetrapod Example: The Fibonacci Numbers

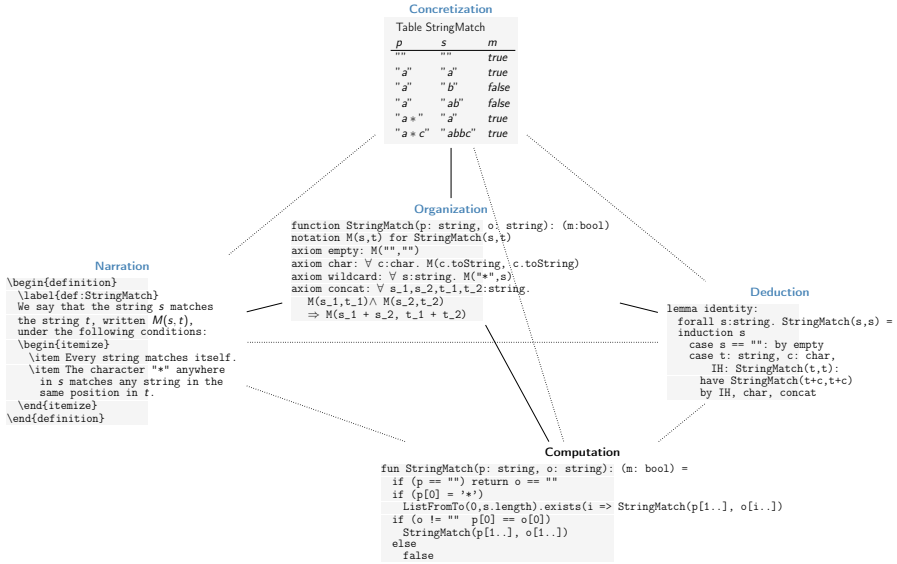
- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)
- **Concretization:** We enumerate the values in a concrete data type (a list of integers)
- **Narration:** We talk/write about the Fibonacci numbers, e.g. in English. (addressed to humans)
- **Computation:** E.g. a python implementation
- **Deduction:** We give theorems and proofs about the Fibonacci numbers, e.g.

$$\forall n \in \mathbb{N}. F_n = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-k-1}{k}.$$

Tetrapod Example: The Fibonacci Numbers

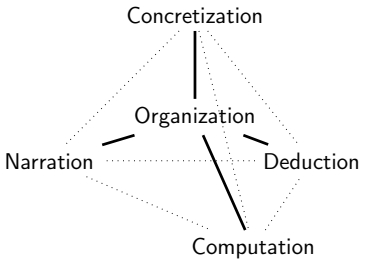
- We present the five aspects of math using the Fibonacci numbers
- **Organization:** We give the syntax/notation, possibly types, and possibly a definition/specification (so we remember)
- **Concretization:** We enumerate the values in a concrete data type (a list of integers)
- **Narration:** We talk/write about the Fibonacci numbers, e.g. in English. (addressed to humans)
- **Computation:** E.g. a python implementation
- **Deduction:** We give theorems and proofs about the Fibonacci numbers, e.g.
- **Observation 3.2.**
  - *In traditional Maths, we effortlessly combine all of these.*
  - *In computational Maths, we use different formats and workflows.*
- **But:** There is great potential for the aspect-specific tools to interact synergistically! (Challenge for CICM/AITP/SCML/...)

A Tetrapod in CS? E.g. a String Matching Function



The Meta-Effect in the Tetrapod

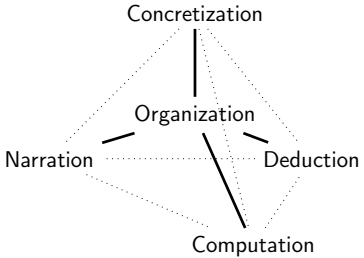
- **Possible/Frequent Objection:** But if I have computation, I can do e.g. deduction as well. Thus I cannot see any difference between the corners, theoretically.
- **Indeed:** All four corner aspects are inter-simulatable.
  (We call this the "meta-effect")
- **But:** Simulation costs naturality and efficiency!
- **So:** We only consider systems/languages in their "natural habitat" for the aspects.



- The tetrapod tries to be a conceptual model that helps you think (only)!
  (e.g. locate your system in the tetrahedron)

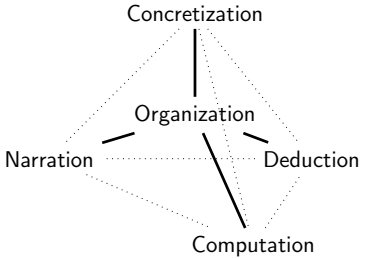
The Tetrapod and ML/LLMs

- **Observation:** We have machine-support for all aspects but narration!
  For processing narration we need human brains (so far).
- **Now:** ML/LLMs are starting to change that. (NL generation is possible)
- **Together with the Meta-Effect:** We can generate the native formal languages for the various corner aspects.
  $\rightsquigarrow$ Agentic loops can use the native corner systems to verify!



The Tetrapod and ML/LLMs

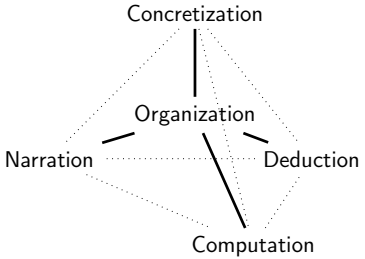
- **Observation:** We have machine-support for all aspects but narration!
  For processing narration we need human brains (so far).
- **Now:** ML/LLMs are starting to change that. (NL generation is possible)
- **Together with the Meta-Effect:** We can generate the native formal languages for the various corner aspects.
  $\rightsquigarrow$ Agentic loops can use the native corner systems to verify!
- **ML/LLM$\leftrightarrow$Meta Applications:**
  - via concretization: ML/LLM-based generation for unit test in programming
  - via computation: ML/LLM-based SC programs (this conference)
  - via deduction: ML/LLM-based formal proofs/verification.



I believe much of what SCML and AITP are about can be understood like this.

The Tetrapod and ML/LLMs

- **Observation:** We have machine-support for all aspects but narration!
  For processing narration we need human brains (so far).
- **Now:** ML/LLMs are starting to change that. (NL generation is possible)
- **Together with the Meta-Effect:** We can generate the native formal languages for the various corner aspects.
  $\rightsquigarrow$ Agentic loops can use the native corner systems to verify!



- **ML/LLM↔Meta Applications:**
  - via concretization: ML/LLM-based generation for unit test in programming
  - via computation: ML/LLM-based SC programs (this conference)
  - via deduction: ML/LLM-based formal proofs/verification.

  I believe much of what SCML and AITP are about can be understood like this.
- I remain skeptical that SC specifications can be prompted (The ideas have to come from somewhere)
- **Reinvesting the AI Acceleration into Formality:**
  Let's invest in theory, practice, scaling, and education of symbolic specification.

4 Background: Types of AI

Two ways of reaching Artificial Intelligence?

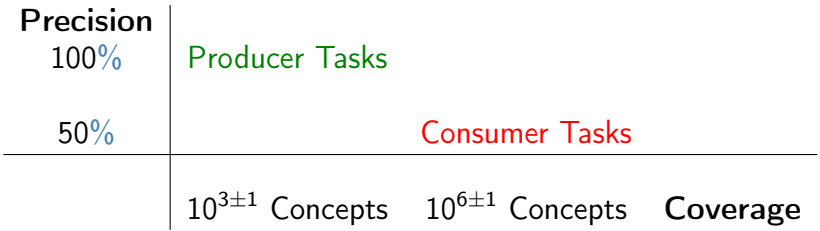
- We can classify the AI approaches by their coverage and the analysis depth.(they are complementary)

Analysis ↑ vs. Coverage →	Narrow	Wide
Deep	symbolic AI-1	not there yet cooperation?
Shallow	no-one wants this	statistical/sub symbolic AI-2

- **This semester** we will cover foundational aspects of symbolic AI. (deep/narrow processing)
- **next semester** concentrate on statistical/subsymbolic AI. (shallow/wide-coverage)

Environmental Niches for both Approaches to AI

- **Observation:** There are two kinds of applications/tasks in AI
  - Consumer tasks: consumer grade applications have tasks that must be fully generic and wide coverage. (e.g. machine translation like Google Translate)
  - Producer tasks: producer grade applications must be high-precision, but can be domain-specific (e.g. multilingual documentation, machinery-control, program verification, medical technology)



after Aarne Ranta [Ran17].

- **General Rule:** Subsymbolic AI is well suited for consumer tasks, while symbolic AI is better suited for producer tasks.
- A domain of producer tasks I am interested in: mathematical/technical documents.

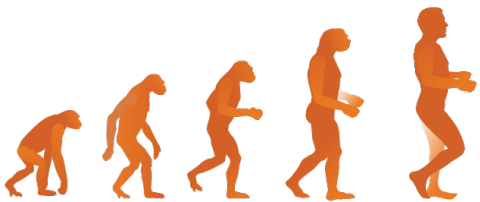
5 Evolution of Programming by Agentic LLMs

Evolution of Programming?

- **Evolution of Programming:** At least that is what some colleagues/companies are advocating!
  1. Opcodes $\rightsquigarrow$ bit sequences that are directly given to the CPU
  2. Assembler languages $\rightsquigarrow$ simple instructions directly mappable to opcodes
  3. Primitive programming languages $\rightsquigarrow$ `if/then/else`, `goto`,...
  4. Structured languages $\rightsquigarrow$ `while/do`, `repeat until` ...
  5. High-level Languages $\rightsquigarrow$ functional/declarative constructs, ...

Evolution of Programming?

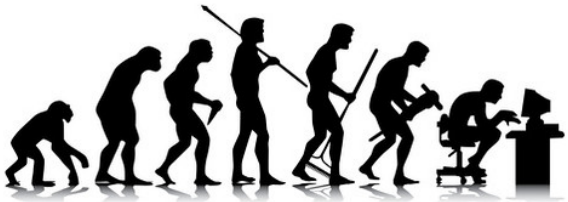
- **Evolution of Programming:** At least that is what some colleagues/companies are advocating!
  1. Opcodes $\rightsquigarrow$ bit sequences that are directly given to the CPU
  2. Assembler languages $\rightsquigarrow$ simple instructions directly mappable to opcodes
  3. Primitive programming languages $\rightsquigarrow$ `if/then/else`, `goto`,...
  4. Structured languages $\rightsquigarrow$ `while/do`, `repeat until` ...
  5. High-level Languages $\rightsquigarrow$ functional/declarative constructs, ...
  6. Natural language specifications $\rightsquigarrow$ prompts?
- Really? (actually this picture may be just as flawed as the ones below.)
- **Example 5.1 (Human Evolution).**



https://www.flickr.com/photos/55524309@N05/5377715421.png

Evolution of Programming?

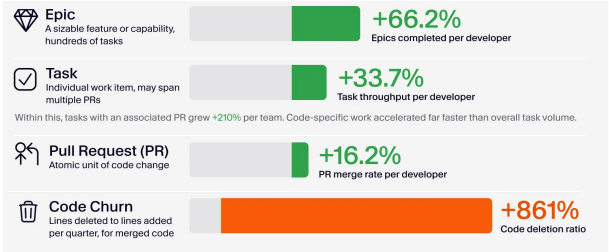
- **Evolution of Programming:** At least that is what some colleagues/companies are advocating!
  1. Opcodes $\rightsquigarrow$ bit sequences that are directly given to the CPU
  2. Assembler languages $\rightsquigarrow$ simple instructions directly mappable to opcodes
  3. Primitive programming languages $\rightsquigarrow$ `if/then/else`, `goto`,...
  4. Structured languages $\rightsquigarrow$ `while/do`, `repeat until` ...
  5. High-level Languages $\rightsquigarrow$ functional/declarative constructs, ...
  6. Natural language specifications $\rightsquigarrow$ prompts?
- Really? (actually this picture may be just as flawed as the ones below.)
- **Example 5.1 (Human Evolution).**



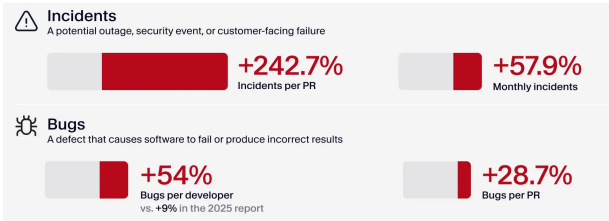
https://openclipart.org/image/800px/348815.png

Interesting GitHub study: AI Acceleration Whiplash [AER2626]

- Faros AI 2026 Engineering Report: A study on GitHub telemetry data from the last two years.
  - The productivity is up . . .



  - . . . while code quality is down!



Ten takeaways from the Acceleration Whiplash report (from [AER2626])

1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.

Ten takeaways from the Acceleration Whiplash report (from [AER2626])

1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.
5. Bugs are accelerating, not stabilizing.

Ten takeaways from the Acceleration Whiplash report (from [AER2626])

1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.
5. Bugs are accelerating, not stabilizing.
6. AI made it easy to start work. It did not make it easy to finish it.

Ten takeaways from the Acceleration Whiplash report (from [AER2626])

1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.
5. Bugs are accelerating, not stabilizing.
6. AI made it easy to start work. It did not make it easy to finish it.
7. The most experienced people in your organization are being buried. We call it the senior engineer tax.

Ten takeaways from the Acceleration Whiplash report (from [AER2626])

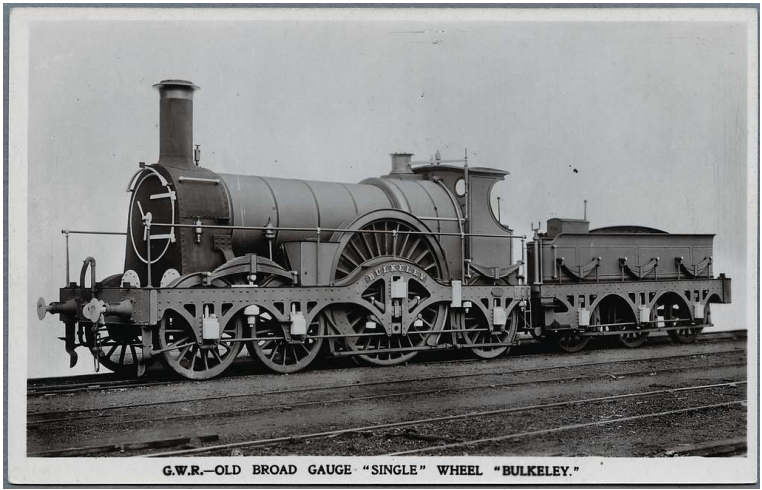
1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.
5. Bugs are accelerating, not stabilizing.
6. AI made it easy to start work. It did not make it easy to finish it.
7. The most experienced people in your organization are being buried. We call it the senior engineer tax.
8. More code is entering production with no review at all.

Ten takeaways from the Acceleration Whiplash report (from [AER2626])

1. AI crossed a threshold. It is now the primary author of code.
2. The business value is real. Roadmaps are finally moving.
3. But the throughput numbers have an asterisk.
4. For every code change merged, the probability of a production incident has more than tripled.
5. Bugs are accelerating, not stabilizing.
6. AI made it easy to start work. It did not make it easy to finish it.
7. The most experienced people in your organization are being buried. We call it the senior engineer tax.
8. More code is entering production with no review at all.
9. Strong engineering foundations do not protect you. Two years of telemetry says so.

But Steam Engines also had Problems in the Industrial Revolution

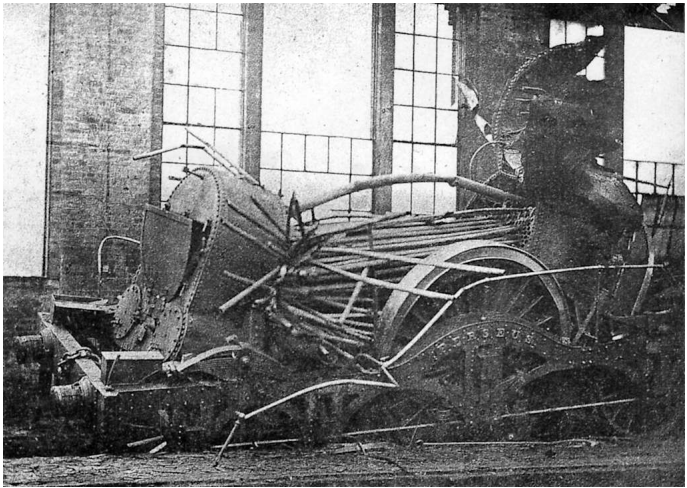
- **Example 5.2 (Steam Engines Problem/Solution).**



https://picryl.com/media/great-western-railway-gwr-bulkeley-1e42f7

But Steam Engines also had Problems in the Industrial Revolution

- **Example 5.2 (Steam Engines Problem/Solution).**

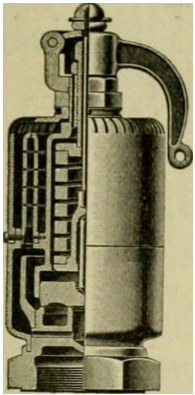


https://picryl.com/media/1862-westbourne-park-ada93a

But Steam Engines also had Problems in the Industrial Revolution

- **Example 5.2 (Steam Engines Problem/Solution).**

Problem: Papin's safety valve from 1679 was not robust/safe enough for steam locomotives
Solution: Ramsbottom's spring-loaded safety valve from 1856 was $\rightsquigarrow$ safety!

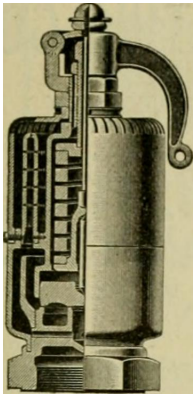


https://www.flickr.com/photos/internetarchivebookimages/14737324276/

But Steam Engines also had Problems in the Industrial Revolution

- **Example 5.2 (Steam Engines Problem/Solution).**

Problem: Papin's safety valve from 1679 was not robust/safe enough for steam locomotives
Solution: Ramsbottom's spring-loaded safety valve from 1856 was $\rightsquigarrow$ safety!



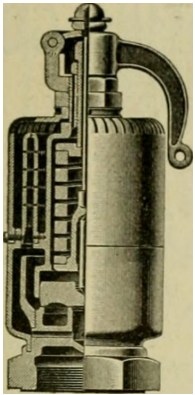
https://www.flickr.com/photos/internetarchivebookimages/14737324276/

- **Important Question:** What are the safety-valves for the LLM Revolution?

But Steam Engines also had Problems in the Industrial Revolution

- **Example 5.2 (Steam Engines Problem/Solution).**

Problem: Papin's safety valve from 1679 was not robust/safe enough for steam locomotives
Solution: Ramsbottom's spring-loaded safety valve from 1856 was $\rightsquigarrow$ safety!



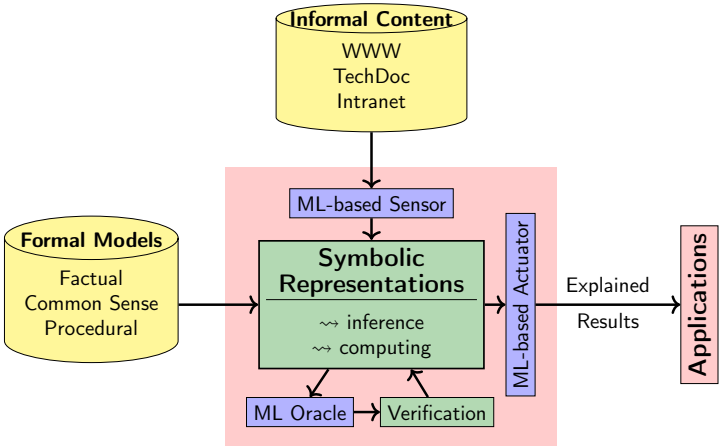
https://www.flickr.com/photos/internetarchivebookimages/14737324276/

- **Important Question:** What are the safety-valves for the LLM Revolution?
- **Observation/Caveat:** Monolithic systems (one technology end-to-end) were never the solution in engineering!

6 Symbolic Core and Other Safety Valves

The Symbolic Core Architecture (Keep SC/ML Safe in spite of AI)

- **Definition 6.1.** An AI system is called a symbolic core system, if it uses symbolic representations, symbolic computation, and inference at the core to produce results with explanations. It may use other forms of AI technology to perform sensory and actuatory tasks at the periphery of the system or as oracles under control by a symbolic verification subsystem.



Reinvesting the AI Acceleration into Formality

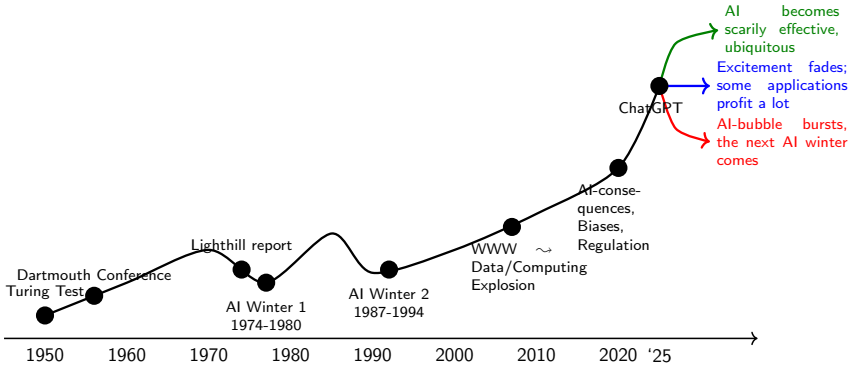
- **Idea:** If there is indeed an AI acceleration effect (productivity increase), let us invest that into formality!
- Symbolic representations are formal! Formal models/libraries can be re-used.
- Agentic loops work best if we have formal specifications and working verification (as in Rust).
- Specification is at the center of this enterprise. (But many other aspects involved $\rightsquigarrow$ see below)

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)



Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).
- **My Conclusion:** Instead of just playing along with the ML/LLM hype (gadget lust), we should think about consequences of whole-sale adoption of ML/LLM
  - what are long-term consequences of cognitive laziness (addiction, dependency, loss of competencies)
  - how do we have to change education given that assessment often fails. (gradable competency proxies easily solved by LLMs)
  - can we save the scientific publication system? (rescind "publish or perish"?)

Conclusion: How to think about the ML/LLM Hype

- **Scientifically** I am not really sure what to think of ML/LLM methods $\rightsquigarrow$ no predictions.
- **Economically** It seems clear that we are in a bubble, which might (or not) burst after the Anthropic/OpenAI IPOs! (again, no predictions about the "plateau of productivity")
  - There is just waaay to much money in the system. (generated by overblown predictions/promises)
  - There is no "technology with limitless, ever-accelerating potential". (cf. the dot-com-bubble of 2000)
- ⚠ **Observation:** All of the promises are about "efficiency" (reducing effort) rather than quality (often dubious).
- **My Conclusion:** Instead of just playing along with the ML/LLM hype (gadget lust), we should think about consequences of whole-sale adoption of ML/LLM
  - what are long-term consequences of cognitive laziness (addiction, dependency, loss of competencies)
  - how do we have to change education given that assessment often fails. (gradable competency proxies easily solved by LLMs)
  - can we save the scientific publication system? (rescind "publish or perish"?)
- **To end on a positive note:** In Math and Symbolic Computation we are still relatively safe, as we have relatively robust quality criteria. (invest into formalization!)