# LLMs, Computational Theory, and Redux: New Directions for CC in Computational Complexity

**Paul Bodily**
Computer Science Department
Idaho State University
Pocatello, ID 83209 USA
bodipaul@isu.edu

## Abstract

In 2022, we identified several open CC problems in computational complexity theory (CCT), arguing that generating novel NP-complete reductions and constructing gadgets represent genuine creative challenges (Bodily and Ventura 2022). At the time, automated code generation was already a recognized CC subproblem, with Colton and colleagues investigating the creative dimensions of automatic software synthesis (Colton, Powley, and Cook 2018; Colton et al. 2019). Much has changed. The rapid rise of large language models (LLMs) capable of producing syntactically correct, functionally reasonable code has shifted the question from *whether* machines can write code to *what kinds* of code writing require genuine creative capacity. Problems in computational theory, we argue, are precisely the kind that do.

This abstract reports on work in progress connecting three threads: (1) the open CCT problems identified in our prior work, (2) Redux, an interactive knowledgebase of NP-complete problems, reductions, and solution algorithms (Marchetti et al. 2024), and (3) a graduate course, CS 6673: Advanced Computational Creativity, to be offered this Fall, in which students will explore LLMs as creative agents within the Redux ecosystem. Together, these threads form the basis for a focused investigation into where LLM-based code generation succeeds, where it struggles, and whether its outputs can meaningfully be called creative.

### Redux as a Testbed

Redux (Marchetti et al. 2024) is a web-accessible, open-source platform with interactive visualizations of NP-complete problems, reductions, and solution algorithms—a natural testbed for LLM creativity experiments. The course will investigate the following tasks:

- **Automated problem instance generation**: Can LLMs produce instances that highlight cases where algorithms perform well versus poorly, spanning structural properties (e.g., sparse vs. dense graphs)?
- **Natural language to Redux-formatted instances**: Can LLMs parse informal problem descriptions and produce correctly formatted Redux-compatible encodings?
- **Problem and solver recommendation**: Can LLMs identify which Redux problems and solvers best match a user's real-world use case?
- **Redux Live AI Learning Assistant**: Can an LLM guide learners step-by-step through algorithm and reduction visualizations and suggest what to explore next?
- **Novel vs. existing reductions**: Do LLMs reproduce known reductions, or can they discover structurally novel ones?
- **Reduction proofs**: Can LLMs construct valid correctness proofs for polynomial-time reductions?

### The Creativity Question

Some may question whether these tasks require creativity at all. We argue they do. Foundational frameworks define creativity by appeal to novelty, value, and surprise (Boden 2004), with intentionality as a further criterion (Colton and Wiggins 2012) formalized in prior work (Bodily and Ventura 2022). Each task above demands all four. Generating an instance that exposes subtle algorithmic behavior requires something *novel* (not a stock textbook example), *valuable* (pedagogically informative), *intentional* (targeting a specific property), and potentially *surprising*. Generating a correct and novel NP-complete reduction is, if anything, more demanding—requiring both structural invention and formal proof.

### Broader Significance

This work extends the agenda of applying CC to scientific and mathematical domains (Pease et al. 2019). Redux provides a structured, reproducible experimental setting with clear evaluation criteria, while the graduate course ensures human expert judgment on correctness, novelty, and explanatory value. Results will speak directly to theoretical questions about the limits of LLM creativity in formal domains—questions of broad interest to the TCS & CC community.

## References

Bodily, P., and Ventura, D. 2022. Open computational creativity problems in computational theory. In *Proceedings of the 13th International Conference on Computational Creativity (ICCC 2022)*, 146–153.

Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*, 2nd ed. Routledge, London.

Colton, S.; Powley, E. J.; and Cook, M. 2018. Investigating and automating the creative act of software engineering: A position paper. In *Proceedings of the 9th International Conference on Computational Creativity (ICCC 2018)*, 224–231.

Colton, S.; Pease, A.; Cook, M.; and Chen, C. 2019. The HR3 system for automatic code generation in creative settings. In *Proceedings of the 10th International Conference on Computational Creativity (ICCC 2019)*, 108–115.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, 21–26.

Marchetti, K.; Sevaljevic, A.; Diviney, A.; Eardley, C.; Phillips, R.; Khadka, R.; Igbokwe, D.; and Bodily, P. 2024. Redux: An interactive, dynamic knowledge base for teaching NP-completeness. In *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education (ITiCSE 2024)*, 255–261. ACM.

Pease, A.; Colton, S.; Warburton, C.; Nathanail, A.; Preda, I.; Arnold, D.; Winterstein, D.; and Cook, M. 2019. The importance of applying computational creativity to scientific and mathematical domains. In *Proceedings of the 10th International Conference on Computational Creativity (ICCC 2019)*, 250–257.