REFERENCES

[1] Gul Agha and Karl Palmskog. 2018. A survey of statistical model checking. *ACM Transactions on Modeling and Computer Simulation* 28, 1 (2018), 6:1–6:39. https://doi.org/10.1145/3158668

[2] Nasir Ali, Yann-Gaël Guéhéneuc, and Giuliano Antoniol. 2013. Trustrace: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links. *IEEE Transactions on Software Engineering* 39, 5 (2013), 725–741. https://doi.org/10.1109/TSE.2012.71

[3] Xavier Amatriain. 2024. Prompt Design and Engineering: Introduction and Advanced Methods. (2024). arXiv:cs.SE/2401.14423 https://arxiv.org/abs/2401.14423

[4] Chetan Arora, John Grundy, and Mohamed Abdelrazek. 2023. Advancing Requirements Engineering through Generative AI: Assessing the Role of LLMs. (2023). arXiv:cs.SE/2310.13976 https://arxiv.org/abs/2310.13976

[5] Chetan Arora, Mehrdad Sabetzadeh, Lionel Briand, and Frank Zimmer. 2016. Extracting domain models from natural-language requirements: approach and industrial evaluation. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS '16)*. Association for Computing Machinery, New York, NY, USA, 250–260. https://doi.org/10.1145/2976767.2976769

[6] Erika Asnina, Bernards Gulbis, Janis Osis, Gundars Alksnis, Uldis Donins, and Armands Slihte. 2011. Backward Requirements Traceability within the Topology-based Model Driven Software Development. In *MDA & MDSD 2011 - Proceedings of the 3rd International Workshop on Model-Driven Architecture and Modeling-Driven Software Development, In conjunction with ENASE 2011, Beijing, China, June 2011*, Janis Osis and Oksana Nikiforova (Eds.). SciTePress, 36–45.

[7] Arshad Beg, Diarmuid O'Donoghue, and Rosemary Monahan. 2025. Formalising Software Requirements using Large Language Models. (2025). arXiv:cs.SE/2506.10704 https://arxiv.org/abs/2506.10704

[8] Arshad Beg, Diarmuid O'Donoghue, and Rosemary Monahan. 2025. A Short Survey on Formalising Software Requirements using Large Language Models. (2025). arXiv:cs.SE/2506.11874 https://arxiv.org/abs/2506.11874

[9] Ron Bell. 2006. Introduction to IEC 61508. In *Proceedings of the 10th Australian Workshop on Safety Critical Systems and Software - Volume 55 (SCS '05)*. Australian Computer Society, Inc., AUS, 3–12.

[10] Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwasniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Onur Mutlu, and Torsten Hoefler. 2024. Topologies of Reasoning: Demystifying Chains, Trees, and Graphs of Thoughts. *CoRR* abs/2401.14295 (2024). https://doi.org/10.48550/arXiv.2401.14295 arXiv:2401.14295

[11] Benjamin Brosgol. 2011. Do-178c: the next avionics safety standard. *Ada Lett.* 31, 3 (Nov. 2011), 5–6. https://doi.org/10.1145/2070336.2070341

[12] Andrew Butterfield and Frédéric Tuong. 2023. Applying Formal Verification to an Open-Source Real-Time Operating System. In *Theories of Programming and Formal Methods - Essays Dedicated to Jifeng He on the Occasion of His 80th Birthday (Lecture Notes in Computer Science)*, Jonathan P. Bowen, Qin Li, and Qiwen Xu (Eds.), Vol. 14080. Springer, 348–366. https://doi.org/10.1007/978-3-031-40436-8_13

[13] Bora Caglayan, Mingxue Wang, John D. Kelleher, Shen Fei, Gui Tong, Jiandong Ding, and Puchao Zhang. 2024. BIS: NL2SQL Service Evaluation Benchmark for Business Intelligence Scenarios. In *Service-Oriented Computing: 22nd International Conference, ICSOC 2024, Tunis, Tunisia, December 3–6, 2024, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, 357–372. https://doi.org/10.1007/978-981-96-0808-9_27

[14] Marco Casadio, Tanvi Dinkar, Ekaterina Komendantskaya, Luca Arnaboldi, Matthew L. Daggitt, Omri Isac, Guy Katz, Verena Rieser, and Oliver Lemon. 2025. NLP Verification: Towards a General Methodology for Certifying Robustness. (2025). arXiv:cs.CL/2403.10144 https://arxiv.org/abs/2403.10144

[15] Farid Cerbah and Jérôme Euzenat. 2001. Using Terminology Extraction to Improve Traceability from Formal Models to Textual Requirements. In *Natural Language Processing and Information Systems*, Mokrane Bouzeghoub, Zoubida Kedad, and Elisabeth Métais (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 115–126.

[16] J. Cleland-Huang, C.K. Chang, and M. Christensen. 2003. Event-based traceability for managing evolutionary change. *IEEE Transactions on Software Engineering* 29, 9 (2003), 796–810. https://doi.org/10.1109/TSE.2003.1232285

[17] Matthias Cosler, Christopher Hahn, Daniel Mendoza, Frederik Schmitt, and Caroline Trippel. 2023. nl2spec: Interactively Translating Unstructured Natural Language to Temporal Logics with Large Language Models. (2023). arXiv:cs.LO/2303.04864 https://arxiv.org/abs/2303.04864

[18] Juan Ortiz Couder, Dawson Gomez, and Omar Ochoa. 2024. Requirements Verification Through the Analysis of Source Code by Large Language Models. In *SoutheastCon 2024*. 75–80. https://doi.org/10.1109/SoutheastCon52093.2024.10500073

[19] Ana Marcia Debiasi Duarte, Denio Duarte, and Marcello Thiry. 2016. TraceBoK: Toward a Software Requirements Traceability Body of Knowledge. In *2016 IEEE 24th International Requirements Engineering Conference (RE)*. 236–245. https://doi.org/10.1109/RE.2016.32

[20] Elicit. 2025. Elicit - The AI Research Assistant. https://elicit.com. (2025). Accessed: 2025-04-11 at 12:49PM.

[21] Michael D. Ernst. 2017. Natural Language is a Programming Language: Applying Natural Language Processing to Software Development. In *2nd Summit on Advances in Programming Languages (SNAPL 2017) (Leibniz International Proceedings in Informatics (LIPIcs))*, Benjamin S. Lerner, Rastislav Bodík, and Shriram Krishnamurthi (Eds.), Vol. 71. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 4:1–4:14. https://doi.org/10.4230/LIPIcs.SNAPL.2017.4

[22] Wen Fan, Marilyn Rego, Xin Hu, Sanya Dod, Zhaorui Ni, Danning Xie, Jenna DiVincenzo, and Lin Tan. 2025. Evaluating the Ability of Large Language Models to Generate Verifiable Specifications in VeriFast. (2025). arXiv:cs.SE/2411.02318 https://arxiv.org/abs/2411.02318

[23] Wenji Fang, Mengming Li, Min Li, Zhiyuan Yan, Shang Liu, Hongce Zhang, and Zhiyao Xie. 2024. AssertLLM: Generating Hardware Verification Assertions from Design Specifications via Multi-LLMs. In *2024 IEEE LLM Aided Design Workshop (LAD)*. 1–1. https://doi.org/10.1109/LAD62341.2024.10691792

[24] Mohamad Fazelnia, Mehdi Mirakhorli, and Hamid Bagheri. 2024. Translation Titans, Reasoning Challenges: Satisfiability-Aided Language Models for Detecting Conflicting Requirements. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24)*. Association for Computing Machinery, New York, NY, USA, 2294–2298. https://doi.org/10.1145/3691620.3695302

[25] Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. 2011. Run-time efficient probabilistic model checking. In *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011*, Richard N. Taylor, Harald C. Gall, and Nenad Medvidovic (Eds.). ACM, 341–350. https://doi.org/10.1145/1985793.1985840

[26] Simon Foster, James Baxter, Ana Cavalcanti, Jim Woodcock, and Frank Zeyda. 2020. Unifying semantic foundations for automated verification tools in Isabelle/UTP. *Science of Computer Programming* 197 (2020), 102510. https://doi.org/10.1016/j.scico.2020.102510

[27] Saurabh Gadia, Cyrille Artho, and Gedare Bloom. 2016. Verifying Nested Lock Priority Inheritance in RTEMS with Java Pathfinder. In *Formal Methods and Software Engineering*, Kazuhiro Ogata, Mark Lawford, and Shaoying Liu (Eds.). Springer International Publishing, Cham, 417–432. https://doi.org/10.1007/978-3-319-47846-3_26

[28] Marie-Claude Gaudel. 1995. Testing can be formal, too. In *TAPSOFT'95: Theory and Practice of Software Development, 6th International Joint Conference CAAP/FASE (Lecture Notes in Computer Science)*, Peter D. Mosses, Mogens Nielsen, and Michael I. Schwartzbach (Eds.), Vol. 915. Springer, 82–96. https://doi.org/10.1007/3-540-59293-8_188

[29] E. C. Genvigir and N. L. Vijaykumar. 2010. Requirements Traceability. In *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization*, Muthu Ramachandran and Ricardo de Carvalho (Eds.). IGI Global Scientific Publishing, 102–120. https://doi.org/10.4018/978-1-60566-731-7.ch008

[30] Vincenzo Gervasi and Bashar Nuseibeh. 2002. Lightweight validation of natural language requirements. *Softw. Pract. Exper.* 32, 2 (Feb. 2002), 113–133. https://doi.org/10.1002/spe.430

[31] Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, and Wilfried Steiner. 2016. ARSENAL: Automatic Requirements Specification Extraction from Natural Language. In *NASA Formal Methods*, Sanjai Rayadurgam and Oksana Tkachuk (Eds.). Springer International Publishing, Cham, 41–46.

[32] Joseph A. Goguen and Rod M. Burstall. 1992. Institutions: abstract model theory for specification and programming. *J. ACM* 39, 1 (Jan. 1992), 95–146. https://doi.org/10.1145/147508.147524

[33] Arda Goknil, Ivan Kurtev, Klaas van den Berg, and Jan-Willem Veldhuis. 2011. Semantics of trace relations in requirements models for consistency checking and inferencing. *Software & Systems Modeling* 10, 1 (2011), 31–54. https://doi.org/10.1007/s10270-009-0142-3

[34] George Granberry, Wolfgang Ahrendt, and Moa Johansson. 2025. Specify What? Enhancing Neural Specification Synthesis by Symbolic Methods. In *Integrated Formal Methods*, Nikolai Kosmatov and Laura Kovács (Eds.). Springer Nature Switzerland, Cham, 307–325.

[35] George Granberry, Wolfgang Ahrendt, and Moa Johansson. 2025. Towards Integrating Copiloting and Formal Methods. In *Leveraging Applications of Formal Methods, Verification and Validation. Specification and Verification*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer Nature Switzerland, Cham, 144–158.

[36] Sol J. Greenspan, Alexander Borgida, and John Mylopoulos. 1986. A requirements modeling language and its logic. *Information Systems* 11, 1 (1986), 9–23. https://doi.org/10.1016/0306-4379(86)90020-7

[37] Jin Guo, Jinghui Cheng, and Jane Cleland-Huang. 2017. Semantically enhanced software traceability using deep learning techniques. In *Proceedings of the 39th International Conference on Software Engineering (ICSE '17)*. IEEE Press, 3–14. https://doi.org/10.1109/ICSE.2017.9

[38] Christopher Hahn, Frederik Schmitt, Julia J. Tillman, Niklas Metzger, Julian Siber, and Bernd Finkbeiner. 2022. Formal Specifications from Natural Language. (2022). arXiv:cs.SE/2206.01962 https://arxiv.org/abs/2206.01962

[39] Abigail Hammer, Matthew Cauwels, Benjamin Hertz, Phillip H. Jones, and Kristin Y. Rozier. 2022. Integrating runtime verification into an automated UAS traffic management system. *Innov. Syst. Softw. Eng.* 18, 4 (2022), 567–580. https://doi.org/10.1007/S11334-021-00407-5

[40] Jane Huffman Hayes, Alex Dekhtyar, Senthil Karthikeyan Sundaram, E. Ashlee Holbrook, Sravanthi Vadlamudi, and Alain April. 2007. REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *Innovations in Systems and Software Engineering* 3, 3 (September 2007), 193–202. https://doi.org/10.1007/s11334-007-0024-1

[41] Robert M. Hierons, Kirill Bogdanov, Jonathan P. Bowen, Rance Cleaveland, John Derrick, Jeremy Dick, Marian Gheorghe, Mark Harman, Kalpesh Kapoor, Paul J. Krause, Gerald Lüttgen, Anthony J. H. Simons, Sergiy A. Vilkomir, Martin R. Woodward, and Hussein Zedan. 2009. Using formal specifications to support testing. *ACM Comput. Surv.* 41, 2, 9:1–9:76. https://doi.org/10.1145/1459352.1459354

[42] Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James R. Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. Found in the middle: Calibrating Positional Attention Bias Improves Long Context Utilization. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 14982–14995. https://doi.org/10.18653/V1/2024.FINDINGS-ACL.890

[43] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[44] Marieke Huisman, Dilian Gurov, and Alexander Malkis. 2024. Formal Methods: From Academia to Industrial Practice. A Travel Guide. (2024). arXiv:cs.SE/2002.07279 https://arxiv.org/abs/2002.07279

[45] Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. 2004. Approximate probabilistic model checking. In *Verification, Model Checking, and Abstract Interpretation, 5th International Conference, VMCAI 2004 (Lecture Notes in Computer Science)*, Bernhard Steffen and

Giorgio Levi (Eds.), Vol. 2937. Springer, 73–84. https://doi.org/10.1007/978-3-540-24622-0_8

[46] Padma Iyenghar, Elke Pulvermueller, and Clemens Westerkamp. 2011. Towards Model-Based Test automation for embedded systems using UML and UTP. In *ETFA2011*. 1–9. https://doi.org/10.1109/ETFA.2011.6058982

[47] Albert Qiaochu Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Mił oś, Yuhuai Wu, and Mateja Jamnik. 2022. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 8360–8373.

[48] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 22199–22213.

[49] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. 2002. PRISM: probabilistic symbolic model checker. In *Computer Performance Evaluation, Modelling Techniques and Tools, 12th International Conference, TOOLS 2002 (Lecture Notes in Computer Science)*, Tony Field, Peter G. Harrison, Jeremy T. Bradley, and Uli Harder (Eds.), Vol. 2324. Springer, 200–204. https://doi.org/10.1007/3-540-46029-2_13

[50] Iat Tou Leong and Raul Barbosa. 2023. Translating Natural Language Requirements to Formal Specifications: A Study on GPT and Symbolic NLP. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. 259–262. https://doi.org/10.1109/DSN-W58399.2023.00065

[51] Henrik Leopold, Jan Mendling, and Artem Polyvyanyy. 2014. Supporting Process Model Validation through Natural Language Generation. *IEEE Transactions on Software Engineering* 40, 8 (2014), 818–840. https://doi.org/10.1109/TSE.2014.2327044

[52] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[53] Jia Li, Ge Li, Yongmin Li, and Zhi Jin. 2025. Structured Chain-of-Thought Prompting for Code Generation. *ACM Trans. Softw. Eng. Methodol.* 34, 2, Article 37 (Jan. 2025), 23 pages. https://doi.org/10.1145/3690635

[54] Mengming Li, Wenji Fang, Qijun Zhang, and Zhiyao Xie. 2024. SpecLLM: Exploring Generation and Review of VLSI Design Specification with Large Language Model. (2024). arXiv:cs.AR/2401.13266 https://arxiv.org/abs/2401.13266

[55] Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024. One-Shot Learning as Instruction Data Prospector for Large Language Models. (2024). arXiv:cs.CL/2312.10302 https://arxiv.org/abs/2312.10302

[56] Matt Luckcuck, Marie Farrell, Louise A. Dennis, Clare Dixon, and Michael Fisher. 2019. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. *ACM Comput. Surv.* 52, 5, Article 100 (Sept. 2019), 41 pages. https://doi.org/10.1145/3342355

[57] Lezhi Ma, Shangqing Liu, Yi Li, Xiaofei Xie, and Lei Bu. 2024. SpecGen: Automated Generation of Formal Program Specifications via Large Language Models. (2024). arXiv:cs.SE/2401.08807 https://arxiv.org/abs/2401.08807

[58] Abdulkadir Ahmad Madaki and Wan Mohd Nazmee Wan Zainon. 2022. A Review on Tools and Techniques for Visualizing Software Requirement Traceability. In *Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications*, Nor Muzlifah Mahyuddin, Nor Rizuan Mat Noor, and Harsa Amylia Mat Sakim (Eds.). Springer Singapore, Singapore, 39–44.

[59] Shantanu Mandal, Adhrik Chethan, Vahid Janfaza, S M Farabi Mahmud, Todd A Anderson, Javier Turek, Jesmin Jahan Tithi, and Abdullah Muzahid. 2023. Large Language Models Based Automatic Synthesis of Software Specifications. (2023). arXiv:cs.SE/2304.09181 https://arxiv.org/abs/2304.09181

[60] Md Rakib Hossain Misu, Cristina V. Lopes, Iris Ma, and James Noble. 2024. Towards AI-Assisted Synthesis of Verified Dafny Methods. *Proc. ACM Softw. Eng.* 1, FSE, Article 37 (July 2024), 24 pages. https://doi.org/10.1145/3643763

[61] Eric Mugnier, Emmanuel Anaya Gonzalez, Ranjit Jhala, Nadia Polikarpova, and Yuanyuan Zhou. 2024. Laurel: Generating Dafny Assertions Using Large Language Models. (2024). arXiv:cs.LO/2405.16792 https://arxiv.org/abs/2405.16792

[62] Prasita Mukherjee and Benjamin Delaware. 2024. Towards Automated Verification of LLM-Synthesized C Programs. (2024). arXiv:cs.PL/2410.14835 https://arxiv.org/abs/2410.14835

[63] Anmol Nayak, Hari Prasad Timmapathini, Vidhya Murali, Karthikeyan Ponnalagu, Vijendran Gopalan Venkoparao, and Amalinda Post. 2022. Req2Spec: Transforming Software Requirements into Formal Specifications Using Natural Language Processing. In *Requirements Engineering: Foundation for Software Quality: 28th International Working Conference, REFSQ 2022, Birmingham, UK, March 21–24, 2022, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 87–95.

[64] Erika Nazaruka and J?nis Osis. 2018. Determination of Natural Language Processing Tasks and Tools for Topological Functioning Modelling. In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)*. SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT, 501–512. https://doi.org/10.5220/0006817205010512

[65] Sabina-Cristiana Necula, Florin Dumitriu, and Valerică Greavu-Şerban. 2024. A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. *Electronics* 13, 11 (2024). https://doi.org/10.3390/electronics13112055

[66] Rani Nelken and Nissim Francez. 1996. Automatic translation of natural language system specifications into temporal logic. In *Computer Aided Verification*, Rajeev Alur and Thomas A. Henzinger (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 360–371.

[67] Ali Nouri, Beatriz Cabrero-Daniel, Fredrik Törner, Håkan Sivencrona, and Christian Berger. 2024. Engineering Safety Requirements for Autonomous Driving with Large Language Models. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*. 218–228. https://doi.org/10.1109/RE59067.2024.00029

[68] Wiktor Nowakowski, Michał Śmiałek, Albert Ambroziewicz, and Tomasz Straszak. 2013. Requirements-level language and tools for capturing software system essence. *Computer Science and Information Systems* 10, 4 (2013), 1499–1524.

[69] M. Osborne and C.K. MacNish. 1996. Processing natural language software requirement specifications. In *Proceedings of the Second International Conference on Requirements Engineering*. 229–236. https://doi.org/10.1109/ICRE.1996.491451

[70] Rob Palin, David Ward, Ibrahim Habli, and Roger Rivett. 2011. ISO 26262 safety cases: Compliance and assurance. In *6th IET International Conference on System Safety 2011*. 1–6. https://doi.org/10.1049/cp.2011.0251

[71] F.A.C. Pinheiro and J.A. Goguen. 1996. An object-oriented tool for tracing requirements. In *Proceedings of the Second International Conference on Requirements Engineering*. 219–. https://doi.org/10.1109/ICRE.1996.491449

[72] Anamaria-Roberta Preda, Christoph Mayr-Dorn, Atif Mashkoor, and Alexander Egyed. 2024. Supporting High-Level to Low-Level Requirements Coverage Reviewing with Large Language Models. In *Proceedings of the 21st International Conference on Mining Software Repositories (MSR '24)*. Association for Computing Machinery, New York, NY, USA, 242–253. https://doi.org/10.1145/3643991.3644922

[73] Xin Quan, Marco Valentino, Louise A. Dennis, and André Freitas. 2024. Verification and Refinement of Natural Language Explanations through LLM-Symbolic Theorem Proving. (2024). arXiv:cs.CL/2405.01379 https://arxiv.org/abs/2405.01379

[74] Lasse M. Reinpold, Marvin Schieseck, Lukas P. Wagner, Felix Gehlhoff, and Alexander Fay. 2024. Exploring LLMs for Verifying Technical System Specifications Against Requirements. (2024). arXiv:cs.SE/2411.11582 https://arxiv.org/abs/2411.11582

[75] Patrick Rempel and Parick Mäder. 2017. Preventing Defects: The Impact of Requirements Traceability Completeness on Software Quality. *IEEE Transactions on Software Engineering* 43, 8 (2017), 777–797. https://doi.org/10.1109/TSE.2016.2622264

[76] A.M. Salem. 2006. Improving Software Quality through Requirements Traceability Models. In *IEEE International Conference on Computer Systems and Applications, 2006.* 1159–1162. https://doi.org/10.1109/AICCSA.2006.205236

[77] Ahmed M Salem. 2010. Model for Enhancing Requirements Traceability and Analysis. *International Journal of Advanced Computer Science and Applications* 1, 5 (2010). https://doi.org/10.14569/IJACSA.2010.010503

[78] Sabnam Sengupta, Ananya Kanjilal, and Swapan Bhattacharya. 2008. Requirement Traceability in Software Development Process: An Empirical Approach. In *2008 The 19th IEEE/IFIP International Symposium on Rapid System Prototyping*. 105–111. https://doi.org/10.1109/RSP.2008.14

[79] Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 12113–12139. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.811

[80] Xujie Si, Aaditya Naik, Hanjun Dai, Mayur Naik, and Le Song. 2020. Code2Inv: A Deep Learning Framework for Program Verification. In *Computer Aided Verification*, Shuvendu K. Lahiri and Chao Wang (Eds.). Springer International Publishing, Cham, 151–164.

[81] Norbert Tihanyi, Ridhi Jain, Yiannis Charalambous, Mohamed Amine Ferrag, Youcheng Sun, and Lucas C. Cordeiro. 2024. A New Era in Software Security: Towards Self-Healing Software via Large Language Models and Formal Verification. (2024). arXiv:cs.SE/2305.14752 https://arxiv.org/abs/2305.14752

[82] RICHARD TORKAR, TONY GORSCHEK, ROBERT FELDT, MIKAEL SVAHNBERG, UZAIR AKBAR RAJA, and KASHIF KAMRAN. 2012. REQUIREMENTS TRACEABILITY: A SYSTEMATIC REVIEW AND INDUSTRY CASE STUDY. *International Journal of Software Engineering and Knowledge Engineering* 22, 03 (2012), 385–433. https://doi.org/10.1142/S021819401250009X arXiv:https://doi.org/10.1142/S021819401250009X

[83] Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively Prompt Pre-trained Language Models for Chain of Thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2714–2730. https://doi.org/10.18653/v1/2022.emnlp-main.174

[84] Bangchao Wang, Rong Peng, Zhuo Wang, Xiaomin Wang, and Yuanbang Li. 2020. An Automated Hybrid Approach for Generating Requirements Trace Links. *International Journal of Software Engineering and Knowledge Engineering* 30, 07 (2020), 1005–1048. https://doi.org/10.1142/S0218194020500278 arXiv:https://doi.org/10.1142/S0218194020500278

[85] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837.

[86] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *CoRR* abs/2302.11382 (2023). https://doi.org/10.48550/ARXIV.2302.11382 arXiv:2302.11382

[87] Jim Woodcock and Ana Cavalcanti. 2002. The Semantics of Circus. In *ZB 2002: Formal Specification and Development in Z and B*, Didier Bert, Jonathan P. Bowen, Martin C. Henson, and Ken Robinson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 184–203. https://doi.org/10.1007/3-540-45648-1_10

[88] Jim Woodcock and Ana Cavalcanti. 2004. A Tutorial Introduction to Designs in Unifying Theories of Programming. In *Integrated Formal Methods*, Eerke A. Boiten, John Derrick, and Graeme Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 40–66.

[89] Haoze Wu, Clark Barrett, and Nina Narodytska. 2024. Lemur: Integrating Large Language Models in Automated Program Verification. (2024). arXiv:cs.FL/2310.04870 https://arxiv.org/abs/2310.04870

[90] Weijia Xu, Andrzej Banburski-Fahey, and Nebojsa Jojic. 2023. Reprompting: Automated Chain-of-Thought Prompt Inference Through Gibbs Sampling. *CoRR* abs/2305.09993 (2023). https://doi.org/10.48550/ARXIV.2305.09993 arXiv:2305.09993

[91] Yilongfei Xu, Jincao Feng, and Weikai Miao. 2024. Learning from Failures: Translation of Natural Language Requirements into Linear Temporal Logic with Large Language Models. In *2024 IEEE 24th International Conference on Software Quality, Reliability and Security (QRS)*. 204–215. https://doi.org/10.1109/QRS62785.2024.00029

[92] Rongjie Yan, Chih-Hong Cheng, and Yesheng Chai. 2015. Formal consistency checking over specifications in natural languages. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1677–1682.

[93] Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J Prenger, and Animashree Anandkumar. 2023. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 21573–21612.

[94] Junyi Yao, Yijiang Liu, Zhen Dong, Mingfei Guo, Helan Hu, Kurt Keutzer, Li Du, Daquan Zhou, and Shanghang Zhang. 2024. PromptCoT: Align Prompt Distribution via Adapted Chain-of-Thought. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7027–7037. https://doi.org/10.1109/CVPR52733.2024.00671

[95] Xi Ye and Greg Durrett. 2022. The Unreliability of Explanations in Few-shot Prompting for Textual Reasoning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 30378–30392.

[96] Haodi Zhang, Min Cai, Xinhe Zhang, Chen Jason Zhang, Rui Mao, and Kaishun Wu. 2023. Self-Convinced Prompting: Few-Shot Question Answering with Repeated Introspection. (2023). arXiv:cs.CL/2310.05035 https://arxiv.org/abs/2310.05035