MeTTaMath: Integrating Formal Verification into an AGI Cognitive Architecture via the MeTTa language

Zarathustra Amadeus Goertzel

Czech Technical University in Prague, Czech Republic

Introduction: This abstract presents the project to integrate sound, verifiable reasoning into the core of the Hyperon AGI system [7]. The first completed exploration is the implementation of a Metamath [15] verifier in the MeTTa [16]. The Hyperon architecture is designed to foster *cognitive synergy* among diverse AI components such as neural networks and symbolic reasoners, utilizing a shared representation for cognitive and procedural knowledge. MeTTa[1] (Meta Type Talk) is the language designed for this purpose. MeTTa is a gradually-typed language with elements of declarative and functional programming, and as a *homoiconic* language, its programs can be natively treated as data. The core semantics depend on *pattern matching* and *rewriting* over metagraphs (called atompsaces) [6]. Metamath is a language for formal verification in terms of token-level *string substitution*, respecting typecodes and disjoint variable constraints (as a way to deal with quantifier scopes)[2]. Metamath was chosen for its apparent alignment with MeTTa and the small verifier.

Integrating verified reasoning directly into AGI systems may be crucial for building robust and trustworthy AI with *sound reasoning*. While it is possible to outsource verification to external systems, my experiencing developing a proof-of concept formal meta-ethics ontology in SUMO [10, 12] highlighted the importance of having an in-house trusted proof kernel within a knowledge system. The recent success of DeepMind achieving Gold scores in the IMO [13] suggests that LLM-based AIs' reasoning capacities are improving, which in the limit should lead to the integration of formal methods[3]. Thus while my focus is on the AGI architectures that aim to explicitly foster cognitive synergy among diverse AI algorithms (such as Hyperon via the common knowledge language of MeTTa), as I argued at AITP'24 [11], LLM-based AI systems seem to have increasingly sophisticated architectures, so the integration of formal reasoning into them may become equally relevant.

Verifier Implementation: The initial approach was to directly copy the simple Python verifier for Metamath: *mmverify.py*[4]. Each function was re-implemented in MeTTa, checking for correctness, to reduce the risk of errors. The mmverify code parses a Metamath file sequentially, adding floating hypotheses ('type declarations'), essential hypotheses ('assumptions'), variables, and disjoint variable constraints to the appropriately scoped frame in a frames stack, which are used to construct assertions to add to the labels dictionary. The *verify* function is called on proof statements before they are added.

The MeTTa interpreter used[5] is still very slow[6], so the core text parsing (and preprocessing) is done in Python. Once a statement keyword (denoted by `$`) is parsed, the appropriate MeTTa

[1] Tutorial: `https://metta-lang.dev/`

[2] A large amount of mathematics has already been formalized within Metamath: `https://us.metamath.org/mpeuni/mmset.html`.

[3] However, in theory, a small LLM model may be able to replicate a proof kernel.

[4] `https://github.com/david-a-wheeler/mmverify.py`

[5] Hyperon Experimental 0.2.6

[6] Significant performance improvements are expected with the development of https://github.com/trueagi-io/MORK (MeTTa Optimal Reduction Kernel).

function is inserted from the Python (e.g., `$p` results in `add_p`). The implementation passes the metamath test suite[7] tests under 1000 lines long[8].

In MeTTa, *spaces* are used as the primary data structure. A space in MeTTa is a database of *atoms* (which can be of the types: *symbol atoms*, *expressions*, *grounded atoms*, and *variable atoms*) that can be queried via pattern matching (generally via the functions `(match $space $pattern $rewrite)` and `(unify $space $pattern $rewrite $fail)`). Two spaces are used: one for the `&stack` used to construct proof terms and one `&kb` space to store labels and frames. Thanks to pattern-matching, items in the frame can be stored by adding tags, "(FSDepth $depth)".

Example output and MeTTa programs for a few examples can be found on Github[9] (specifically, demo0.mm, disjoint2.mm, and 180 lines of hol.mm).

Avenues for future work: We would like to use Metamath data to do experiments with *inference control* and reasoning using generic *forward and backward chainers*[10] One motivation fro this project is Geisweiller's AITP'14 presentation, "Meta-Reasoning in MeTTa for Inference Control via Provably Pruning the Search Tree", which aims to ultimately have the AGI system verify its own cognitive algorithms and reasoning, which may be especially important if doing probabilistic proof search.

The verifier represents a *deep embedding*, and for reasoning, we'd like a *shallow embedding*. Geisweiller[11] and I[12] both found transformed demo0.mm into a format that can be checked with the backward chainer in MeTTa. Parentheses needed to be added to help guide the backward-chaining. Disjoint variable checking isn't implemented yet. Geisweiller's version looks increasingly like Metamath Zero [1, 2] (MM0), and MeTTa aims to deal with fresh variables "emphcorrectly", too, so it may be that importing math from MM0 is a wiser approach. Wernhard and Zombori's work with Metamath via CD Tools [19, 20] may also be helpful: they extract the compressed proof terms (as trees) into Prolog and analyze the proof structure, as well as looking for novel lemmas that can further compress the library.

A simple, sound kernel at the core of an AGI architecture should provide firm footing for other cognitive algorithms to verify (parts of) their solutions. When doing reasoning over uncertainties, such as with Probabilistic Logic Networks (PLN) [8], if the likelihoods and certainties appoarch 1, then it should be possible to extract a verifiable proof. There is the idea to use to do *conjecturing* or to guide proof-search by estimating the likelihood of statements to be theorems[13]. One ambitious goal is to integrate *meta-learning* about how to do the proof-search into the AI systems themselves, advancing the autonomous prover ideas seen in MaLARea [18], going beyond traditional AITP projects such as ENIGMA [9, 14]. We hope that integrating formal reasoning into an AGI framewrok will help to bridge the gap between "*higher-order*" reasoning with big steps for efficient proof search and *fast, low-level* verification, bridging the gap between Poincaré-style and Hilbert-style mathematics. Another potential application, in the example of the Alien Coding experiments [3, 4] over the OEIS [17] would be to learn and tweak the *language* that program generation is done in.

[7] https://github.com/zariuq/metamath-test

[8] The mmverify.py verifier had one small error and harmlessly didn't check for some properties of the specs, so a three tests were added.

[9] https://github.com/zariuq/mmverify.py/tree/master/examples

[10] See various experimental implementations here: https://github.com/trueagi-io/chaining/tree/main/experimental. The curried chainer may be interesting to investigate due to breaking down inference rules into minimal components.

[11] https://github.com/ngeiswei/chaining/blob/metamath-xt/experimental/metamath/demo0.metta

[12] https://github.com/zariuq/mmverify.py/blob/mettification/examples/demo0_bc.metta

[13] See Geisweiller's AITP'25 submission, "Estimating the Probability of a Conjecture to be a Theorem with PLN for Inference Control" [5]

Suggestions and pointers on how to integrate formal verification into AGI-aspiring cognitive inference control experiments will be much welcome. Especially as to potential pitfalls one could naively run into and ways to work around them.

Acknowledgements Thanks to Josef Urban for directing me to the mmverify.py implementation, and to Mario Cairneiro for a brief discussion on the nature of disjoint variables. Thanks to Gemini, ChatGPT, and Claude for much support. I appreciate the AITP community for welcoming work in progress submitted without comprehensive related literature review and the level of proof-reading that they deserve. This work is supported by Dar Novamente — Cognitive Inference Control.

References

[1] Mario Carneiro. Metamath zero: Designing a theorem prover prover. In *Intelligent Computer Mathematics: 13th International Conference, CICM 2020, Bertinoro, Italy, July 26–31, 2020, Proceedings*, page 71–88, Berlin, Heidelberg, 2020. Springer-Verlag.

[2] Mario Carneiro. *Metamath Zero: From Logic, to Proof Assistant, to Verified Compilation*. Ph.d. dissertation, Carnegie Mellon University, August 2022. Department of Philosophy, Pure and Applied Logic Program.

[3] Thibault Gauthier, Miroslav Olšák, and Josef Urban. Alien coding. *International Journal of Approximate Reasoning*, 162:109009, 2023.

[4] Thibault Gauthier and Josef Urban. Learning conjecturing from scratch. *arXiv e-prints*, pages arXiv–2503, 2025.

[5] Nil Geisweiller. Estimating the probability of a conjecture to be a theorem with pln for inference control. sep 2025.

[6] Ben Goertzel. Reflective metagraph rewriting as a foundation for an AGI ”language of thought”. *CoRR*, abs/2112.08272, 2021.

[7] Ben Goertzel, Vitaly Bogdanov, Michael Duncan, Deborah Duong, Zarathustra Goertzel, Jan Horlings, Matthew Ikle', Lucius Greg Meredith, Alexey Potapov, Andre' Luiz de Senna, Hedra Seid Andres Suarez, Adam Vandervorst, and Robert Werko. Opencog hyperon: A framework for agi at the human level and beyond, 2023.

[8] Ben Goertzel, Matthew Iklé, Izabela Freire Goertzel, and Ari Heljakka. *Probabilistic logic networks: A comprehensive framework for uncertain inference*. Springer Science & Business Media, 2008.

[9] Zarathustra A. Goertzel, Jan Jakubův, Cezary Kaliszyk, Miroslav Olšák, Jelle Piepenbrock, and Josef Urban. The Isabelle ENIGMA. In June Andronick and Leonardo de Moura, editors, *13th International Conference on Interactive Theorem Proving (ITP 2022)*, volume 237 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[10] Zarathustra Amadeus Goertzel. Formal ethics ontology in SUMO: Progress report and lessons learned. September 2023.

[11] Zarathustra Amadeus Goertzel. Atps as universal ais: What do agi architectures suggest for atp research? September 2024.

[12] Zarathustra Amadeus Goertzel. Formal meta-ethics ontology wiki. https://gardenofminds.art/esowiki/main/, 2025.

[13] Google DeepMind. Gemini deep think achieves gold-medal performance at the international mathematical olympiad. https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international jul 2025. Blog post.

[14] Jan Jakubův and Josef Urban. ENIGMA: efficient learning-based inference guiding machine. In Herman Geuvers, Matthew England, Osman Hasan, Florian Rabe, and Olaf Teschke, editors,

Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings, volume 10383 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 2017.

[15] Norman D. Megill and David A. Wheeler. *Metamath: A Computer Language for Mathematical Proofs.* Lulu Press, Morrisville, North Carolina, 2019. `http://us.metamath.org/downloads/metamath.pdf`.

[16] Lucius Gregory Meredith, Ben Goertzel, Jonathan Warrell, and Adam Vandervorst. Meta-metta: an operational semantics for metta, 2023.

[17] Neil J. A. Sloane. The on-line encyclopedia of integer sequences. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants*, pages 130–130, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[18] Josef Urban, Geoff Sutcliffe, Petr Pudlák, and Jičí Vyskočil. MaLARea SG1 - Machine Learner for Automated Reasoning with Semantic Guidance. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Proc. of the 4th IJCAR, Sydney*, volume 5195 of *LNAI*, pages 441–456. Springer, 2008.

[19] Christoph Wernhard and Zsolt Zombori. Exploring metamath proof structures. September 2024.

[20] Christoph Wernhard and Zsolt Zombori. Mathematical knowledge bases as grammar-compressed proof terms: Exploring metamath proof structures, 2025.