

Geachte meneer Barendrecht,

my name is Maxim, I write to you because I love lambda calculus and meditations. I represent a small group of people from Ukraine who do a formal verification and build the new theorem prover. Also, I belong to Nyingma lineage of Tibetan Buddhism and we also have Mahamudra (Shamatha+Vipassana) empowering as Kagyu lineage do.

Let me explain the reason why I'm writing this letter. We host a software house building full stack applications in Erlang language. And we reached the state when we need formal verification for our software. We are working on our own prover core based on pure CoC enriched with an infinite number of predicative universes<sup>1</sup>. However, the target is to support even higher inductive types and HoTT. But we came up with conclusion that the core of our prover shouldn't contain Equality, Fixpoint, (Co)-Inductive definitions, Recursor/Induction, thus be as small as possible. We even found a Ph.D. paper from University of Nijmegen that somehow denies this requirement<sup>2</sup>. The other feature of our prover is to have replaceable and switchable encodings (Church, Scott, Parigot, CPS, etc). The categorical model of encoding engine is built using Lean prover<sup>3</sup>. The top-level language called EXE<sup>4</sup> would be in CiC semantics and translated to CoC using our categorical encoding model. Now we have built the Induction term for Bool using our pure CoC language<sup>5</sup>. Also, we do have stable prelude<sup>6</sup> in toy-Church encoding for Hello World demo with IO and IOI (co)-monads, that already compiles (extracts) to Erlang bytecode. The overall size of our prover core is 600 lines of code in Erlang<sup>7</sup>.

In the beginning, I wanted to write you as I (and our team) dream to do a Ph.D. with your supervision. But I realize that my knowledge and your time you could spend on this project is limited. Also, we have language limitations, country and bureaucracy barriers.

We have an idea to reincarnate the AUTOMATH prover in its original syntax based on our compact OM core (AUT-OM flavor). Such we would have contemporary EXE language (with polynomial functors as in Coq/Agda) and AUTOMATH flavor both based on OM core. Of course, we would be glad and ready to any depth level of cooperations from your side, as we treat you as heart son of de Bruijn, the father of all provers. We don't want to search mentors among French (Coq), British (McBride, Altenkirch, Brady) and other schools as we don't share their ideas, however, we use their findings and respect them a much.

Please take us under your supervision and give us your official blessings, at least for making official AUTOMATH front-end to our prover.

With regards, our team: Maxim Sokhatsky, Paul Lyutko and Andrew Melkinov.

---

<sup>1</sup><http://groupoid.space>

<sup>2</sup>Herman Geuvers. Induction Is Not Derivable in Second Order Dependent Type Theory

<sup>3</sup><https://github.com/groupoid/exe/tree/master/prelude/lean>

<sup>4</sup><https://github.com/groupoid/exe>

<sup>5</sup><https://github.com/groupoid/om/tree/master/priv/posets>

<sup>6</sup><https://github.com/groupoid/om/tree/master/priv/normal>

<sup>7</sup><https://github.com/groupoid/om>