Application for a Computing Time Project on the RWTH Compute Cluster

Project extension proposal for project "pmGenerator"

Period: 01.04.2026 – 31.03.2027

An exhaustive generator to find shortest known condensed detachment proofs, focussing on minimalist Hilbert systems for propositional and modal logic

Samiro Discher

March 20, 2026

Principal investigator: Thomas Noll*
Technical contact & project contributor: Samiro Discher*

* Lehrstuhl für Informatik 2 (Softwaremodellierung und Verifikation)

Abstract

Utilization of a proof generator with shared memory parallelization making heavy use of Intel's oneTBB library, and distributed memory parallelization via MPI for a computing-intensive filtering method.

As described in the original proposal[1], the tool *pmGenerator* can generate exhaustive proof collections in concise formal representations. Since version 1.2, released on March 3, 2024, it allows user-defined axioms to customize systems based on rules `D` for condensed detachment and `N` for necessitation. The latter can be used to define systems of modal logic and is disabled by default. Plenty of features to assist with generating shorter proofs from longer known proofs were added up to the current version towards the 1.2.3 release. This includes fully automated proof compression algorithms that led to significant progress in proof minimization – as presented under "Achieved Results" and in the previous extension proposal.[2] Furthermore, the tool can now be used in a more targeted way, thanks to new features that allow conversion from (user-friendly) Fitch-style natural deduction, via `--ndconvert`, and from other Hilbert systems, via `--rebase`.

Complex proof systems can best be explored using extensive automation. This applies in particular to the seven minimal 1-bases for propositional logic in terms of $\{\rightarrow, \neg\}$ under modus ponens, which is encompassed by condensed detachment. As mentioned in previous proposals, these minimal 1-bases are Meredith's single axiom `CCCCCpqCNrNsrtCCtpCsp` and Walsh's six axioms `CCpCCNpqrCsCCNtCrtCpt`, `CpCCqCprCCNrCCNstqCsr`, `CpCCNqCCNrsCptCCtqCrq`, `CpCCNqCCNrsCtqCCrtCrq`, `CCpqCCCrCstCqCNsNpCps` and `CCCpqCCCNrNsrtCCtpCsp`. Based on formerly obtained data, completeness results for all of these systems can be easily reproduced with *pmGenerator*. Moreover, I recently published the first derivations of these axioms from one another, as outlined under "Achieved Results". So far, I did not attempt to reproduce the claim that those seven axioms are the *only* ones of their kind, but this is a challenge soon to be tackled. Along with delving deeper into minimalist proof systems and seeking out additional ways to make effective use of the tool.

Keywords: Logic, Proof theory, Hilbert systems, Condensed detachment

[1] `https://xamidi.github.io/pmGenerator/pdf/rwth1392_abstract.pdf`
[2] `https://xamidi.github.io/pmGenerator/pdf/rwth1392_extension_2025.pdf`

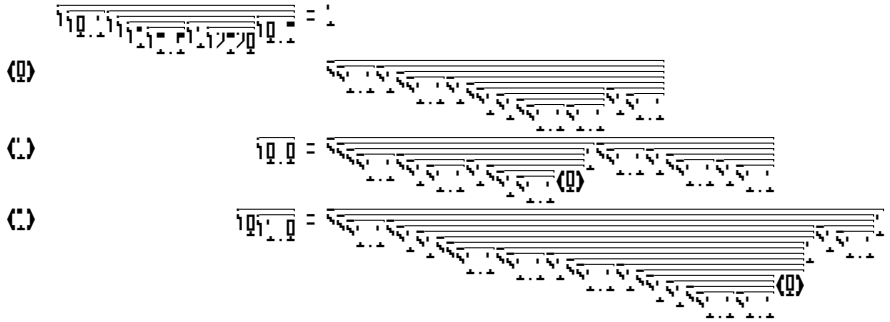
Achieved Results

Apart from supporting the development and testing of the free and open-source software project *pmGenerator*[3], this computing time project generated a lot of knowledge in the past year, including but not limited to:

- Nine proofs reduced by 4014 steps in the "Minimal 1-bases for C-N propositional calculus"[4] proof minimization challenge.
  - This reduced the total number of steps by approximately 17.79%, from 22561 to 18547, further narrowing the gap to its greatest known lower bound, which is currently at 3913.
- More efficient proof compression algorithms, where "`--transform -z`" and regular "`--transform -x`" rounds (and their nested loops) have essentially been replaced by a sequence of "`-x 3`" preparations until no further improvements occur. This removed the option of deterministic proof compression via single-threaded D-rule replacement search for release versions 1.2.2 and later. But the previous approach of getting done less in regular rounds has been kept as a "baby steps" option.
- An algorithm to convert proofs from Fitch-style natural deduction to condensed detachment in a user-defined Hilbert system, given an appropriate knowledge base for the target system, or, without extra information, to the default system. Implemented as `--ndconvert` feature.
- The `--ndconvert` feature was applied on handcrafted natural deduction proofs of the seven minimal classical C-N single axioms in order to create condensed detachment proofs from one another. Compressing these correlative proofs using substantial computing resources led to rather small derivations; the forty-two proofs of interest have a total of 102278 primitive steps. This started a new proof minimization challenge "Traversing minimal classical C-N single axioms"[5].
  - None of these correlative proofs are known to be minimal, in contrast to sixteen proofs of target theorems for the preceding challenge. The greatest known lower bound for their total number of steps is currently at 4158.

[3] `https://github.com/xamidi/pmGenerator`
[4] `https://github.com/xamidi/pmGenerator/discussions/2`
[5] `https://github.com/xamidi/pmGenerator/discussions/10`

Furthermore, I created a proof printer[6] that, apart from converting formula notation and presenting graphical syntax trees, can display abstract D-proofs over C-N formulas in a compact yet structured way. For example,



illustrates proofs of $\psi \to \psi$ and $\psi \to (\varphi \to \psi)$ from Walsh's fifth axiom. They can be written in plain text as

```
    CCpqCCCrCstCqCNsNpCps = 1
[0] DD11D1DD11D1DD1D1DD11D11D1D11
[1] Cpp = DDDD11DD1D11D1DD1D11[0]1DD11D1DD11D1D11
[2] CpCqp = DDD11DD1D1DDD11DD11D1DD11D1DD1D1DD11D11[0]1D1D111
```

and be restored via "`pmGenerator --transform [...] -w -n -t . -j -1`" after removing explicit conclusions also for "`[1]`" and "`[2]`".

Last but not least, I used *pmGenerator* to create a variant[7] of Metamath's "Shortest known proofs of the propositional calculus theorems from Principia Mathematica"[8] collection, but based on axioms `CCpqCCqrCpr,CCNppp,CpCNpq`, in contrast to `CpCqp,CCpCqrCCpqCpr,CCNpNqCqp`. The used system is also known as "Łukasiewicz (L_1)-system" and the latter as "Łukasiewicz (L_3)-system". These names were given in "On axiom systems of propositional calculi", a series of twenty-five papers that was published by the Japan Academy from 1965 to 1967.[9]

[6] `https://xamidi.github.io/logic-structuralizer/`
[7] `https://github.com/xamidi/luk-pmproofs`
[8] `https://us.metamath.org/mmsolitaire/pmproofs.txt`
[9] `https://github.com/xamidi/pmGenerator/discussions/12`