

# LLMs as Proof Reconstructors for ATP Hammers?

## (Project Proposal)

Josef Urban

Czech Technical University in Prague

### Motivation and Plans

This work is motivated by Cyril Cohen’s talk<sup>1</sup> at Chalmers about recovering (typically short and high-level) Coq proof scripts from (typically longer and low-level) Coq proof objects. His team have finetuned DeepSeek on such tasks. Then they used that to transfer a proof script  $S_1$  from one Coq library ( $L_1$ ) to another Coq library ( $L_2$ ) via first compiling the  $S_1$  script (in  $L_1$ ) to a proof object  $P_1$ . After the compilation, they treat  $P_1$  as a proof object  $P_2$  in  $L_2$  and decompile it with the finetuned DeepSeek into a proof script  $S_2$  in  $L_2$ . Even though this is done within one proof assistant, the libraries and styles may be quite different. Yet, Cyril reported encouraging results.

It occurred to me that if such “decompilation” works (at least to some extent), there is a similar task of “decompiling” hammer (ATP) proofs into nice high-level and human readable scripts in declarative systems such as Mizar, Isabelle and Naproche. While there is prior “human intelligence” work on this task (e.g. [1]), it has always been a challenge that may be perhaps AI-complete when taken to the extreme. Consider e.g. the task of “humanly” presenting the long ATP proofs obtained often in equational algebra by Veroff [2] and others [3].

Our planned project is to therefore finetune DeepSeek (or other LM) on the large corpus of TPTP proofs created by E/ENIGMA or Vampire/Deepire and their Mizar (or Isabelle or Naproche) declarative counterparts. Initially, I have created a dataset of 10000 such pairs for training and evaluation. I hope to have the larger finetuning experiment done on them by the time of the conference.

### First Experiments

I have first tried to test the capability of the online Deepseek-R1 on several smaller examples. I have started with the following simple prompt: “Turn this TPTP proof into a nice short high-level declarative Mizar proof.” DeepSeek could (surprisingly) figure out the original Mizar notation of `t100_xcmlpx.1`<sup>2</sup> – a small proof that  $a/(1/b) = a * b$  for complex numbers.

```
theorem
  for a, b being complex number
    holds a / (1 / b) = a * b
proof
  let a, b be complex number ;
  thus a / (1 / b) = a / (b ") by Lm4
  . = a * (1 / (b ")) by Lm14
  . = a * b by Lm16 ; :
end;
```

(a) Original

```
theorem
  for X1, X2 being complex number st X2 <> 0
    holds X1 / (1 / X2) = X1 * X2
proof
  let X1, X2 be complex number;
  assume X2 <> 0;
  thus X1 / (1 / X2) = X1 * (1 / X2)"
  by XCMLPX_1: def 9
  . = X1 * X2 by XCMLPX_1:54;
end;
```

(b) DeepSeek

<sup>1</sup><https://www.cse.chalmers.se/~sattler/workshop-deep-learning-math.html#cyril>

<sup>2</sup>[http://grid01.ciirc.cvut.cz/~mptp/7.13.01\\_4.181.1147/html/xcmlpx\\_1.html#T100](http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/xcmlpx_1.html#T100)

While the references differ, the DeepSeek proof type checks and the gaps can be easily ATP-hammered. The ATP proof has 5kb and 32 lines.<sup>3</sup> After this initial encouragement, DeepSeek however struggled with recovering the original Mizar notation for the ATP proof<sup>4</sup> of `t52_topgen_1`<sup>5</sup> ( `IRRAT` is dense Subset of  $R^1$  ), outputting a proof with unwanted TPTP symbols. (The detailed outputs for this and the remaining examples are in the Appendix.)

In the second iteration I therefore started to use a script that tries to recover the original Mizar notation for the MPTP translation of Mizar symbols. This is nontrivial, because of the Mizar mechanisms such as synonyms, antonyms, expandable modes, etc. The back translation is necessarily crude and may require further work. An example where this imperfect translation changed the semantics is `t105_card_3`<sup>6</sup>, where the theorem is about finiteness, but because finite is an antonym to infinite, the recovered theorem is (wrongly, though still provably) about infiniteness.

In the third iteration, I have started to hide the information about the original conjecture. It seems that at least in one case DeepSeek could just look up on the internet (or remembered) the corresponding Mizar theorem and its proof and printed it almost verbatim. I have changed the name of the article from `card_3` to `myart`, after which DeepSeek still produced reasonable declarative proof, however without such direct copying.

I had more success with DeepSeek than with Chatgpt, perhaps also due to imperfect prompting. With the latest prompt, Chatgpt seems to be more successful and perhaps also competitive to DeepSeek. I made several attempts with Claude 3.7 Sonnet but none of them were comparably good to DeepSeek.

The (current) final prompt is:

*“Turn this TPTP proof into a nice short high-level declarative Mizar proof. Note that the Mizar names of the symbols are always given here after the tptp name, e.g. as in `”v1_tops_1::dense”` . So only use the Mizar names, e.g. `”dense”` here. The results has to be formally correct Mizar code.”*

## References

- [1] Jasmin Christian Blanchette, Sascha Böhme, Mathias Fleury, Steffen Juilf Smolka, and Albert Steckermeier. Semi-intelligible isar proofs from machine-generated proofs. *J. Autom. Reason.*, 56(2):155–200, 2016.
- [2] Michael K. Kinyon, Robert Veroff, and Petr Vojtechovský. Loops with abelian inner mapping groups: An application of automated deduction. In Maria Paola Bonacina and Mark E. Stickel, editors, *Automated Reasoning and Mathematics - Essays in Memory of William W. McCune*, volume 7788 of *LNCS*, pages 151–164. Springer, 2013.
- [3] Jiří Vyskočil, David Stanovský, and Josef Urban. Automated Proof Compression by Invention of New Definitions. In Edmund M. Clarke and Andrei Voronkov, editors, *LPAR (Dakar)*, volume 6355 of *LNCS*, pages 447–462. Springer, 2010.

<sup>3</sup>[http://grid01.ciirc.cvut.cz/~mptp/hs1/t100\\_xcmplx\\_1](http://grid01.ciirc.cvut.cz/~mptp/hs1/t100_xcmplx_1)

<sup>4</sup>[http://grid01.ciirc.cvut.cz/~mptp/hs1/t52\\_topgen\\_1](http://grid01.ciirc.cvut.cz/~mptp/hs1/t52_topgen_1)

<sup>5</sup>[http://grid01.ciirc.cvut.cz/~mptp/7.13.01\\_4.181.1147/html/topgen\\_1.html#T52](http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/topgen_1.html#T52)

<sup>6</sup>[http://grid01.ciirc.cvut.cz/~mptp/7.13.01\\_4.181.1147/html/card\\_3.html#T105](http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/card_3.html#T105)

## A More examples

### A.1 t52\_topgen\_1

original Mizar:

```
reserve T for non empty TopSpace,
A, B for Subset of T,
F, G for Subset-Family of T,
x for set;

theorem Th52:
IRRAT is dense Subset of  $\mathbb{R}^1$ 
proof
Cl A = the carrier of  $\mathbb{R}^1$  by BORSUK_5:28;
hence thesis by TOPS_3:def 2;
end;
```

best deepseek:

```
theorem
v1_tops_1::dense(IRRAT,  $\mathbb{R}^1$ ) & m1_subset_1::Element(IRRAT, k1_zfmisc_1::bool( $\mathbb{R}$ ))
proof
IRRAT is Subset of  $\mathbb{R}$  by BORSUK_5:def 4; -- IRRAT is defined as a subset of  $\mathbb{R}$ 
hence m1_subset_1::Element(IRRAT, k1_zfmisc_1::bool( $\mathbb{R}$ )) by SUBSET_1:def 1;
set T =  $\mathbb{R}^1$ ; -- Real numbers with standard topology
A1: the carrier of T =  $\mathbb{R}$  by TOPMETR:17;
Cl IRRAT = the carrier of T by BORSUK_5:57; -- Closure of IRRAT in  $\mathbb{R}^1$  is entire space
hence v1_tops_1::dense(IRRAT, T) by A1, TOPS_3:def 2;
end;
```

### A.2 t101\_finseq\_3

original Mizar:

```
reserve p,q,r for FinSequence;
reserve u,v,x,y,y1,y2,z,A,D,X,Y for set;
reserve i,j,k,l,m,n for Nat;

theorem
p is one-to-one & rng p = {x,y,z} & x <> y & y <> z & x <> z implies
len p = 3
proof
assume that
A1: p is one-to-one and
A2: rng p = {x,y,z} and
A3: x <> y and
A4: y <> z and
A5: x <> z;
<* x,y,z *> is one-to-one by A3,A4,A5,Th95;
hence thesis by A1,A2,Th100;
end;
```

Best deepseek:

```
environ

vocabularies RELAT_1, FUNCT_1, FINSEQ_1, CARD_1, ARYTM_3, ORDINAL1, XBOOLE_0,
SUBSET_1, TARSKI, NUMBERS, XXREAL_0, NAT_1, FUNCT_7, FINSEQ_2, CARD_2,
FINSEQ_3;
notations TARSKI, XBOOLE_0, SUBSET_1, ORDINAL1, NUMBERS, XCMPLX_0, XXREAL_0,
```

```

NAT_1, RELAT_1, FUNCT_1, FINSEQ_1, FINSEQ_2, FINSEQ_3, CARD_1, CARD_2,
VALUED_0;
constructors XXREAL_0, NAT_1, FINSEQ_3, CARD_2, FINSEQ_2, RELSET_1;
registrations XREAL_0, NAT_1, FINSEQ_1, CARD_1, ORDINAL1, RELSET_1;
requirements REAL, NUMERALS, SUBSET, BOOLE, ARITHM;
theorems FINSEQ_3, CARD_2, CARD_1, FINSEQ_1, TARSKI, FUNCT_1, XBOOLE_0,
XREAL_1, XXREAL_0, NAT_1;
schemes NAT_1;

begin

reserve X for FinSequence,
a,b,c for set;

theorem
for X being FinSequence
st X is one-to-one & rng X = {a,b,c} & a <> b & b <> c & a <> c
holds len X = 3
proof
let X be FinSequence;
assume that
A1: X is one-to-one and
A2: rng X = {a,b,c} and
A3: a <> b & b <> c & a <> c;
A4: card rng X = 3 by A3,CARD_2:58;
thus len X = card dom X by FINSEQ_3:def 1
.= card rng X by A1,CARD_1:70
.= 3 by A4;
end;

end;

```

### A.3 t101\_pboole:

original Mizar:

```

reserve i,j,e,u for set;
reserve I for set;
reserve x,X,Y,Z,V for ManySortedSet of I;

theorem Th101:
X c= Y & Z c= V & X overlaps Z implies Y overlaps V
proof
assume that
A1: X c= Y and
A2: Z c= V;
assume X overlaps Z;
then Y overlaps Z by A1,Th100;
hence thesis by A2,Th99;
end;

```

deepseek - this one is funny because it got the symbols wrong:

```

environ

vocabularies RELAT_1, FUNCT_1, PBOOLE, XBOOLE_0;
notations TARSKI, XBOOLE_0, RELAT_1, FUNCT_1, PARTFUN1, PBOOLE;
constructors TARSKI, XBOOLE_0, RELAT_1, FUNCT_1, PARTFUN1, PBOOLE;
registrations XBOOLE_0, RELAT_1, FUNCT_1, PARTFUN1, PBOOLE;
requirements SUBSET, BOOLE;
theorems PBOOLE, RELAT_1, FUNCT_1, PARTFUN1;

```

```

schemes FUNCT_1;

begin

reserve X for set;
reserve f, g, h, k for Relation;

theorem Th101:
for X being set, f, g, h, k being Relation
st f is_partial_function_on X & g is_partial_function_on X &
h is_partial_function_on X & k is_partial_function_on X &
f tolerates g & h tolerates k & f subfunction_of h
holds g subfunction_of k
proof
let X be set, f, g, h, k be Relation such that
A1: f is_partial_function_on X and
A2: g is_partial_function_on X and
A3: h is_partial_function_on X and
A4: k is_partial_function_on X and
A5: f tolerates g and
A6: h tolerates k and
A7: f subfunction_of h;
A8: f c= h by A7,PBOOLE:def 18;
g c= k by A1,A2,A3,A4,A5,A6,A8,PBOOLE:107;
hence thesis by PBOOLE:def 18;
end;

end;

```

Another version which looks like cheating by lookup:

```

environ

vocabularies RELAT_1, PBOOLE;
notations TARSKI, PBOOLE;
constructors PBOOLE;
registrations PBOOLE;
theorems TARSKI, PBOOLE;
schemes MSUALG_1;

begin

theorem Th101:
for I being set,
A, B, C, D being ManySortedSet of I
st A c= B & C c= D & overlaps I,A,C
holds overlaps I,B,D
proof
let I be set,
A, B, C, D be ManySortedSet of I;
assume that
A1: A c= B and
A2: C c= D and
A3: overlaps I,A,C;
overlaps I,B,C by A1,A3,PBOOLE:143; -- Applying t100_pboole
then overlaps I,C,B by PBOOLE:142; -- Applying symmetry_r4_pboole
then overlaps I,D,B by A2,PBOOLE:143;-- Applying t100_pboole again
hence overlaps I,B,D by PBOOLE:142; -- Applying symmetry_r4_pboole
end;

:: The original theorem t101_pboole is now directly proven by Th101
theorem

```

```

for I being set,
A, B, C, D being ManySortedSet of I
st A c= B & C c= D & overlaps I,A,C
holds overlaps I,B,D by Th101;

```

```
end.
```

But after renaming pboole to myart the cheating disappears and the output is plausible:

```

theorem Th101:
for X1 being set,
X2, X3, X4, X5 being Relation-like (X1)-defined Function-like total set
st X2 c= X3 & X4 c= X5 & overlaps(X1,X2,X4)
holds overlaps(X1,X3,X5)
proof
let X1 be set,
X2, X3, X4, X5 be Relation-like (X1)-defined Function-like total set;
assume that
A1: X2 c= X3 and
A2: X4 c= X5 and
A3: overlaps(X1,X2,X4);
A4: overlaps(X1,X4,X2) by A3, SYM_4;
overlaps(X1,X5,X2) by A2, A4, Th100;
then overlaps(X1,X2,X5) by SYM_4;
hence overlaps(X1,X3,X5) by A1, Th100;
end;

```

## A.4 t105\_card\_3

A longer one - the ATP proof is 12k and 52 lines.<sup>7</sup> original Mizar:

```

reserve A,B,C for Ordinal,
K,L,M,N for Cardinal,
x,y,y1,y2,z,u,X,Y,Z,Z1,Z2 for set,
n for Nat,
f,f1,g,h for Function,
Q,R for Relation;
reserve ff for Cardinal-Function;
reserve F,G for Cardinal-Function;
reserve A,B for set;
reserve A,B for Ordinal;
reserve n,k for Nat;

theorem
for R,S being Relation st R,S are_isomorphic & R is finite
holds S is finite
proof
let R,S be Relation;
given F being Function such that
A1: F is_isomorphism_of R,S;
assume R is finite;
then field R is finite;
then dom F is finite by A1,WELLORD1:def 7;
then rng F is finite by FINSET_1:8;
then field S is finite by A1,WELLORD1:def 7;
hence thesis;
end;

```

deepseek - turning “finite” into “infinite”

<sup>7</sup>[http://grid01.ciirc.cvut.cz/~mptp/hs1/t105\\_card\\_3](http://grid01.ciirc.cvut.cz/~mptp/hs1/t105_card_3)

```

theorem Th105Myart:
for R1, R2 being Relation
st are_isomorphic(R1,R2) & infinite(R1)
holds infinite(R2)
proof
let R1, R2 be Relation;
assume that
A1: are_isomorphic(R1,R2) and
A2: infinite(R1);
consider f being Function such that
A3: f is_isomorphism_of R1,R2 by A1, D8_WELLORD1:def 4;
A4: dom f = field(R1) & rng f = field(R2) & f is one-to-one
by A3, D7_WELLORD1:def 7;
infinite(field(R1)) by A2, FC28_FINSET_1:18;
then infinite(field(R2)) by A4, T8_FINSET_1:19;
hence infinite(R2) by FC14_MYART:20;
end;

```

chatgpt 04-mini-high:

```

theorem
for R, S being Relation-like holds
(are_isomorphic(R, S) & infinite R) implies infinite S
proof
let R, S be Relation-like;
assume that
A1: are_isomorphic(R, S) and
A2: infinite R;

(* from R infinite and relation-like R *)
infinite field R by A2, fc28_finset_1;

(* obtain the isomorphism f : R S *)
choose f such that
A3: Relation-like f & Function-like f & is_isomorphism_of(R, S, f)
by A1, d8_wellord1;

(* in particular f is a bijection with proj1(f)=field R and proj4_4(f)=field S *)
proj1 f = field R & proj4_4 f = field S & one-to-one f
by A3, d7_wellord1;

(* hence field R infinite proj1(f) infinite *)
then infinite proj1 f by A2, fc28_finset_1;

(* apply t8: a one-to-one function with infinite domain has infinite range *)
then infinite proj4_4 f by A3, t8_finset_1;

(* so field S is infinite *)
then infinite field S by A3, d7_wellord1;

(* and conversely infinite field S infinite S *)
hence infinite S by fc14_myart;
end;

```