

# Benchmarks for Higher-Order Automated Reasoning

Chad E. Brown

`cebrown@ags.uni-sb.de`

Universität des Saarlandes,

Saarbrücken, Germany

# Grundlagen: The History

## Case Study:

- *Grundlagen der Analysis*: Edmund Landau 1920's
- Formalized by L. S. van Benthem Jutting in Automath 1970's
- Recovered by Freek Wiedijk 1990's

# Grundlagen: The Book

Mathematical Constructions in Landau's *Grundlagen der Analysis*:

- (Positive) Natural Numbers:  $1, Suc$
- Fractions (Pairs of Natural Numbers)
- (Positive) rationals (equivalence classes of pairs)
- Cuts (some sets of fractions)
- Real Numbers (Cut,  $-Cut$ , or  $0$ )
- Complex Numbers (pairs of Real's)
- Sums and Products of  $n$ -tuples of Complex Numbers

# What do we need?

- (Positive) Natural Numbers:  $1, Suc$
- Peano Axioms
- Pairs (of Natural Numbers, of Reals)
- Quotients by Equivalence Classes
- Sets (of rationals)
- $n$ -tuples for natural numbers  $n$ .

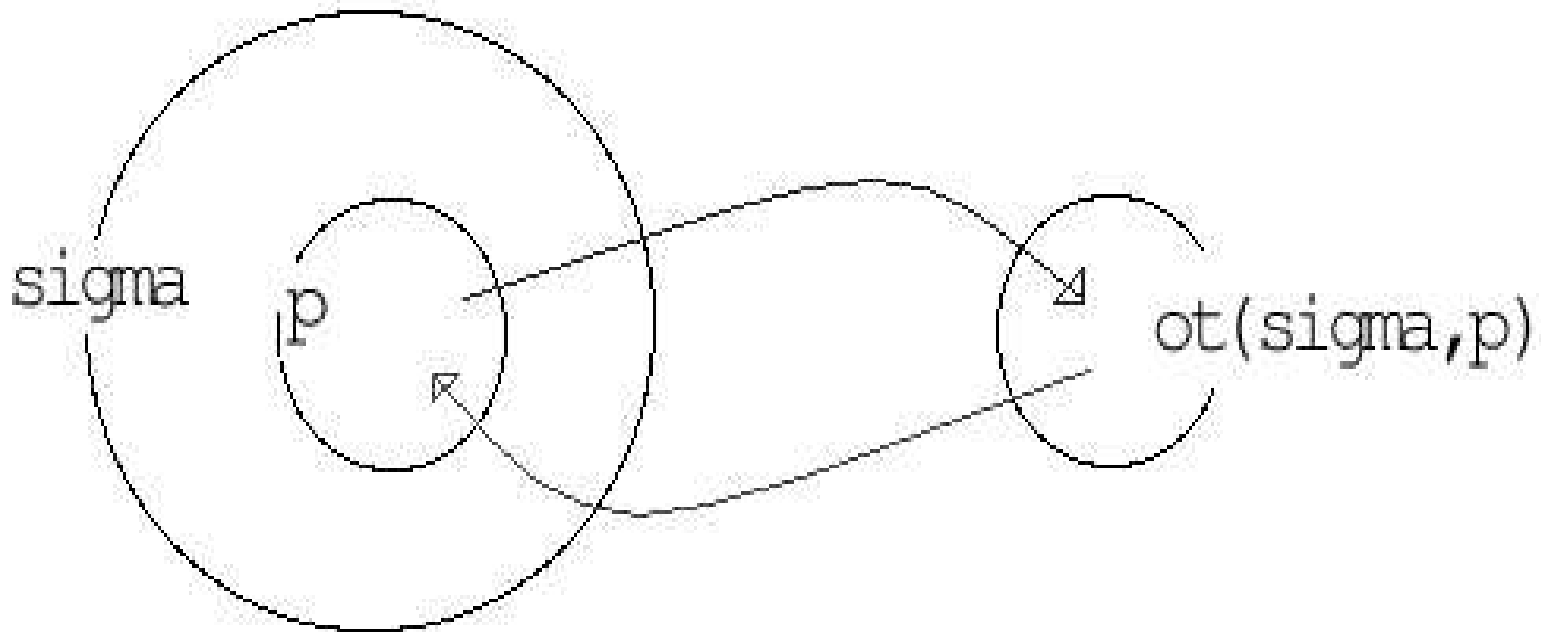
# Grundlagen: The Automath

Automath – Aut-QE “Quasi-Expression”

- Aut-QE provides:  $\lambda$ , application, PROP and TYPE
- Dependent Types (with Simple Types as a special case)
- Build Classical Logic on Top of Aut-QE
- Assume “Dependent Predicate Types”

# Dependent Predicate Types

- $\text{ot}(\text{sigma}, p)$ : Given any type  $\sigma$  and predicate  $p : \sigma \rightarrow \text{PROP}$ ,  $\text{ot}(\text{sigma}, p)$  is a type isomorphic to the collection of  $\sigma$ -elements satisfying  $p$ .



- Example  $1 \text{ to } n$  corresponds to  $\{1, \dots, n\}$

# Porting From Automath

- Extensive use of Predicate Types
- Can these be eliminated (along with other Aut-QE features)  
to obtain a simply typed version of the encoding?
- Yes!
- Key Idea: Interpret Automath Types as PER's (partial equivalence relations) over Simple Types.
- Once Types are interpreted properly, Terms are interpreted (mostly) by erasing Dependent Types.
- The encoding has been ported (**WITHOUT PROOFS**) to TPS and  $\Omega$ mega.

# Simply Typed Encoding

- After porting to simply-typed higher-order logic, there are 3 axioms and one axiom schema:
  - Three Peano Axioms at base type  $\iota$ .
  - Description at all types.
- There are 5998 open theorems which should follow from the axioms.

# How do the encodings compare?

- `satz289b`: If  $f$  is an  $x$ -tuple of complex numbers,  $n$  is between 1 and  $x$ , and  $f_n = 0$ , then the product  $f_1 \cdot \dots \cdot f_x$  of the  $x$ -tuple is 0.

Jutting proves this using the previously proven:

- `satz289`: Any  $x$ -tuple of complex numbers has product 0 iff there exists some  $n$  between 1 and  $x$  such that  $f_n = 0$ .

# Automath Version

- Theorem:

```
[x:nat][f:[t:1to(x)]cx][n:1to(x)]  
[i:is(<n>f,0c)]  
...is(prod(x,f),0c)
```

- Proof:

```
[x:nat][f:[t:1to(x)]cx][n:1to(x)]  
[i:is(<n>f,0c)]  
satz289b:=th4"1.iff"(prop1".8289",  
prop2".8289",satz289,t41".8289")  
:is(prod(x,f),0c)
```

# Simply Typed Version

\_SATZ289B:

$$\forall x_i \forall f_{o(o(o(u))\iota)u} \cdot \forall i_\iota \forall j_\iota [ \_1\text{TO } o_{uu} x_i j \supset$$

$$\_C\text{X } o(o(o(o(u))\iota)\iota)(o(o(o(u))\iota)\iota) [ f_i ] \cdot f_j ]$$

$$\supset \forall n_\iota \_1\text{TO } x_n n \supset .$$

$$\_C\text{IS } o(o(o(o(u))\iota)\iota)(o(o(o(u))\iota)\iota) [ f_n ] \_0\text{C } o(o(o(u))\iota)\iota$$

$$\supset \_C\text{IS } [ \_C\text{PROD } o(o(o(u))\iota)\iota(o(o(o(u))\iota)u)\iota x f ] \_0\text{C}$$

# Simply Typed Version

`_SATZ289B` Omitting Types:

$$\forall x \forall f. \forall i \forall j [\_1\text{TO } x \ i \ j \supset \_C\text{X } [f \ i]. \ f \ j]$$
$$\supset \forall n. \_1\text{TO } x \ n \ n \supset \_C\_IS \ [f \ n]\_0\text{C}$$
$$\supset \_C\_IS \ [\_C\_PROD \ x \ f]\_0\text{C}$$

# Three Potential Versions

- `_SATZ289B` is an abbreviation for the formula:

$$\forall x \forall f. \forall i \forall j [ \_1TO \ x \ i \ j \supset \_CX \ [ \ f \ i \ ]. \ f \ j ]$$

$$\supset \forall n_{\iota}. \_1TO \ x \ n \ n \supset \_C\_IS \ [ \ f \ n \ ] \_0C$$

$$\supset \_C\_IS \ [ \_C\_PROD \ x \ f \ ] \_0C$$

- Version 1:

`_SATZ289B_THM` is the “theorem”

`_SATZ289B`

- A proof of `_SATZ289B_THM` may (should) require the Peano Axioms at type  $\iota$  and the description axiom at all (enough) types.

# Three Potential Versions

- Version 2: `_SATZ289B_ATHM` could be the “theorem with axioms as hypotheses”

$$(\forall \alpha : \textit{Types.Description}^\alpha) \wedge \textit{Peano} \supset \textit{\_SATZ289B}$$

but this is not a legal formula (quantification over types is not allowed).

- In general, we form an `ATHM` if the proof does not depend on the polymorphic axiom of descriptions.

# Three Potential Versions

- Version 3: `_SATZ289B_HTHM` is the “hypothetical theorem”

$$\_8289\_T41 \wedge \_SATZ289 \wedge \_IFF\_TH4 \supset \_SATZ289B$$

where `_8289_T41`, `_SATZ289`, `_IFF_TH4` are previously proven theorems referenced in the Automath proof of `_SATZ289B`.

- In general, form an `HTHM` if none of the previously proven theorems/axioms referenced in the Automath proof are polymorphic.
- `_SATZ289B_HTHM` has a *reasonably sized* proof. Simply apply logical manipulations to `_SATZ289` in order to obtain `_SATZ289B`.

# Reasonable Benchmark Examples

Start with 5998 Grundlagen theorems.

- 385 reasonable THM's, i.e., theorems provable without using axioms.
- 39 reasonable ATHM's, i.e., theorems provable from some Peano Axioms without using description.
- 3766 reasonable HTHM's, i.e., theorems provable from previous non-polymorphic results
- 4190 reasonable theorems.

# The Idea

- A user  $J$  is trying to formalize Landau.
- $J$  formalizes a theorem  $C$ .
- $J$  indicates which previous theorems (or axioms)  $A^1, \dots, A^n$  can be used to prove  $C$ .
- $J$  asks a HOATP to prove the HTHM

$$A^1 \wedge \dots \wedge A^n \supset C$$

# A First TPS Run

- Ran TPS on all 5998 THM's, 39 ATHM's, and 3766 HTHM's
- 9803 potential theorems (most unreasonable)
- TPS with 12 modes (series of flag settings) for five seconds each.
- Results...Succeeded on 468 of 9803 (4.7%)
- 8 of the proven theorems were classified as “unreasonable”
  - (Automath used the induction axiom to define plus, but the induction axiom was not needed in the proofs of these 8 theorems.)
- Among reasonable, 460 of 4190 (11% in  $\leq 5$  seconds)

# A Second TPS Run

- Selected 42 problems randomly from the 4190 reasonable problems (1%).
- Ran TPS with the same 12 modes with time limit of 5 minutes
- TPS succeeded on 10 of 42 – (23.8% success!)
- Maximal time was 100 seconds.
- TPS succeeded on 7 of 42 within 7 seconds – (16.7% success)

# Chapters of Benchmark Examples

Dividing the 4190 reasonable theorems into “Chapters”:

- Chapter 0: 525 - Automath Preliminaries (not really Landau), e.g., propositional theorems
- Chapter 1: 362 - Natural Numbers
- Chapter 2: 555 - Fractions, Rationals
- Chapter 3: 686 - Cuts (Positive Reals)
- Chapter 4: 1136 - Real Numbers
- Chapter 5: 926 - Complex Numbers, Finite Sums, Finite Products

# Original Run by Chapters

- TPS succeeded on 468 of original 9803.
- 378 were from Chapter 0
- 90 were from Chapter 1
- None were from the remaining Chapters 2-5.

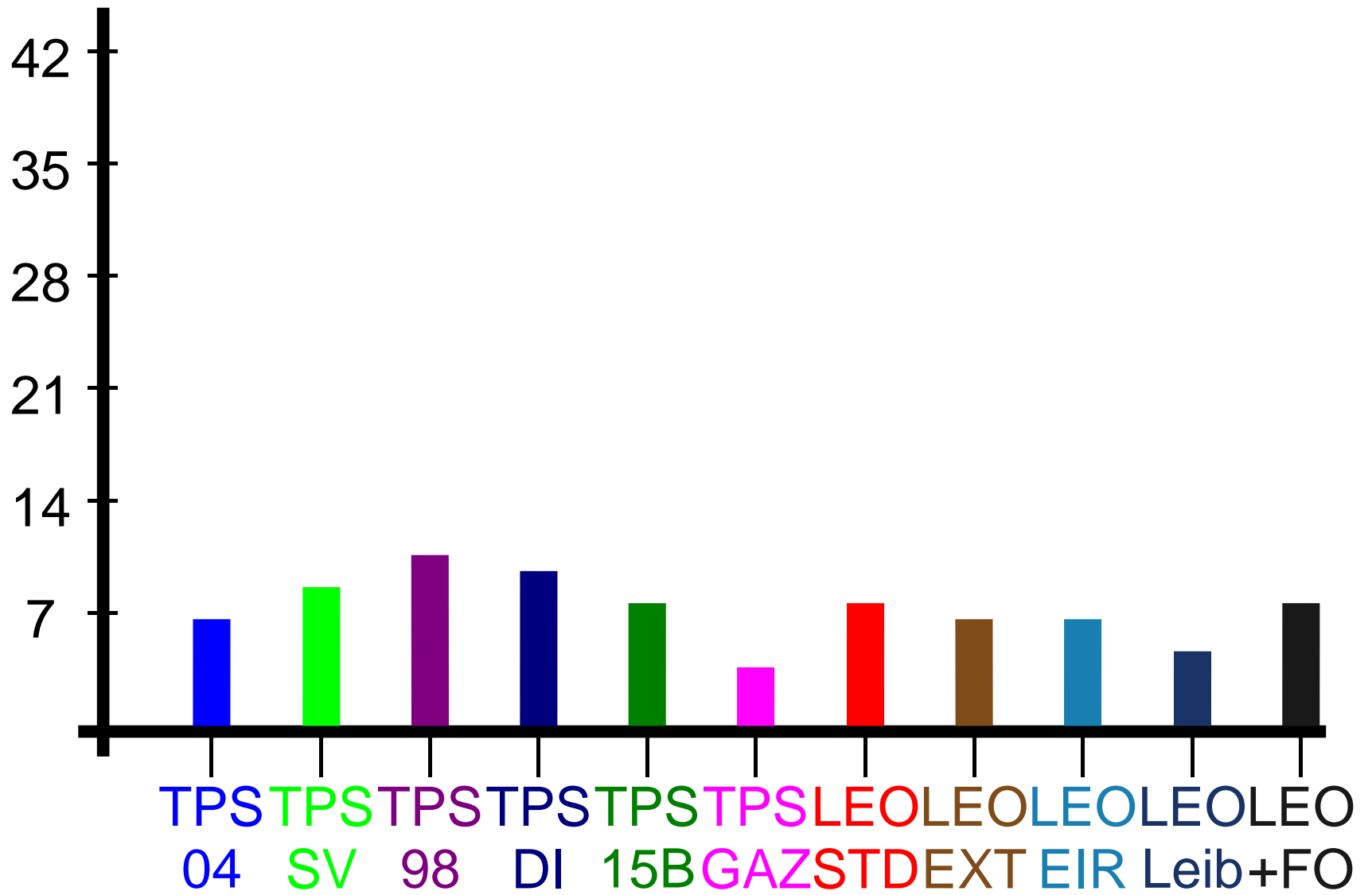
# First Random 42 by Chapters

- TPS succeeded on 10 of the random 42.
- 6 of the successful were from Chapter 0.
- 4 of the successful were from Chapter 1.
- None were from the remaining Chapters 2-5.

# The Second Random 42

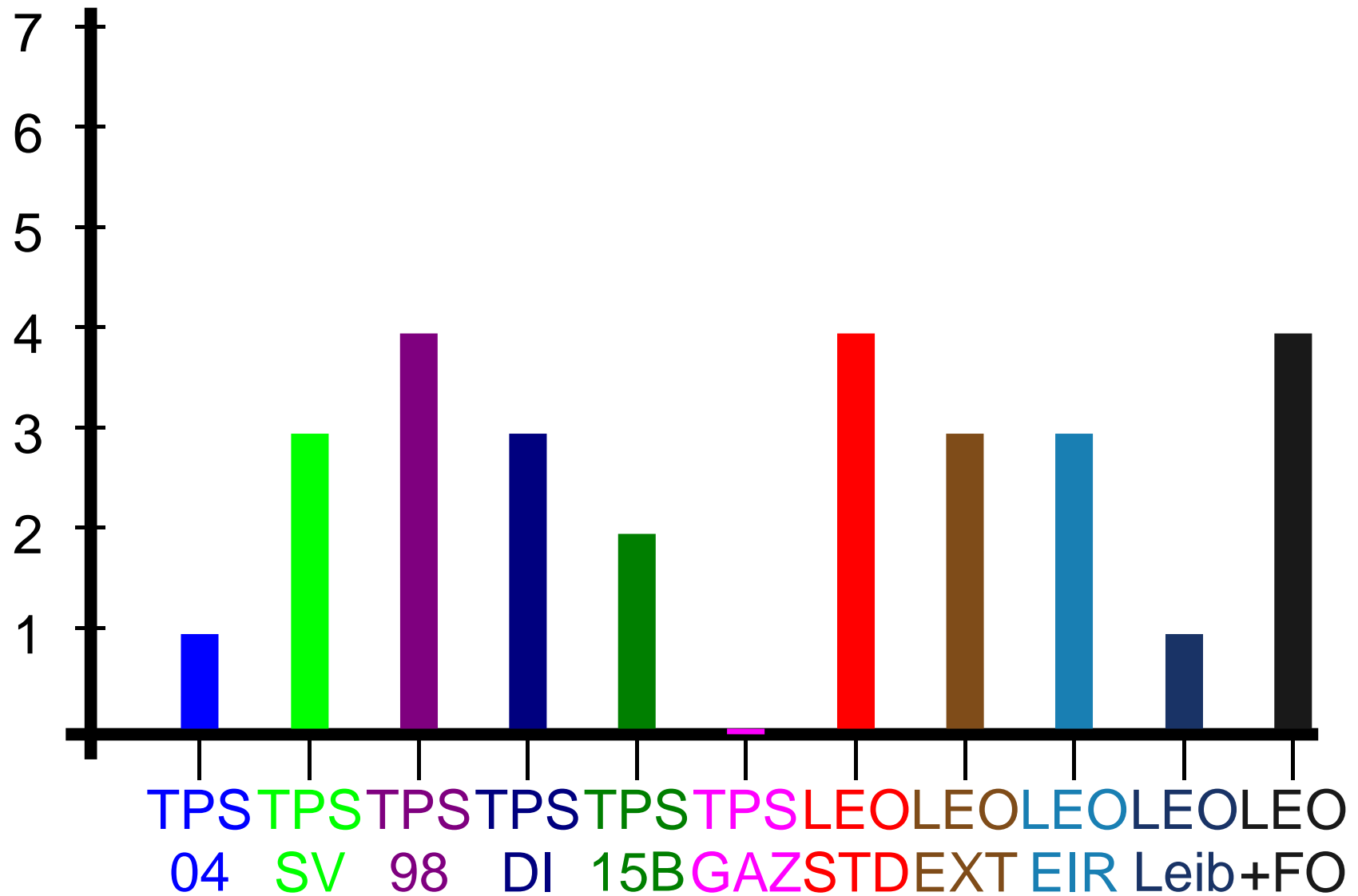
- Randomly Chose 7 problems from each of the 6 chapters.
- Ran TPS with 6 different modes, time limit of 5 minutes.
- Ran LEO (through  $\Omega$ ) with 4 different modes + LEO in cooperation with First-Order Prover Bliksem.
- Secret, Unofficial HOCASC...

# Results





# Results: Chapter 1



# Results: Chapters 2-5

None. Why Not?

- Definition Expansion +  $\beta$ -Reduction Too Expensive
- Definition Expansion Makes The Formulas Too Big.
- For example: TPS tried to expand `_SATZ289B_HTHM` (about finite products of complex numbers) for over  $2\frac{1}{2}$  hours before I gave up.

# Definition Expansion

Minutes

5

4

3

2

1

Time TPS Spends  
Expanding Definitions

Problems

Ch  
0

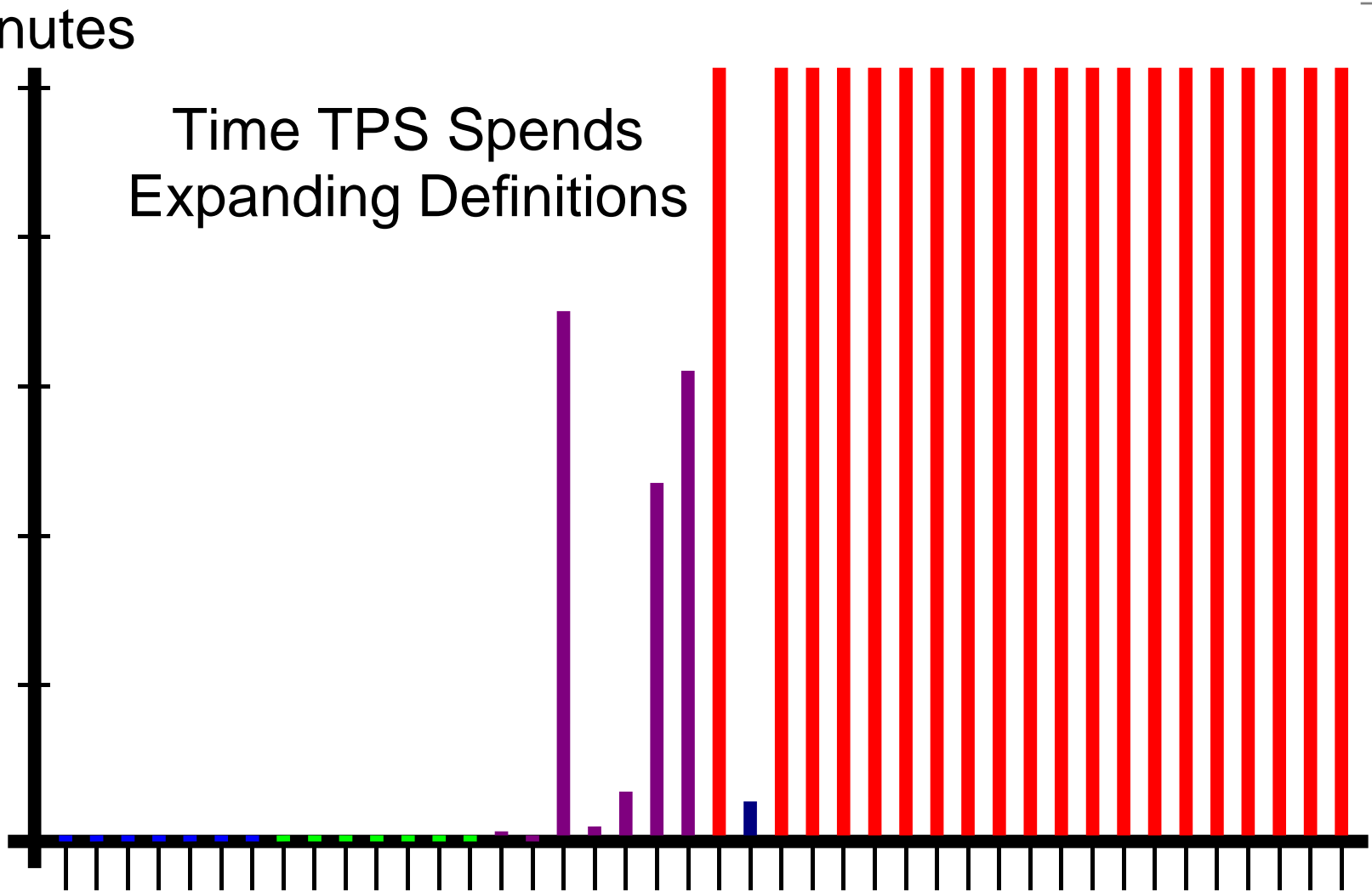
Ch  
1

Ch  
2

Ch  
3

Ch  
4

Ch  
5



# Avoiding Definition Expansion

- TPS can expand definitions of 22 out of 42 (52.4%) within 5 minutes.
- Considered Alternative Representation:  
Replace each abbreviation  $A$  with  $A'$  and add a hypotheses  $A' = D$  where  $D$  is the definition of  $A$ .
- Tried TPS on the 42 Modified Problems.
- Result: Only 3 were still provable at all.

# Avoiding Definition Expansion 2

Second Idea:

- I hacked TPS so only abbreviations appearing in theorem are expanded.
- Matt's MS98-HO-MODE + this hack solves 7 of the 42 problems within 1 minute.
- Only 3 of the 7 from Chapter 0.
- None of the 7 from Chapter 1.
- 2 problems from Chapter 2!
- 2 problems from Chapter 3!

# Avoiding Definition Expansion 3

Third Idea:

- Further hacking of TPS to only expand abbreviations appearing in theorem, and abbreviations appearing in these abbreviations.
- Matt's MS98-HO-MODE + this hack solves 8 of the 42 problems within 1 minute.
- 5 of the 7 from Chapter 0.
- 3 of the 7 from Chapter 1.
- None from Chapters 2-5.
- MS98-HO-MODE solved all these problems (and 3 more) before the hack.

# Avoiding Definition Expansion 4

Fourth Idea (Chris Benz Müller's):

- Tell  $\Omega$ mega to only expand definitions from the current “Chapter” of Landau.
- Results of LEO + Bliksem with this strategy:
- 4 from Chapter 0 (defn strategy irrelevant here).
- 5 from Chapter 1
- 1 from Chapter 2 (`_SATZ73C_HTHM`)
- 2 from Chapter 4

# Easy Hard Example

`_SATZ73C_HTHM` (From Chapter 2)

- TPS can expand definitions in 4 seconds.
- Resulting Formula has
  - 24,774 abstractions
  - 131,378 applications
- $\Omega$ mega tried for more than  $1\frac{1}{2}$  hours.
- Interactive ND Proof in TPS: 46 lines.
- No Automatic Proof by TPS or LEO alone
- LEO + Bliksem can find Automatic Proof if only abbreviations from Landau Chapter 2 are expanded.

# Easy Hard Example

\_SATZ73C\_HTHM Assumptions:

- For fractions  $x, y, z$ , if  $x < y$ , then  $xz < yz$ .
- For fractions  $x, y, z$ , if  $x > y$ , then  $xz > yz$ .
- For fractions  $x, y, z$ , if  $x = y$ , then  $xz = yz$ .
- For fractions  $x, y, z$ , we have
  - if  $xz = yz$ , then  $xz \not< yz$ ,
  - if  $xz > yz$ , then  $xz \not> yz$ , and
  - if  $xz < yz$ , then  $xz \neq yz$ .
- For fractions  $x, y, z$ , either  $x = y$ ,  $x > y$ , or  $x < y$ .

Conclusion: For fractions  $x, y, z$ , if  $xz < yz$ , then  $x < y$ .

# Easy Hard Example

\_SATZ73C\_HTHM Assumptions:

- For fractions  $x, y, z$ , if  $x < y$ , then  $xz < yz$ .
- For fractions  $x, y, z$ , if  $x > y$ , then  $xz > yz$ .
- For fractions  $x, y, z$ , if  $x = y$ , then  $xz = yz$ .
- For fractions  $x, y, z$ , we have
  - if  $xz = yz$ , then  $xz \not< yz$ ,
  - if  $xz > yz$ , then  $xz \not< yz$ , and
  - if  $xz < yz$ , then  $xz \neq yz$ .
- For fractions  $x, y, z$ , either  $x = y$ ,  $x > y$ , or  $x < y$ .

Conclusion: For fractions  $x, y, z$ , if  $xz < yz$ , then  $x < y$ .

# Easy Hard Example

\_SATZ73C\_HTHM Assumptions:

- For fractions  $x, y, z$ , if  $x < y$ , then  $xz < yz$ .
- For fractions  $x, y, z$ , if  $x > y$ , then  $xz > yz$ .
- For fractions  $x, y, z$ , if  $x = y$ , then  $xz = yz$ .
- For fractions  $x, y, z$ , we have
  - if  $xz = yz$ , then  $xz \neq yz$ ,
  - if  $xz > yz$ , then  $xz \neq yz$ , and
  - if  $xz < yz$ , then  $xz \neq yz$ .
- For fractions  $x, y, z$ , either  $x = y$ ,  $x > y$ , or  $x < y$ .

Conclusion: For fractions  $x, y, z$ , if  $xz < yz$ , then  $x < y$ .

# Easy Hard Example

\_SATZ73C\_HTHM Assumptions:

- For fractions  $x, y, z$ , if  $x < y$ , then  $xz < yz$ .
- For fractions  $x, y, z$ , if  $x > y$ , then  $xz > yz$ .
- For fractions  $x, y, z$ , if  $x = y$ , then  $xz = yz$ .
- For fractions  $x, y, z$ , we have
  - if  $xz = yz$ , then  $xz \neq yz$ .
  - if  $xz > yz$ , then  $xz \neq yz$ , and
  - if  $xz < yz$ , then  $xz \neq yz$ .
- For fractions  $x, y, z$ , either  $x = y$ ,  $x > y$ , or  $x < y$ .

Conclusion: For fractions  $x, y, z$ , if  $xz < yz$ , then  $x < y$ .

# Easy Hard Example

\_SATZ73C\_HTHM Assumptions:

- For fractions  $x, y, z$ , if  $x < y$ , then  $xz < yz$ .
- For fractions  $x, y, z$ , if  $x > y$ , then  $xz > yz$ .
- For fractions  $x, y, z$ , if  $x = y$ , then  $xz = yz$ .
- For fractions  $x, y, z$ , we have
  - if  $xz = yz$ , then  $xz \neq yz$ .
  - if  $xz > yz$ , then  $xz \neq yz$ , and
  - if  $xz < yz$ , then  $xz \neq yz$ .
- For fractions  $x, y, z$ , either  $x = y$ ,  $x > y$ , or  $x < y$ .

Conclusion: For fractions  $x, y, z$ , if  $xz < yz$ , then  $x < y$ .

# Hard Example

\_4152\_T31\_HTHM (From Chapter 3)

- This problem is actually unreasonable since one must prove  
1 is a rational number.
- However, neither TPS nor  $\Omega$ mega can expand the definitions within 5 minutes anyway.

# Grundlagen Benchmarks

- Challenge Problems for HO.
- Beginning of a HO TPTP?
- In what format should we store them?
  - Twelf?
  - Automath?
  - Prolog?

# Future Work

- Porting Grundlagen to A Form of Set Theory...
- Comparing Set Theory to Higher-Order Logic