Formalizing Mathematics-In Praxis: First experiences with Isabelle/HOL

ALEXANDRIA: Large-Scale Formal Proof for the Working Mathematician

Angeliki Koutsoukou-Argyraki

Computer Laboratory, University of Cambridge, UK

AITP 2019, Obergurgl, Austria, April 11 2019

Plan

- A comment on my Mathematics background (pen-and-paper Proof Mining) and motivation
- Isabelle/HOL and ALEXANDRIA
- Contributions within ALEXANDRIA
- Difficulties Encountered (Syntax-search-automation)
- Disclaimers and Cautions (to new users)

Proof Mining

G. Kreisel (1950's): *Unwinding of proofs*

"What more do we know if we have proved a theorem by restricted means than if we merely know that it is true ?"

Possible to obtain new quantitative/ qualitative information by logical analysis of proofs of statements of certain logical form. Extraction of constructive information from non-constructive proofs.

Proof Mining

Achieved by using Proof Interpretations.

T_1 transformed into T_2 by transforming every theorem $\phi \in \mathcal{L}(T_1)$ into $\phi^I \in \mathcal{L}(T_2)$ via the proof interpretation I so that $T_1 \vdash \phi \Rightarrow T_2 \vdash \phi^I$ holds.
Then a given proof p of ϕ in T_1 is transformed into a proof p^I of ϕ^I in T_2 by a simple recursion over ϕ in T_1.
This gives new quantitative information.
In particular: For $\phi \equiv \forall x \in \mathbb{N}\ \exists y \in \mathbb{N}\ A(x, y)$ a computational realization of ϕ^I provides a program $P : \forall x \in \mathbb{N}\ A(x, P(x))$. To this end, we need:
$(\forall x \in \mathbb{N}\ \exists y \in \mathbb{N}\ A(x, y))^I \equiv \exists f : \mathbb{N} \to \mathbb{N}\ \forall x \in \mathbb{N}\ A(x, f(x))$, f computable.

Proof Mining

General logical metatheorems by Kohlenbach *et al* use Gödel 's functional Dialectica interpretation and its variations (within specific formal frameworks).
Passage survived by mathematical statements of the logical form $\forall x\ \exists y\ A_{\exists}(x, y)$.
Metatheorems guarantee the extraction of explicit, computable bound on y from the proof.
Bounds are highly uniform : depend only on bounding information on the input data.

Proof Mining

How is the quantitative information(bound) extracted from the proof?

The precise method of extracting the bound is not known a priori. Typically, this is done in three stages : (Important: following process *not automated*. Pen- and-paper! Even though not completely ad hoc is open to the manipulations of the mathematician(s) performing proof mining on a given proof.)

Proof Mining

How is the quantitative information(bound) extracted from the proof?

(i) Write all the statements involved in a formal version using quantifiers.
(ii) The mathematical objects involved must have the correct uniformity. So: we make explicit the quantitative content of their properties (i.e. modulus of continuity for uniform continuity, modulus of accretivity for uniform accretivity, modulus of convexity for uniform convexity, effective irrationality measure for irrationality etc). In that way we obtain quantitative versions of the statements/ lemmas involved.
(iii) Put everything together in a deduction schema just like the one of the original proof, i.e. the structure of the original proof is typically preserved.

Proof Mining

Within past $\approx$ 15 years, U. Kohlenbach *et al* have applied proof mining to : optimization, approximation theory, ergodic theory, fixed point theory, nonlinear analysis in general, and (recently) PDE theory. Applications described as instances of logical phenomena by the general logical metatheorems.

Proof Mining

Within past $\approx$ 15 years, U. Kohlenbach *et al* have applied proof mining to : optimization, approximation theory, ergodic theory,fixed point theory, nonlinear analysis in general, and (recently) PDE theory. Applications described as instances of logical phenomena by the general logical metatheorems.

My motivation (1): What makes a good proof?

- a shorter proof?
- a more "elegant" proof? (subjective...)
- a simpler proof? (Hilbert's 24th problem (1900): *"find criteria for simplicity of proofs, or, to show that certain proofs are simpler than any others. ")*
- Reverse Mathematics: a proof in a weaker subsystem of Z_2 ?
- an interdisciplinary proof ?
- a proof that is easier to combine /reuse ?
- a proof giving better **computational content**?
  - i.e. : bound of lower complexity?
  - i.e. : bound more precise numerically?
  - i.e. : bound more "elegant" ?

My motivation (1): What makes a good proof?

How are the aforementioned proof features related to each other?

**Could we ever ensure that we get the optimal computational content (from a given proof)? ** (e.g. by formalizing first, then proof mining, instead of the other way around?)

A sidenote: a suggestion for a formalizing strategy

Enrich the libraries with formalized proofs where (as much as possible) computational content is made explicit.

This would preserve computational content as the proofs get reused and combined...

...paving the way for automating proof mining!

To this end, constructive proofs are obviously preferable, but there is no need to restrict to only constructive proofs. May opt for proof-mined proofs (that may be even non-constructive!)

A toy example: $\sqrt{2}$ is irrational

A constructive proof by Bishop (see: Bishop, E. : *Schizophrenia in Contemporary Mathematics*, 1973)
$\forall a, b \in \mathbb{Z}^+ |\sqrt{2} - a/b| \geq 1/(4b^2)$ (assuming $a/b \leq 2$).
Proof formalized (A.K-A. and Wenda Li) as:

```
theory sqrt2irrational_constr
  imports
  Main  "HOL-Analysis.Analysis"

   "HOL-Decision_Procs.Approximation"

begin

hide_const floatarith.Max

declare [[smt_timeout = 100000]]
lemma identity_square_dif:
  fixes a b:: "'a:: comm_ring"
  shows " a*a - b*b = (a-b)*(a+b)  "
  by (auto simp add:algebra_simps)
```

A toy example: $\sqrt{2}$ is irrational

```
lemma valuation_dif:
  fixes a b::int assumes  "a ∈ ℤ " and "b ∈ ℤ " and "a>0"
 and "b>0" shows " a^2 ≠2* (b^2)"
proof-
  have is_even: "multiplicity 2 (a^2) = 2* multiplicity 2 a"
    apply (subst prime_elem_multiplicity_power_distrib)
    using ‹a>0› by (auto simp add:prime_imp_prime_elem)
  have val1: "even( multiplicity 2 (a^2))" using is_even by auto
  have *: "multiplicity 2 (2*b^2) = (multiplicity 2 (b^2)) +1"
    apply (subst multiplicity_times_same)
    using ‹b>0› by auto
  have **: "(multiplicity 2 (b^2))+1 = (2* (multiplicity 2 b))+1"
    apply (subst prime_elem_multiplicity_power_distrib)
    using ‹b>0› by (auto simp add:prime_imp_prime_elem)
  have is_odd: "multiplicity 2 (2*b^2) = (2* (multiplicity 2 b))+1"
    using * ** by auto
  have val2: "odd(multiplicity 2 (2*b^2))" using is_odd by auto
  have dif: "multiplicity 2 (a^2) ≠ multiplicity 2 (2*b^2) "
    using val1 val2 by auto
  show ?thesis using dif by auto
qed
```

A toy example: $\sqrt{2}$ is irrational

```
theorem sqrt2isirrational_Bishop:
  fixes a b ::int assumes "a ∈ ℤ" and "b ∈ ℤ" and  "a >0" and "b >0"
      and "a/b ≤2"
    shows "¦ a/b - sqrt(2)¦ ≥  1/(4*b^2)"

proof-
  have *:"¦ a/b - sqrt(2)¦ *¦ a/b + sqrt(2)¦ ≥ 1/b^2"
  proof -
    have "¦ a/b - sqrt(2)¦ *¦ a/b + sqrt(2)¦ =¦ (a/b)^2 - 2¦ "
      using identity_square_dif
      by (metis abs_mult abs_numeral real_sqrt_mult_self semiring_normalization_rules(29))
    also have "... = ¦ a^2/b^2 - 2 ¦ "
      by (simp add: power_divide)
    also have "... = (1/b^2)* (b^2)*¦( a^2/b^2) - 2¦"
      using assms(4) by auto
    also have "... =  (1/b^2)*¦(b^2)*( a^2/b^2) -(b^2)* 2 ¦   "
      using ‹b>0› by (simp add:divide_simps)
    also have "... = (1/b^2)*¦ a^2 -(b^2)* 2 ¦  "
      by auto
    also have "... ≥ (1/b^2)"
```

A toy example: $\sqrt{2}$ is irrational

```
  proof -
    have " a^2 ∈ ℤ " using  ‹ a ∈ ℤ › by auto
    moreover have "2*( b^2) ∈ ℤ " using ‹ b ∈ ℤ › by auto
    moreover have "a^2 ≠2* (b^2)"
      by (simp add: assms(1) assms(2) assms(3) assms(4) valuation_dif)
    ultimately have "¦ a^2 - 2*(b^2) ¦ ≥1 " by linarith
    then show ?thesis
      using  ‹ ¦ a^2 - 2*(b^2) ¦ ≥1›
      by (metis (no_types, hide_lams) divide_right_mono mult.commute mult.left_neutral
          of_int_1_le_iff of_int_mult semiring_normalization_rules(29) times_divide_eq_right
          zero_le_power2)
  qed
  finally show ?thesis .
qed

have "¦ a/b - sqrt(2)¦ ≥  1/(4*b^2)"
proof -
  have "1/(4*b^2) ≤ (1/¦2 + sqrt(2)¦) *( 1/b^2)"
```

A toy example: $\sqrt{2}$ is irrational

```
    proof -
      have "2 + sqrt(2) < 4"
        by (simp add: sqrt2_less_2)
      then show ?thesis using ‹b>0›
        apply (simp add:divide_simps)
        by (smt mult_less_0_iff not_sum_power2_lt_zero real_sqrt_gt_0_iff)
    qed
    also have "... ≤ (1/¦ a/b + sqrt(2)¦) *( 1/b^2)"
    proof -
      define t1 t2 where "t1=¦ a/b + sqrt(2)¦" and "t2=¦2 + sqrt 2¦"
      have "t1 ≤ t2"
        using assms(3) assms(4) assms(5) unfolding t1_def t2_def by auto
      moreover have "t1 >0"
        unfolding t1_def using "*" by auto
      ultimately show ?thesis

        apply (fold t1_def t2_def)
        using ‹b>0 ›by (simp add:divide_simps)
    qed
    also have "... ≤ ¦ a/b - sqrt(2)¦"
    proof -
      have "¦ a/b + sqrt(2)¦ > 0" using * by auto
      then show ?thesis using * ‹b>0› by (auto simp add:divide_simps)
    qed
    finally show ?thesis.
  qed
qed
```

Motivation (2): higher standards of rigour and correctness needed

"...We believe that when later generations look back at the development of mathematics one will recognise four important steps: (1) the Egyptian-Babylonian-Chinese phase, in which correct computations were made, without proofs; (2) the ancient Greeks with the development of proof; (3) the end of the nineteenth century when mathematics became rigorous; (4) the present, when mathematics (supported by computer) finally becomes fully precise and fully transparent."

Barendregt, H. and Wiedijk, F., *The Challenge of Computer Mathematics*, Transactions A of the Royal Society 363 no. 1835, 2351-2375 (2005)

Motivation (3)

Reimagining mathematical practice in light of new AI developments. New way of working will shape our way of thinking.

An anecdote indicative of the current climate: the panel discussion of the workshop *"Foundations in Mathematics: Modern Views"* (April 2018, Munich) that attracted young (mostly student-level) mathematicians, philosophers and logicians, the dominant view discussed arguing for the importance of exploring the foundations of mathematics was their significance for computerized mathematical proofs which among the participants of the discussion **was regarded as an inevitable development**.

ALEXANDRIA

Large-scale formal proof for the working mathematician

5-year ERC project (since Sept. 2017) Computer Laboratory, University of Cambridge, UK.
PI: Larry Paulson. Participating : Wenda Li, Anthony Bordg, Yiannos Stathopoulos(to join soon), A. K.-A., interns: Martin Baillon and Paulo Emílio de Vilhena, (and many more friends in Cambridge). An international community of Isabelle experts in touch through the Isabelle mailing lists.
The proof assistant Isabelle/HOL (developed by Larry Paulson and Tobias Nipkow) used to conduct proofs in the structured proof language *Isar* allowing for proof text understandable both by humans and machines. Simple types. Sledgehammer.

ALEXANDRIA

Large-scale proof for the working mathematician

The goals of ALEXANDRIA are to contribute to:

- Expanding Libraries of formal proofs (short-term)
  - (a) formalize proofs of undergraduate level mathematics- see: http://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/index.html
  - (b) formalize research level proofs-see: https://www.isa-afp.org
- Improving Automation (short-term)
- Consolidating/organizing libraries of formal proofs (short-term)
- Improving Search(short-term)
- Verification of research level mathematics (long-term)
- Assisting mathematicians (through automation and search) with writing new research level proofs (long-term)

Theorem

(Theorem 3 in : Hančl, J. : Irrational Rapidly Convergent Series, Rend. Sem. Mat. Univ. Padova, Vol. 107 (2002).) Let $A \in \mathbb{R}$ *with* $A > 1$. *Let* $\{d_n\}_{n=1}^{\infty} \in \mathbb{R}$ *with* $d_n > 1$ *for all* $n \in \mathbb{N}$. *Let* $\{a_n\}_{n=1}^{\infty}, \{b_n\}_{n=1}^{\infty} \in \mathbb{Z}^+$ *such that :* (1) $\lim_{n\to\infty} a_n^{\frac{1}{2^n}} = A$, *for all sufficiently large* $n \in \mathbb{N}$ *:* (2) $\frac{A}{a_n^{\frac{1}{2^n}}} > \prod_{j=n}^{\infty} d_j$ *and*

(3) $\lim_{n\to\infty} \frac{d_n^{2^n}}{b_n} = \infty$. *Then* $\sum_{n=1}^{\infty} \frac{b_n}{a_n}$ *is an irrational number.*

Corollary

(Corollary 2 in :Hančl, J. : Irrational Rapidly Convergent Series, Rend. Sem. Mat. Univ. Padova, Vol. 107 (2002).)Let $A \in \mathbb{R}$ *with* $A > 1$. *Let* $\{a_n\}_{n=1}^{\infty}, \{b_n\}_{n=1}^{\infty} \in \mathbb{Z}^+$ *such that :* $\lim_{n\to\infty} a_n^{\frac{1}{2^n}} = A$ *and for all sufficiently large* $n \in \mathbb{N}$ *(in particular* $n \geq 6$*)* $a_n^{\frac{1}{2^n}}(1 + 4(2/3)^n) \leq A$ *and* $b_n \leq 2^{(4/3)^{n-1}}$. *Then* $\sum_{n=1}^{\infty} \frac{b_n}{a_n}$ *is an irrational number.*

Consequence of the theorem by setting $d_n = 1 + (2/3)^n$.

Theorem

(Theorem 2.1 in :Hančl, J. and Rucki, P. : The Transcendence of Certain infinite Series, Rocky Mountain Journal of Mahematics, Vol. 35, No 2, (2005)). Let $\delta \in \mathbb{R}$ *with* $\delta > 0$. *Let* $\{a_k\}_{k=1}^{\infty}, \{b_k\}_{k=1}^{\infty} \in \mathbb{Z}^+$ *such that :*
$\limsup_{k\to\infty} \frac{a_{k+1}}{(a_1 a_2 \ldots a_k)^{2+\delta}} \frac{1}{b_{k+1}} = \infty$ *and* $\liminf_{k\to\infty} \frac{a_{k+1}}{a_k} \frac{b_k}{b_{k+1}} > 1$.
Then $\sum_{k=1}^{\infty} \frac{b_k}{a_k}$ *is a transcendental number.*

Theorem

(Theorem 2.2 in: Hančl, J. and Rucki, P. : The Transcendence of Certain infinite Series, Rocky Mountain Journal of Mahematics, Vol. 35, No 2, (2005)). Let $\delta, \epsilon \in \mathbb{R}$ *with* $\delta > 0, \epsilon > 0$. *Let* $\{a_k\}_{k=1}^{\infty}, \{b_k\}_{k=1}^{\infty} \in \mathbb{Z}^+$, *such that :*
$\limsup_{k\to\infty} \frac{a_{k+1}}{(a_1 a_2 \ldots a_k)^{2+2/\epsilon+\delta}} \frac{1}{b_{k+1}} = \infty$ *and for every sufficiently large* k $\sqrt[1+\epsilon]{\frac{a_{k+1}}{b_{k+1}}} \geq \sqrt[1+\epsilon]{\frac{a_k}{b_k}} + 1$.
Then $\sum_{k=1}^{\infty} \frac{b_k}{a_k}$ *is a transcendental number.*

The proof uses Roth's theorem on diophantine approximations to algebraic numbers (Roth, K. F. , *Rational Approximations to Algebraic Numbers*, Mathematika, Vol. 2. Part 1, No 3, 1955) the proof of which has not been formalized and was implemented as an assumption.

Theorem

(Roth, 1955) Let α be any algebraic number, not rational. If $|\alpha - \frac{h}{q}| < \frac{1}{q^\kappa}$ has an infinity of solutions in integers h, q $(q > 0)$ then $\kappa \leq 2$.

ALEXANDRIA

...other projects :

- Octonion development (after Paulson's Quaternion development, see AFP)
- currently working on formalizing irrationality criteria for infinite series by Erdős (with Wenda Li)
- Collecting suggestions for the new version(s) of Isabelle/HOL wrt improvements in automation and additions in the library.
- Manual for the Analysis Library (with TU Munich, ongoing).
- Intelligent search, automated user support (with Yiannos Stathopoulos and Wenda Li)

ALEXANDRIA

Lawrence Paulson

Reorganizing the Libraries, generalizing and improving the proofs. Several major projects incorporated into the main Analysis and Algebra libraries :

- The theory of infinite products
- Measure theory including change-of-variables theorems for integration
- Abstract topology: Hausdorff spaces, etc.
- Algebra: core topics in group theory
- Algebraic topology: Homology theory (pending)

Moreover:

- An Isabelle/HOL formalization of Green's Theorem (AFP, Abdulaziz and Paulson)
- The Prime Number Theorem (AFP, Eberl and Paulson)

Contributions in Computer Algebra : Implemented verified procedures for counting complex roots of polynomials in a region, also in the difficult case where the roots lie on the border of the region. This is important as numerous engineering problems are based on reasoning about complex roots of certain characteristic polynomials.

- Li and Paulson. Counting Polynomial Roots in Isabelle/HOL: A Formal Proof of the Budan-Fourier Theorem. CPP 2019
- Li and Paulson. Evaluating winding numbers and counting complex roots through Cauchy indices in Isabelle/HOL. J. Automated Reasoning (in press)
- Li, Passmore and Paulson. Deciding univariate polynomial problems using untrusted certificates in Isabelle/HOL. J. Automated Reasoning 62 (2019)

In the AFP:

- Evaluate Winding Numbers through Cauchy Indices
- Count the Number of Complex Roots
- The Budan-Fourier Theorem and Counting Real Roots with Multiplicity

ALEXANDRIA

Anthony Bordg

Background in Homotopy Type Theory, Category Theory, Coq experience, contributed to UniMath library.
In the AFP:

- Projective Geometry (Hessenberg's theorem, Desargues's theorem)
- The Localization of a Commutative Ring

Currently in progress:

- A library of tensor analysis
- The mathematics of quantum computing

ALEXANDRIA

Martin Baillon and Paulo Emílio de Vilhena

Interns from École Polytechnique de Paris (20-week internships, partly supported by the project) Worked on formalization of abstract algebra, both reorganised and extended. Formalized a significant part of Galois theory. This work was incorporated into Isabelle's Algebra library (2018).

Difficulties Encountered

I. Syntax

Isar: intuitive structure, easily readable. jEdit interface is very user-friendly. Structured proofs is a major advantage. Certain features that may seem surprising to a new user. Examples:

- proof patterns: have $a < b$ also have ... $< c$ finally show $a < c$ by auto , have $a < b$ moreover have ... $< c$ ultimately show $a < c$ by auto
- must always include type information ! arabic numb.
- symbols for exponentiation differ according to type of base ($^\wedge$ or powr) , switch type from integer to real (of_int, of_real) e.g. with division
- keywords like “where”, “that ...when” , “at_top”, “sequentially”
- join and meet operators for lattices: $\wedge, \vee$ instead of $\sqcap, \sqcup$, the absolute value symbol, arrows
- overall the extremely high level of detail required.

Difficulties Encountered

II. Search

"find_theorems" is not always helpful to the user. For instance, many fundamental search words (e.g. "Borel", "Zorn", "Gauss", "product", "inverse", "operator", "Hilbert", "Lebesgue", "derivative", "Euclidean", "rational", "polynomial", "series", "Weierstrass", "Noether", "summation", "fraction", "supremum", "infimum", "pythagorean", "multiplication", "converge", "convergence", "mapping") give no results.

Difficulties Encountered

II. Search

Manual search in the Library can be time-consuming :

1. fast growing size of the Library, especially the Analysis Library.
2. general difficulty in classifying mathematical knowledge (very often borders between disciplines are unclear)
3. in Math literature: different names in different contexts for the same notion
4. in Math literature: same name for different notions

Another big challenge: Searching for proof patterns and algorithms!

Difficulties Encountered

III. Automation

E.g:

```
theorem distri:
  fixes c b a ::nat
  shows "(5/c^2)* b^2 * ¦(a^2/ b^2) -2 ¦ = ¦(5/c^2)*(  b^2 )* (a^2/ b^2) -(5/c^2)*  ( b^2)* 2 ¦ "
  sorry

theorem simpli:
  fixes
 c  a   ::"int ⇒ int" and s b k  ::nat
  assumes "∀ n ≥ s. (  ¦ ( a n) *( c n) ¦*k*b^4   )/ b  = c (n+1) "  and "b  ≠ 0 "
  shows "∀ n ≥ s.  ¦ ( a n) *( c n)    ¦* k*b^3  = c (n+1) "
  sorry

lemma factor:
  fixes a b c d e :: real
  assumes "(a-1) ≥ 1" and " (b^2 + a*b*c )≥1" and "c^2 ≥1" and "d^9 ≥1" and "e^7 ≥ 1"
  shows  "(a-1)*(b^2 + a*b*c )* c^2 * d^9 * e^7 ≥ 1 "
  sorry
```

IV. Using the already formalized material

Disclaimers and Cautions (to new users)

Mechanization of Mathematics is not a Panacea for Correctness! Use proof assistants responsibly!

Verifying mathematics is reminiscent of a relative consistency proof (*not a problem for formalists*), on two levels:

1. core of the system, underlying architecture
2. correctness of mathematical assumptions

Possible to make undetected mistakes in very naive ways: proving something different than what was initially intended or claimed by either (a) using a misleading name of the proved statement (b) even a typo like a misplaced parenthesis e.g. showing $f(n+1)$ instead of $f(n)+1$.
Also remember the explosion principle (ex falso sequitur quodlibet)

Disclaimers and Cautions (to new users)

Different kinds of "wrong" in Mathematics- Use proof assistants responsibly!

Proving a conclusion that is too general (logically correct: $A \rightarrow A \vee B$ but mathematically undesirable)

```
theorem wrongroot:
  fixes  x::real
  assumes "x^2 -1=0  "
  shows "x=1∨x=-1∨x=4 "
  using assms
apply (auto simp add:power2_eq_square algebra_simps)

  using square_eq_1_iff by blast
```

Disclaimers and Cautions (to new users)

Different kinds of "wrong" in Mathematics- Use proof assistants responsibly!

Using a superfluous assumption (logically correct: $A \wedge B \longrightarrow B$ but mathematically undesirable)

```
theorem superflu:
  fixes x y z A::real
assumes "x= (y-1)^2 +4*z "and "y=A+2" and  "z=5*y"
  shows
 "x= (y-1)^2 +20*y "
  using assms apply simp done
```

A logical inconsistency in the assumptions

```
theorem incoherent:

assumes " ∀x. P x" and " ∃ x. ¬P x"
  shows " ∀x. P x  "
  using assms by auto
```

Disclaimers and Cautions (to new users)

Different kinds of "wrong" in Mathematics- Use proof assistants responsibly!

Lack of precision when approximating

```
theorem notprecise1:
  fixes y::"real⇒real "
assumes "y  = (λx. cos x + sin x) "
  and "∀ x. (sin x=x-x^3/6 ∧ cos x=(1::real)-x^2/2) "
shows
" y x = x+1-x^3/6-x^2/2 "
  using assms by auto
```

Requirement of an additional assumption

```
theorem simply:
  fixes a b c d::real
  assumes "a * b = c * d"
(*  and "d ≠0"*)
  shows " c=(a * b)/d"
  sorry

(*  using assms apply simp done*)
```

Disclaimers and Cautions (to new users)

Different kinds of "wrong" in Mathematics- Use proof assistants responsibly!

Assuming (a) wrong fact(s)/ assumption(s) that cannot be fulfiled

```
theorem wrongfact:
  fixes  y z  ::nat
  assumes
"prime(y*z) " and "y>2 " and "z>2 "
  shows "prime(y*z*1)"
  using assms by auto

theorem wrongfact2:
  assumes " pi ∈ Rats " and  "(k::nat) ∈ Rats "
  shows " pi*k ∈ Rats "
  using assms by auto

theorem wrongfact3:
  assumes " sqrt 2 ∈ Rats "
  shows "1/ sqrt 2 ∈ Rats "
  using assms by auto

theorem unfulfiledassum:
  fixes  f::"nat⇒nat"
  assumes "∀ y x.(x ≤ y⟶f y ≤  f x)" and "f = (λx. 1+x) "
shows "f 2≤ f 1 "
  using assms by auto
```

Thank you