IMO Grand Challenge: can we contribute?

A talk by Nikolay V. Shilov (shiloviis@mail.ru)

for joint meeting of XAI (https://t.me/XAISM) and

STEP (https://persons.iis.nsk.su/en/STEP-2023, https://t.me/+cgiEKFcC1TliNzZi)

October 25, 2023

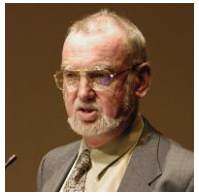
IMO on IU Math Club

International Mathematical Olympiads…

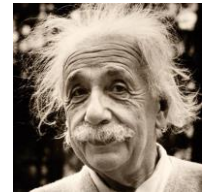
- International Mathematical Olympiad Foundation: https://imof.co/.
- International Mathematical Olympiad: https://www.imo-official.org/.
- IMO Problems (1959-…): https://www.imo-official.org/problems.aspx.

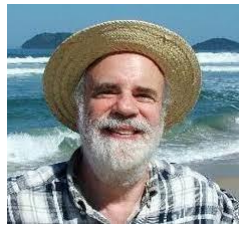
Welcome to IU Math Club!

(https://t.me/InnoMath)



Do you know who these people are?





(https://t.me/+QDS2Fy-HMa8wZWE6)

IMO2019 Problems (have been discussed by IU Math Club)

IMO 2019
60TH INTERNATIONAL MATHEMATICAL OLYMPIAD

English (eng), day 1

Tuesday, July 16, 2019

Problem 1. Let $\mathbb{Z}$ be the set of integers. Determine all functions $f\colon \mathbb{Z} \to \mathbb{Z}$ such that, for all integers a and b,

$$f(2a) + 2f(b) = f(f(a+b)).$$

Problem 2. In triangle ABC, point A_1 lies on side BC and point B_1 lies on side AC. Let P and Q be points on segments AA_1 and BB_1, respectively, such that PQ is parallel to AB. Let P_1 be a point on line PB_1, such that B_1 lies strictly between P and P_1, and $\angle PP_1C = \angle BAC$. Similarly, let Q_1 be a point on line QA_1, such that A_1 lies strictly between Q and Q_1, and $\angle CQ_1Q = \angle CBA$.

Prove that points P, Q, P_1, and Q_1 are concyclic.

Problem 3. A social network has 2019 users, some pairs of whom are friends. Whenever user A is friends with user B, user B is also friends with user A. Events of the following kind may happen repeatedly, one at a time:

> Three users A, B, and C such that A is friends with both B and C, but B and C are not friends, change their friendship statuses such that B and C are now friends, but A is no longer friends with B, and no longer friends with C. All other friendship statuses are unchanged.

Initially, 1010 users have 1009 friends each, and 1009 users have 1010 friends each. Prove that there exists a sequence of such events after which each user is friends with at most one other user.

IMO 2019
60TH INTERNATIONAL MATHEMATICAL OLYMPIAD

English (eng), day 2

Wednesday, July 17, 2019

Problem 4. Find all pairs (k, n) of positive integers such that

$$k! = (2^n - 1)(2^n - 2)(2^n - 4)\cdots(2^n - 2^{n-1}).$$

Problem 5. The Bank of Bath issues coins with an H on one side and a T on the other. Harry has n of these coins arranged in a line from left to right. He repeatedly performs the following operation: if there are exactly $k > 0$ coins showing H, then he turns over the k^{th} coin from the left; otherwise, all coins show T and he stops. For example, if $n = 3$ the process starting with the configuration THT would be $THT \to HHT \to HTT \to TTT$, which stops after three operations.

(a) Show that, for each initial configuration, Harry stops after a finite number of operations.

(b) For each initial configuration C, let $L(C)$ be the number of operations before Harry stops. For example, $L(THT) = 3$ and $L(TTT) = 0$. Determine the average value of $L(C)$ over all 2^n possible initial configurations C.

Problem 6. Let I be the incentre of acute triangle ABC with $AB \neq AC$. The incircle ω of ABC is tangent to sides BC, CA, and AB at D, E, and F, respectively. The line through D perpendicular to EF meets ω again at R. Line AR meets ω again at P. The circumcircles of triangles PCE and PBF meet again at Q.

Prove that lines DI and PQ meet on the line through A perpendicular to AI.

What special about IMO2019 problems?

- Problems 1 and 5(a) are the Computer Science problems about
  - recursion elimination,
  - algorithm termination.
- The first topic (recursion elimination) has been touched in *Discrete Mathematics & Logic* course (BS-I),
- the second (algorithm termination) – has been touched in *Introduction to Programming* course (BS-I) and just a week ago in course on *Differential Equations* (BS-II).

What is the most special about 60th IMO – IMO2019? – IMO Grand Challenge!

- The International Mathematical Olympiad (IMO) is perhaps the most celebrated mental competition in the world and as such is among the ultimate grand challenges for Artificial Intelligence (AI).
- The challenge: build an AI that can win a gold medal in the competition.

https://imo-grand-challenge.github.io/

Note: The IMO Grand Challenge is not affiliated with the IMO Foundation https://imof.co/.

IMO Grand Challenge:
is it a Grand Challenge indeed?

IMO Grand Challenge

(https://imo-grand-challenge.github.io/)

- To remove ambiguity about the scoring rules, we propose the formal-to-formal (F2F) variant of the IMO: the AI receives a formal representation of the problem (in the Lean Theorem Prover, https://leanprover.github.io/), and is required to emit a formal (i.e., machine-checkable) proof. We are working on a proposal for encoding IMO problems in Lean and will seek broad consensus on the protocol.

IMO Grand Challenge – Other proposed rules

(https://imo-grand-challenge.github.io/)

- *Credit*. Each proof certificate that the AI produces must be checkable by the Lean kernel in 10 minutes (which is approximately the amount of time it takes a human judge to judge a human's solution). Unlike human competitors, the AI has no opportunity for partial credit.
- *Resources*. The AI has only as much time as a human competitor (4.5 hours for each set of 3 problems), but there are no other limits on the computational resources it may use during that time.

IMO Grand Challenge – Other proposed rules

(https://imo-grand-challenge.github.io/)

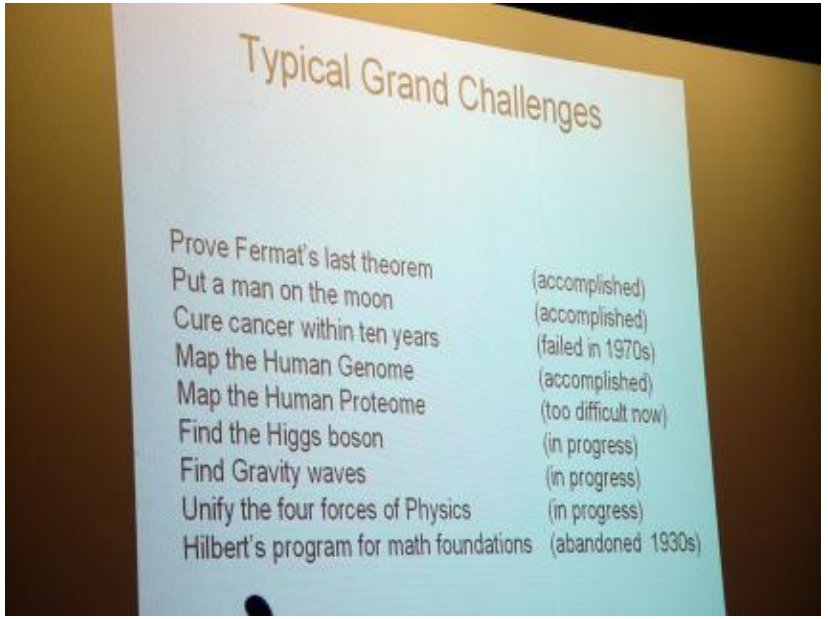
- *Reproducibility*. The AI must be open-source, released publicly before the first day of the IMO, and be easily reproduceable. The AI cannot query the Internet.
- *Challenge*. The grand challenge is to develop an AI that earns enough points in the F2F version of the IMO (described above) that, if it were a human competitor, it would have earned a gold medal.
- Note: this is only a preliminary proposal for the rules. To get involved in the discussion, please join our Zulip channel.

IMO Grand Challenge – Committee

(https://imo-grand-challenge.github.io/)

- Daniel Selsam (Microsoft Research→ OpenAI)
- Leonardo de Moura (Microsoft Research)
- Kevin Buzzard (Imperial College London)
- Reid Barton (University of Pittsburgh)
- Percy Liang (Stanford University)
- Sarah Loos (Google AI)
- Freek Wiedijk (University of Nijmegen)

Where to learn about Grand Challenges?



- Hoare T. *The Verifying Compiler: A Grand Challenge for Computing Research*. In: Broy M., Zamulin A.V. (eds) International Andrei Ershov Memorial Conference on Perspectives of System Informatics PSI 2003. Lecture Notes in Computer Science, vol 2890, pp 1-12. (https://doi.org/10.1007/978-3-540-39866-0_1)

A shorter (preliminary?) version to read

- Hoare T. The Verifying Compiler: A Grand Challenge for Computing Research. In: Böszörményi L., Schojer P. (eds) Joint Modular Languages Conference JMLC 2003. Lecture Notes in Computer Science, vol 2789, pp 25-35. (https://doi.org/10.1007/978-3-540-45213-3_4)

The Verifying Compiler: A Grand Challenge for Computing Research*

Tony Hoare

Microsoft Research Ltd., 7 J.J. Thomson Ave, Cambridge CB3 0FB, UK
thoare@microsoft.com

Abstract. I propose a set of criteria which distinguish a grand challenge in science or engineering from the many other kinds of short-term or long-term research problems that engage the interest of scientists and engineers. As an example drawn from Computer Science, I revive an old challenge: the construction and application of a verifying compiler that guarantees correctness of a program before running it.

1 Introduction

The primary purpose of the formulation and promulgation of a grand challenge is to contribute to the advancement of some branch of science or engineering. A grand challenge represents a commitment by a significant section of the research community to work together towards a common goal, agreed to be valuable and achievable by a team effort within a predicted timescale. The challenge is formulated by the researchers themselves as a focus for the research that they wish to pursue in any case, and which they believe can be pursued more effectively by advance planning and coordination. Unlike other common kinds of research initiative, a grand challenge should not be triggered by hope of short-term economic, commercial, medical, military or social benefits; and its initiation should not wait for political promotion or for prior allocation of special funding. The goals of the challenge should be purely scientific goals of the advancement of skill and of knowledge. It should appeal not only to the curiosity of scientists and to the ambition of engineers; ideally it should appeal also to the imagination of the general public; thereby it may enlarge the general understanding and appreciation of science, and attract new entrants to a rewarding career in scientific research.

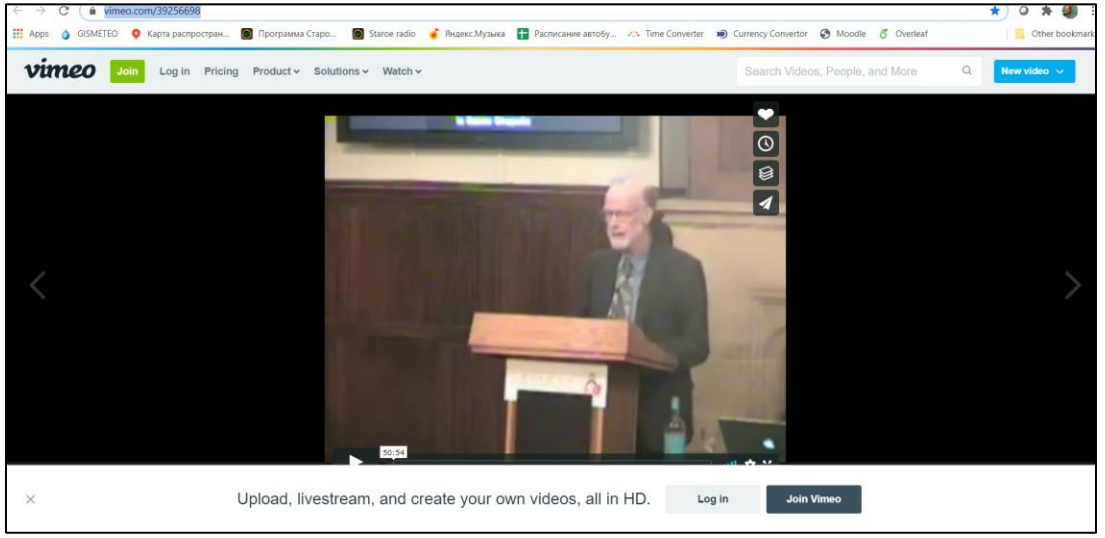
An opportunity for a grand challenge arises only rarely in the history of any particular branch of science. It occurs when that branch of study first reaches an adequate level of maturity to predict the long-term direction of its future progress, and to plan a project to pursue that direction on an international scale. Much of the work required to

* Reprinted with the kind allowance of Springer Verlag from the proceedings of ECOOP 2003, LNCS 2743

L. Böszörményi and P. Schojer (Eds.): JMLC 2003, LNCS 2789, pp. 25–35, 2003.

If you would like – just watch!

- The Verifying Compiler: A Grand Challenge for computing research of the 21st - Tony Hoare - Gresham College Lectures, 2012 (https://vimeo.com/39256698)



Tony Hoare on The Verifying Compiler as a Grand Challenge for Computing Research

(From presentation for Redmond Faculty Summit Redmond Faculty Summit August 2, 2004, available at https://pdfs.semanticscholar.org/469d/cadd1c347e9a1ee05a5950cd47e45387d553.pdf)

Typical Grand Challenges	
Prove Fermat's last theorem	(accomplished)
Put a man on the moon	(accomplished)
Cure cancer within ten years	(failed in 1970s)
Find the Higgs boson	(in progress)
Map the Human Genome	(accomplished)

In Computing Science	
Prove that P is not equal to NP	(open)
The Turing test	(outstanding)
A championship chess program	(completed 1997)
The verifying compiler	(in progress)

Tony Hoare on The Verifying Compiler as a Grand Challenge for Computing Research

(From presentation for Redmond Faculty Summit Redmond Faculty Summit August 2, 2004, available at https://pdfs.semanticscholar.org/469d/cadd1c347e9a1ee05a5950cd47e45387d553.pdf)

A Grand Challenge

- May take fifteen years or more to complete
- Attracts world-wide participation
- Has a clear test of success or failure
- Pursues scientific ideals
- Enlarges fundamental understanding
  - pursues scientific ideals
  - in an area of significance to mankind

A Grand Challenge needs

- Maturity of the state of the art
- Support from the scientific community
- Commitment from the teams that engage in it
- Understanding from funding agencies

Tony Hoare on The Verifying Compiler as a Grand Challenge for Computing Research

(From presentation for Redmond Faculty Summit Redmond Faculty Summit August 2, 2004, available at https://pdfs.semanticscholar.org/469d/cadd1c347e9a1ee05a5950cd47e45387d553.pdf)

The Verifying Compiler

A verifying compiler uses automated mathematical and logical reasoning to check the correctness of the programs that it compiles. Correctness is specified by types, assertions, specifications, and other redundant annotations that accompany the code of the program.

Test of success

- Significant software products are analysed mechanically and formally verified,
  - ranging from safety-critical and embedded codes to open source and legacy applications
  - verified at an appropriate level of safety/soundness/security/service.
- Verified programs replace existing versions in use
  - subsequent evolution will maintain correctness.
- Verification is integrated into commercial toolsets

Tony Hoare on The Verifying Compiler as a Grand Challenge for Computing Research

(From presentation for Redmond Faculty Summit Redmond Faculty Summit August 2, 2004, available at https://pdfs.semanticscholar.org/469d/cadd1c347e9a1ee05a5950cd47e45387d553.pdf)

Fundamental understanding

- What is this program for?
  - Its specification tells you its function
- How does it work?
  - Annotation at interfaces explains how.
- Why does it work?
  - The theory of programming explains why.
- Are the answers accurate?
  - A verifying compiler provides a reliable check

Scientific ideals

- The project complements commercially motivated evolution of existing products
  - which follow market demand
  - to discover more faults in existing programs.
  - appeal to current educational level of programmers
  - with many pictures
- But academic research pursues ideals of purity, accuracy, completeness -- and correctness
  - far beyond the current needs of the market place

Tony Hoare on The Verifying Compiler as a Grand Challenge for Computing Research

(From presentation for Redmond Faculty Summit Redmond Faculty Summit August 2, 2004, available at https://pdfs.semanticscholar.org/469d/cadd1c347e9a1ee05a5950cd47e45387d553.pdf)

Beneficial

- The understanding and knowledge gained on completion of the project promises benefit to mankind.
- Reduction in program errors could even now save $22 to $60 billion per year in US (US Dept. Commerce Planning Report 02-03, May 2002).

The team must include …

- Programming theorists
- Programming tool-set builders
- Compiler writers and optimisers
- Design pattern architects
- Sympathetic users to test the assertions
- Open source code contributors
- Proof-tool builders, model checkers,…
- Teachers and students to do the work

Test “IMO Grand Challenge” for a Grand Challenge (Ref. slide 17)

• May take fifteen years or more to complete	Maybe
• Attracts world-wide participation	Maybe
• Has a clear test of success or failure	Yes!
• Pursues scientific ideals	Yes!
• Enlarges fundamental understanding:	
○ pursues scientific ideals	Yes!
○ in an area of significance to mankind	Maybe

What does “IMO Grand Challenge” needs? (Ref. slide 17)

• Maturity of the state of the art	Maybe
• Support from the scientific community	Maybe
• Commitment from the engaged teams	Depends on us (not US!)
• Understanding from funding agencies	Depends on us (not US!) and funding bodies (US maybe)

Can/Should we (IT, CS, SE, AI and MathEdu people) catch the train?

What can we try? – Catch the train maybe?

- From Daniel Selsam – personal communication (October 14, 2020):
  - *… I gave a talk on the IMO-GC a few weeks ago (https://www.youtube.com/watch?v=GtAo8wqWHHg) and do not have much new to report on top of that. The two pressing TODOs are*
    a) *formalize historical problems in mathlib, which largely means formalizing the definitions+lemmas necessary for short proofs to exist,*
    b) *and finish Lean4 and port mathlib to it.*

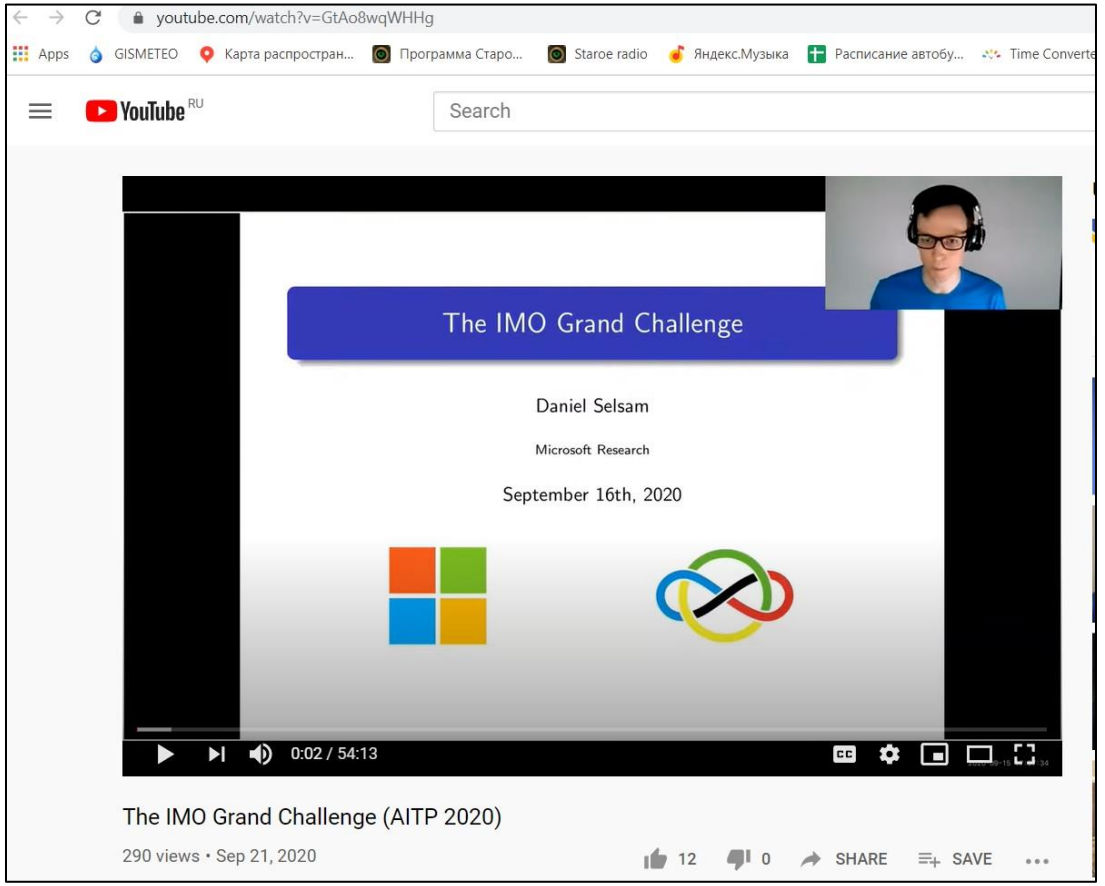
What can we try? – Catch the train maybe?

- Continued:
  - *While (b) is currently bottlenecked on the core Lean4 developers, (a) is naturally decentralized and decomposable.*
  - *Several people all over the world have already started formalizing problems (https://github.com/leanprover-community/mathlib/tree/master/archive/imo and https://github.com/leanprover-community/mathlib/labels/imo).*
  - *I think it will be hard to experiment with automation and ML productively until Lean4 is ready.*

If you would like – just watch!

- Daniel Selsam: *The IMO Grand Challenge*. Invited talk at 5th Conference on Artificial Intelligence and Theorem Proving (AITP 2020), September 13-19, 2020, Aussois, France

  (https://www.youtube.com/watch?v=GtAo8wqWHHg)



What *ELSE* we can try?

- (Ref. slide 9) They “propose the formal-to-formal (F2F) variant of the IMO: the AI receives a formal representation of the problem…”
  - Why not to try to extract a formal representation from a natural language problem description? (Maybe we should suggest a quasi-formal template for problems’ description in structured natural language?)

What *ELSE* we can try?

- (Ref. slide 9) “the AI receives a formal representation of the problem (in the Lean Theorem Prover), and is required to emit a formal (i.e. machine-checkable) proof.”
  - Do we have some experience with theorem provers for program verification? Why not to expand our experience and try automatic solution of mathematical contest problems?
  - We also have experience with some classes of problems (like recursion elimination and algorithm termination) that can help to solve automatically some mathematical contest problems.

What *ELSE* we can try?

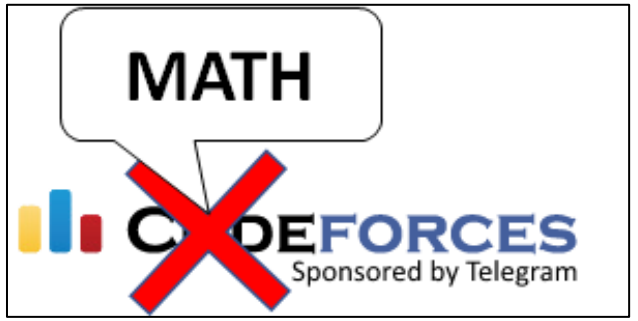
- Codeforces (https://codeforces.com/) is a website that hosts competitive programming contests. It is maintained by a group of competitive programmers from ITMO.

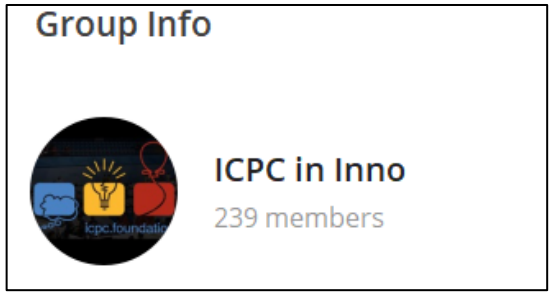
(https://en.wikipedia.org/wiki/Codeforces)

- Since 2013, Codeforces claims to surpass Topcoder in terms of active contestants. As of 2018, it has over 600,000 registered users. Codeforces along with other similar websites are used by top sport programmers like Gennady Korotkevich, Petr Mitrichev, Benjamin Qi and Makoto Soejima, and by other programmers interested in furthering their careers.

What *ELSE* we can try?

- There are no automatic tools (like Codeforces) for training and rating students for mathematical contests and Olympiad.
  - We have people experienced with programming contests and Codeforces technology.
  - Why not to try to design the first tool like Codeforces but for training and rating students for mathematical contests and Olympiad?





Engage students…

Автономная некоммерческая организация высшего образования
«Университет Иннополис»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)
по направлению подготовки
09.03.01 - «Информатика и вычислительная техника»

GRADUATION THESIS
(BACHELOR'S GRADUATION THESIS)
Field of Study
09.03.01 – «Computer Science»

Направленность (профиль) образовательной программы
«Информатика и вычислительная техника»
Area of Specialization / Academic Program Title:
«Computer Science»

Тема / Topic	**Формализация и решение задач Международной математической олимпиады на языке программирования Lean в рамках IMO Grand Challenge / Formalizing and solving International Math Olympiad problems using Lean programming language within IMO Grand Challenge**

Работу выполнил / Thesis is executed by	**Михайлов Руслан Николаевич / Ruslan Mikhailov**	подпись / signature
Руководитель выпускной квалификационной работы / Supervisor of Graduation Thesis	**Шилов Николай Вячеславович / Nikolay Shilov**	подпись / signature

3.3.1 Number theory

For the first example, here is a modified version of the short number theory problem, which goes by the number 4, from the Swedish Math Olympiad 2002 [57]. The formalization of this math problem demonstrates the fundamentals which are used in most math olympiad tasks which I manually processed.

Swedish Math Olympiad 2002 Q4

Prove that there is no such integer $n \geqslant 10$ such that $n^{\frac{1}{n-7}}$ is an integer

Table VI: Formalized Statements of Example Problem 1

Given	Goal
$n \in \mathbf{N}$ $n \geqslant 10$	$n^{\frac{1}{n-7}} \notin \mathbf{N}$

3.3.2 Algebraic expressions

For the second example, I decided to put a problem with the same algebraic type as the one in the previous example. The problem has also originated from the Malaysian math olympiad [58]. Hence, it was a real math problem that was given to the students:

Junior Olympiad of Malaysia 2015 Q3

Let a, b, and c be positive real numbers greater or equal to 3. Prove that: $3(abc + b + 2c) \geqslant 2(ab + 2ac + 3bc)$

Table VII: Formalized Statements of Example Problem 2

Given	Goal
$a, b, c \in \mathbf{R}$ $a \geqslant 3$ $b \geqslant 3$ $c \geqslant 3$	$3(abc + b + 2c) \geqslant 2(ab + 2ac + 3bc)$

Team must include … (Ref. slide 20)

- Programming theorists
- Programming tool-set builders
- Not Compiler writers **but NLP professionals**
- Design pattern architects **(Math contest and Edu experts)**
- Sympathetic users to test the assertions
- Open source code contributors
- Not Proof-tool builders,… **but proof-assistant users**
- Teachers and students to do the work

Collaborators? - Alexander Podkolzin

(http://intsys.msu.ru/staff/podkolzin/)

- Automatic math problem solver:
  - An intelligent system based on a new understanding of the functioning of intuition and decision-making is implemented in a scientific version for solving mathematical problems on a PC.
  - A special formalism for its description has been developed using a new high-level programming language. The system can be used in program verification, CHIP synthesis, etc.
  - One of the versions of the solver is protected by a series of US patents.

(http://intsys.msu.ru/invest/solver/)

Understanding from funding agencies (Ref. slide 22)

- Via Mathematical Centers maybe?

Why Not Try? – Let Us Try!