The Formalization of Mathematics

January 31, 2020
Thomas Hales, Pitt

to Peter Koepke

Logo of the Alfred P. Sloan Foundation,

Alfred P. Sloan
FOUNDATION

UNIVERSITY OF
PITTSBURGH
VERITAS 1787 VIRTUS

Carnegie
Mellon
University

THANG LONG
UNIVERSITY

VIỆN TOÁN HỌC

Abstract

A formal proof is a mathematical proof that has been checked by computer. The axioms and primitive rules of logic are programmed into a computer, and a proof is not regarded as verified until every step is exhaustively justified by first principles. Examples of proofs that have been formalized by various groups include the Kepler conjecture on sphere packings (2014), the independence of the Continuum Hypothesis (2019), and the Odd-order theorem in finite group theory (2012).

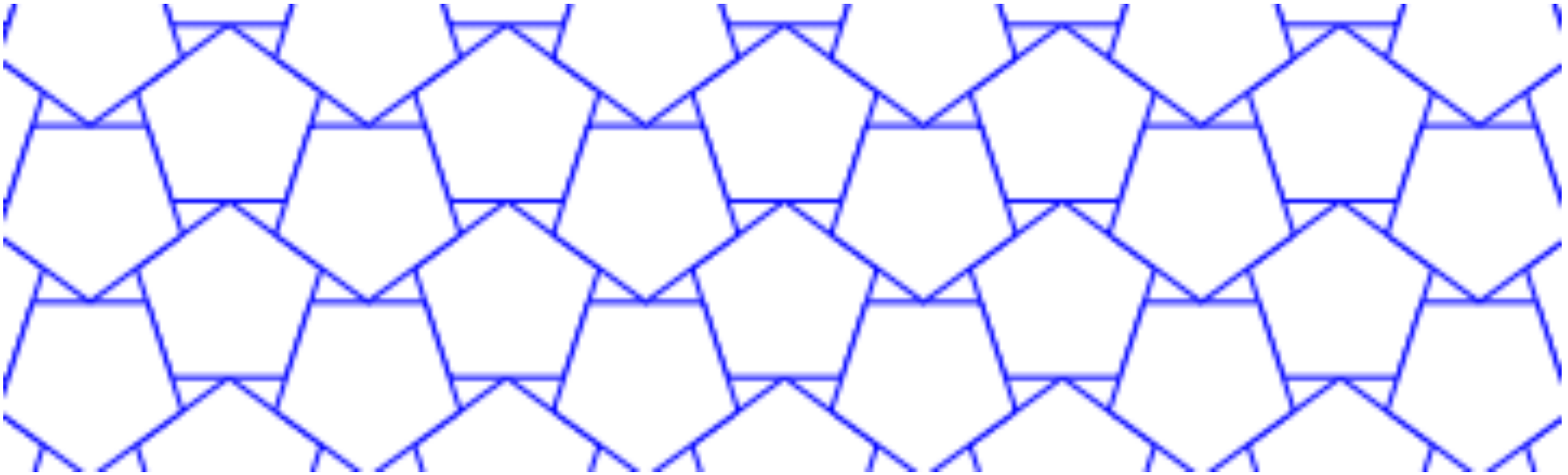
Buoyed up by these successful formalization projects, we are exploring how these tools might bring general benefit to the mathematical community.

Part I.

Computer-Assisted Proofs

- My motivation for formalization comes from computer assisted proofs (mostly in discrete geometry).

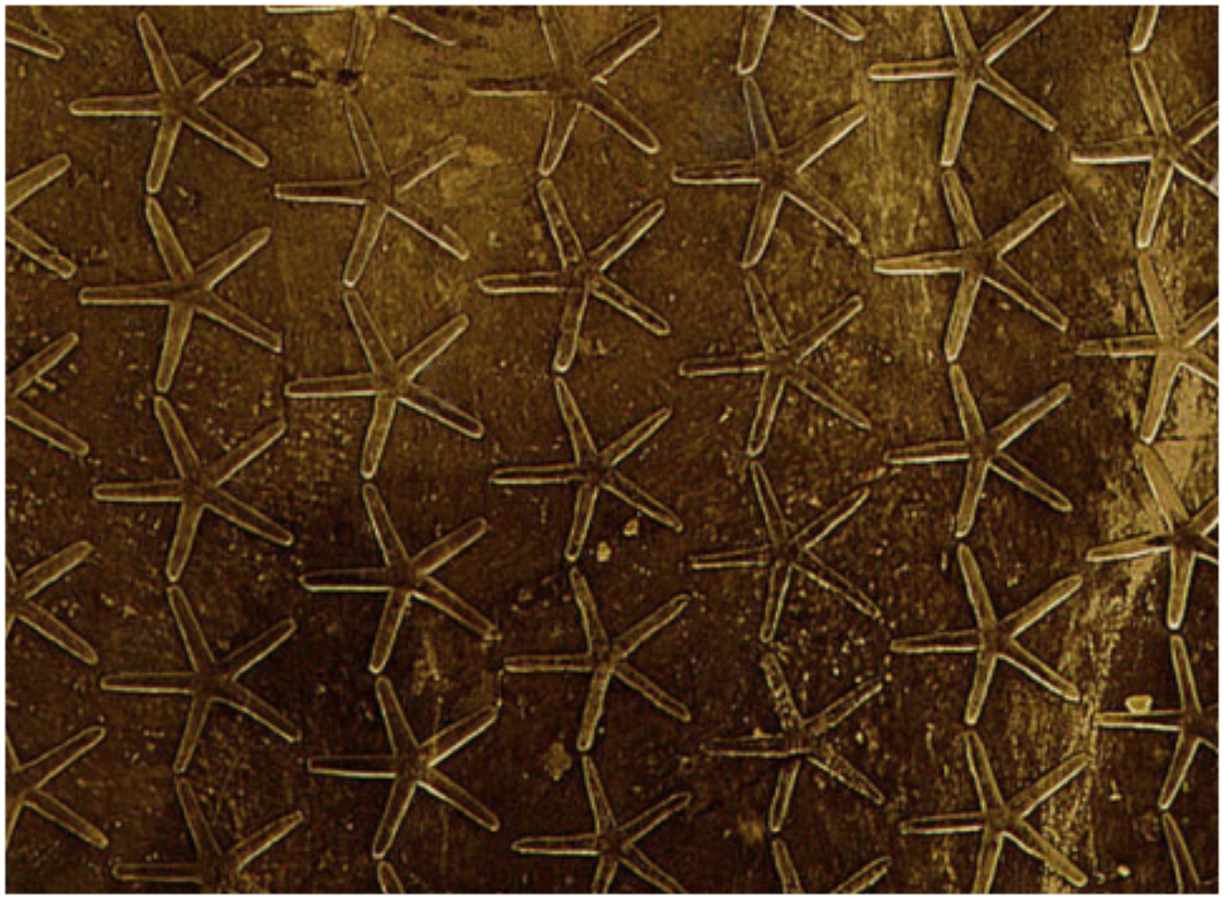
This is the densest packing of regular pentagons in the plane (Kusner-H.-, 2016).



It is a computer assisted proof.

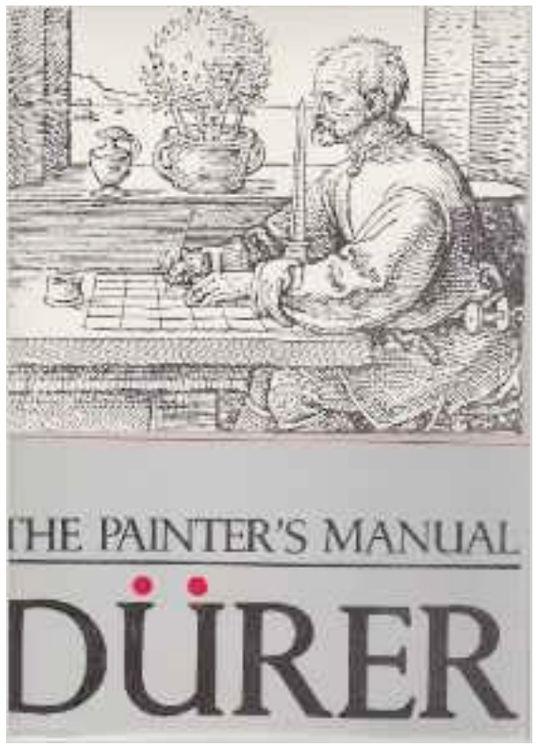
In the end, the solution of the pentagon-packing problem is 60 pages of text and 5000 lines of OCaml computer code that takes about 60 hours to run on a laptop computer. We use a wonderful interval arithmetic package to control for computer roundoff errors. The proof is computer-assisted, but is not formally verified in a proof assistant.

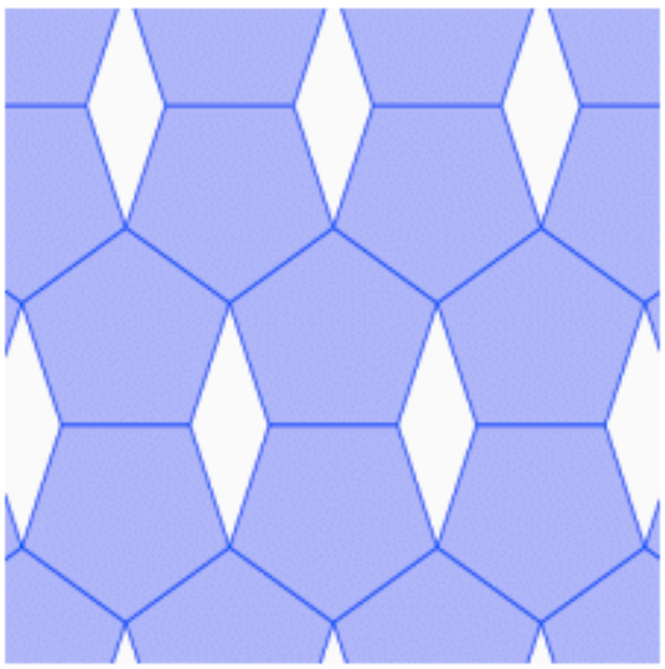
Why Pentagons?



Pyramid of Unas

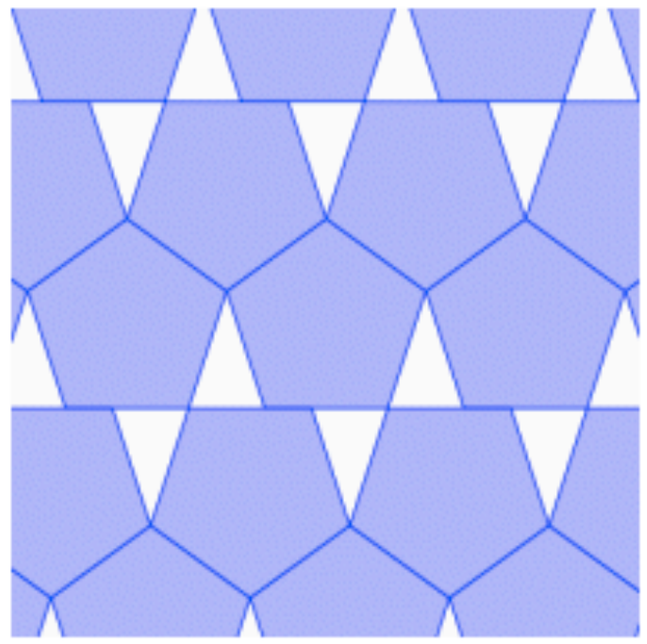
Patterns of five-pointed stars appear in the Egyptian pyramid of Unas (Fifth dynasty, 24th century BCE). With a little imagination, we can enclose each star in a pentagon to obtain a periodic packing of regular pentagons.



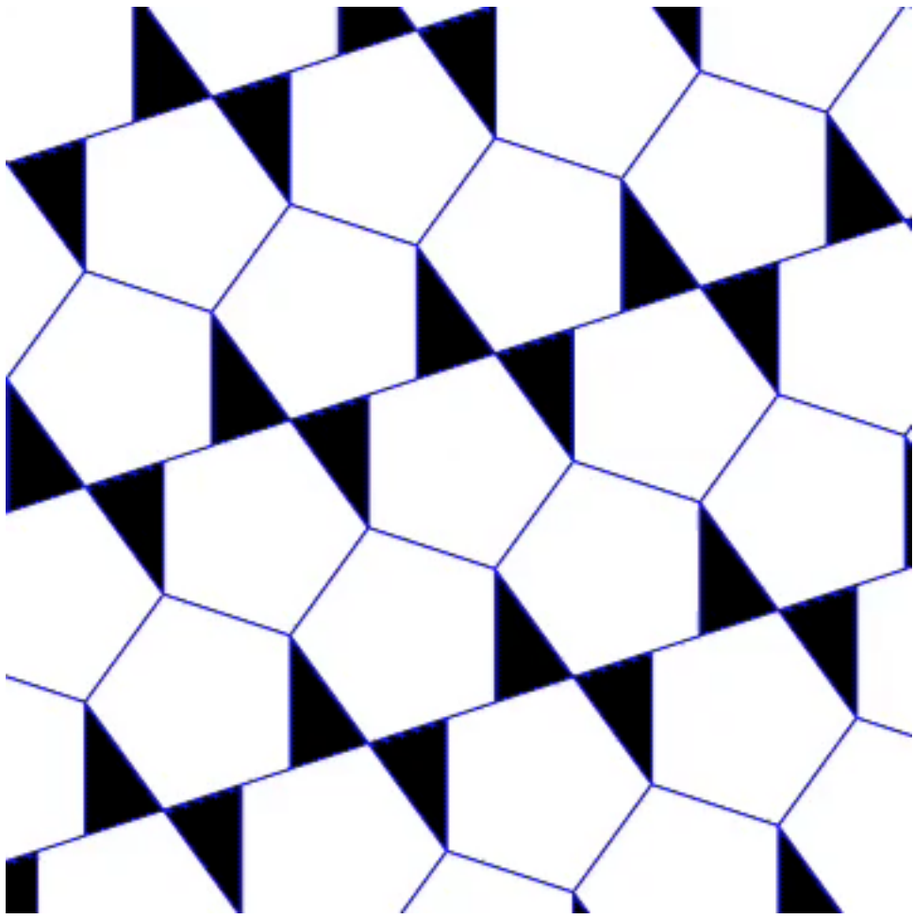


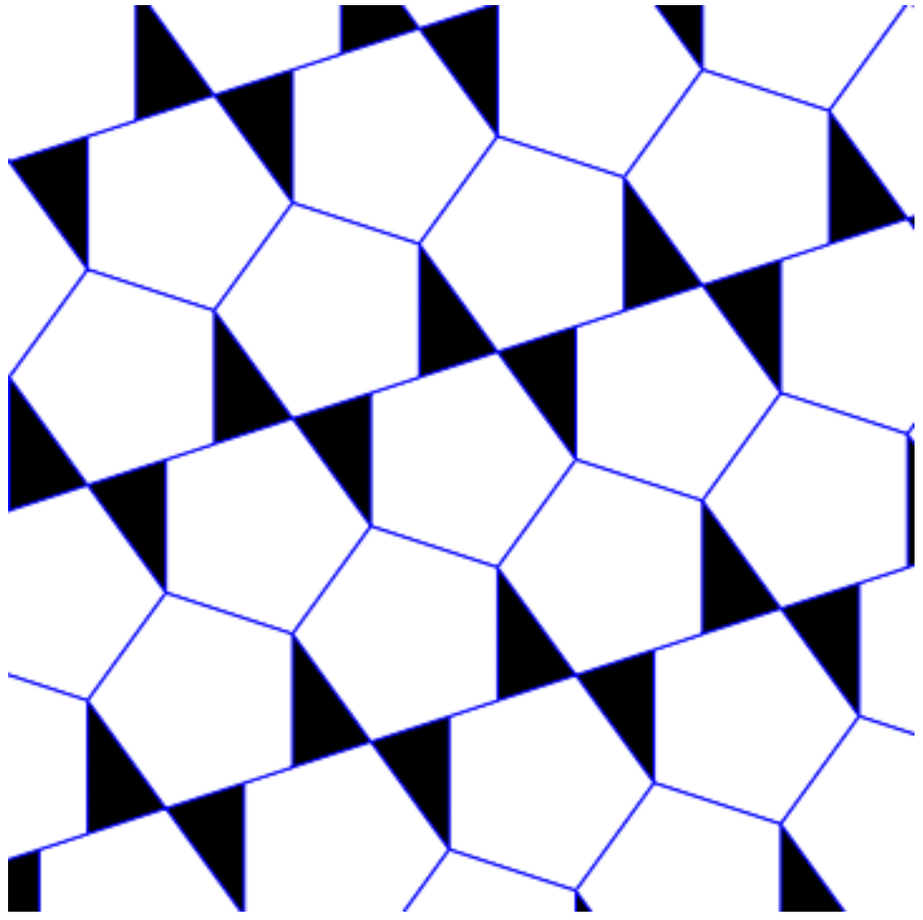
Dürer's packing

In 1525, Albrecht Dürer described some pentagon arrangements, including one that is relevant to our work. Variants of Dürer's packing can be obtained by translating its layers. Later, Kepler produced some pentagon arrangements, but they are of limited interest to us because of their low density.



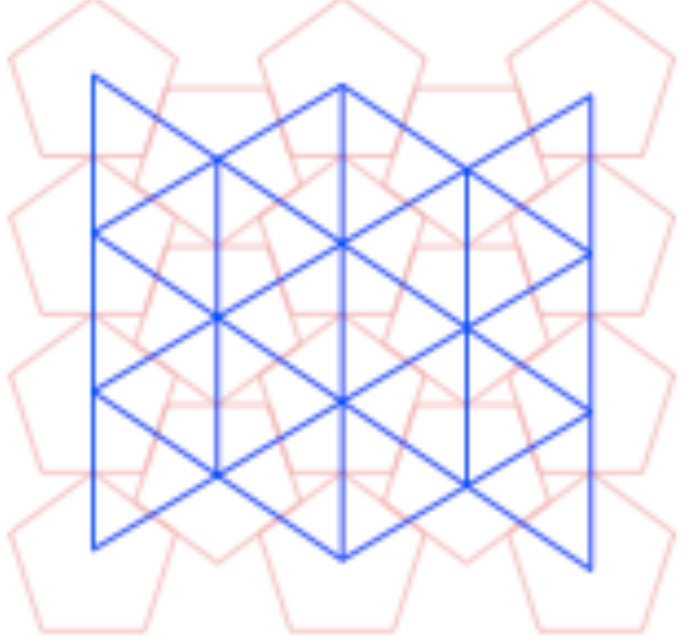
Shifted layers in Dürer's packing.

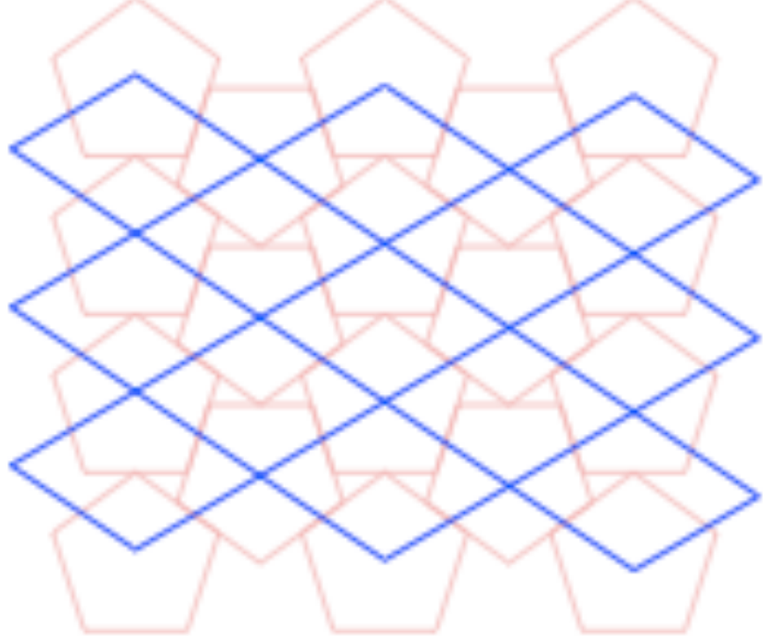




In 1990, Kuperberg and Kuperberg made a general study of high density packings of convex bodies in the plane. They showed that any convex body can be placed in a one-parameter family of double lattice packings and proved that this family always includes the densest of all double-lattice packings. The Kuperberg family of pentagon packings undulates between the shifting layers of Dürer's packing and the pentagonal ice-ray. Based on these results, they too made the pentagonal ice-ray conjecture.

This problem is easily fixed by pairing up Delaunay triangles along their longest edges. We call these pairs of Delaunay triangles ***dimers***. Wöden Kusner started serious calculations on pentagon packings in early 2013. His 2014 thesis contains a proof of the local optimality of the dimer in the pentagonal ice-ray among all pentagon packings.





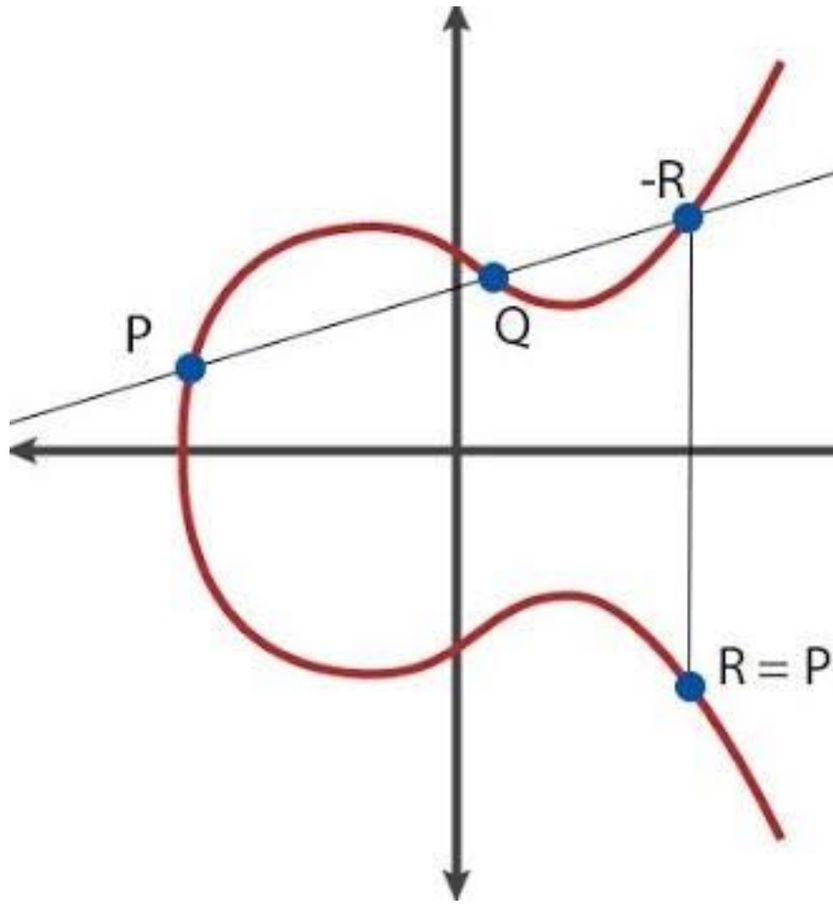
Elliptic Curve Cryptography

The Group Law for Edwards Curves

Thomas C. Hales

Abstract

This article gives an elementary computational proof of the group law for Edwards elliptic curves following Bernstein, Lange, et al., Edwards, and Friedl. The associative law is expressed as a polynomial identity over the integers that is directly checked by polynomial division. No preliminaries such as intersection numbers, Bézout's theorem, projective geometry, divisors, or Riemann Roch are required. The proofs have been designed to facilitate the formal verification of elliptic curve cryptography.



This article started with my frustration in teaching the elliptic curve group law in an undergraduate course in cryptography. I needed a simple

Lemma 2.1. *Let $A, B, C \in E \setminus \{O\}$. If $A \neq \pm B$, $B \neq \pm C$, $A + B \neq \pm C$ and $B + C \neq \pm A$, then*

$$(A + B) + C = A + (B + C).$$

Proof. Write $(x_1, y_1) := (A + B) + C$ and $(x_2, y_2) := A + (B + C)$. Let

$$\begin{aligned} \alpha &:= \frac{y_B - y_A}{x_B - x_A}, \quad \beta := \frac{y_A + y_C - \alpha(2x_A + x_B - \alpha^2)}{x_A + x_B + x_C - \alpha^2}, \\ \gamma &:= \frac{y_B - y_C}{x_B - x_C}, \quad \tau := \frac{y_A + y_B - \gamma(2x_B + x_C - \gamma^2)}{x_A + x_B + x_C - \gamma^2}. \end{aligned}$$

Using Equation (1) we get

$$\begin{aligned} x_1 &= \beta^2 + x_A + x_B - x_C - \alpha^2, \quad y_1 = -y_C + \beta(2x_C - x_A - x_B - \beta^2 + \alpha^2), \\ x_2 &= \tau^2 + x_B + x_C - x_A - \gamma^2, \quad y_2 = -y_A + \tau(2x_A - x_B - x_C - \tau^2 + \gamma^2). \end{aligned}$$

Setting

$$\begin{aligned} \widetilde{\alpha} &:= y_B - x_A, \quad \widetilde{\beta} := (y_A + y_C)(x_B - x_A)^3 - \widetilde{\alpha}((2x_A + x_B)(x_B - x_A)^2 - \widetilde{\alpha}^2), \\ \widetilde{\gamma} &:= y_B - y_C, \quad \widetilde{\tau} := (y_A + y_B)(x_B - x_C)^3 - \widetilde{\gamma}((2x_B + x_C)(x_B - x_C)^2 - \widetilde{\gamma}^2), \\ \widetilde{\eta} &:= x_B - x_A, \quad \widetilde{\mu} := x_B - x_C. \end{aligned}$$

one can show that $x_1 = x_2$ is equivalent to

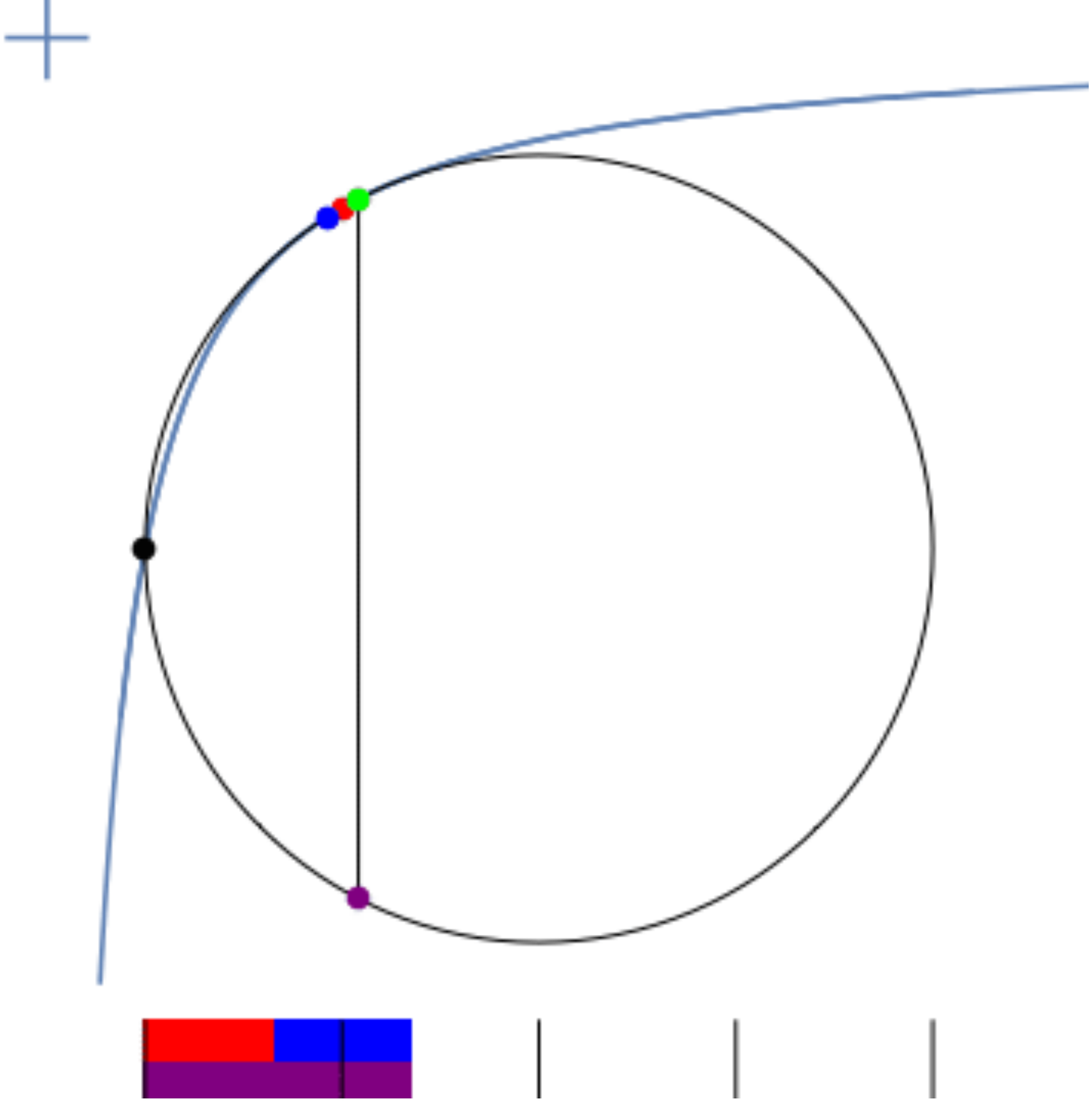
$$\begin{aligned} &(\widetilde{\beta}^2(x_B - x_C)^2 + (((2x_A - 2x_C)(x_B - x_C)^2 + \widetilde{\gamma}^2)(x_B - x_A)^2 - \widetilde{\alpha}^2(x_B - x_C)^2) \\ &((x_A + x_B + x_C)(x_B - x_A)^2 - \widetilde{\alpha}^2)^2)((x_A + x_B + x_C)(x_B - x_A)^2 - \widetilde{\gamma}^2)^2 \\ -\widetilde{\tau}^2&((x_A + x_B + x_C)(x_B - x_A)^2 - \widetilde{\alpha}^2)^2(x_B - x_A)^2 = 0 \end{aligned}$$

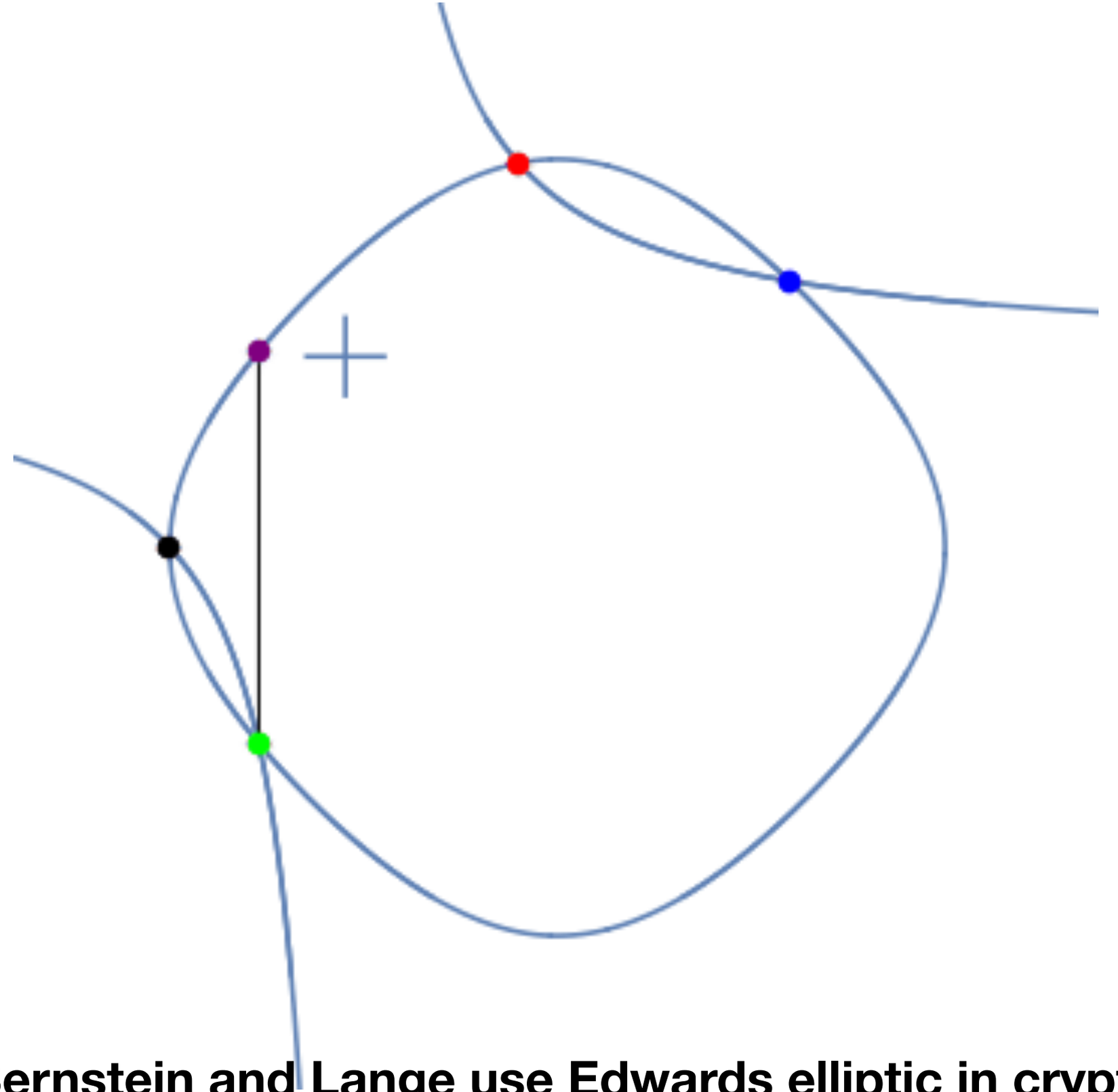
and $y_1 = y_2$ is equivalent to

$$\begin{aligned} &(y_A - y_C)((x_A + x_B + x_C)\widetilde{\eta}^2 - \widetilde{\alpha}^2)^3((x_A + x_B + x_C)\widetilde{\mu}^2 - \widetilde{\gamma}^2)^3\widetilde{\eta}^3\widetilde{\mu}^3 \\ +\widetilde{\beta}&(((2x_C - x_A - x_B)\widetilde{\eta}^2 + \widetilde{\alpha}^2)((x_A + x_B + x_C)\widetilde{\eta}^2 - \widetilde{\eta}^2)^2 - \widetilde{\beta}^2) \\ -\widetilde{\tau}&(\end{aligned}$$

Friedl's direct calculation of the associative law for elliptic curves

The computer calculations took "several hours" (in 1998).





Bernstein and Lange use Edwards elliptic in cryptography because they avoid timing-attacks.

The proof of associativity is given by a short Mathematica calculation. (I used Mathematica, because that is what I use, but other computer algebra systems should work equally well.) First, we write an explicit formula for the Edwards curve and take three points on the curve:

$$e[x\_, y\_] := x^2 + y^2 - 1 - d x^2 y^2; \quad e_1 = e[x_1, y_1]; \quad e_2 = e[x_2, y_2]; \quad e_3 = e[x_3, y_3].$$

Then we create an explicit formula for addition ($\oplus$) using our description with the hyperbola. The formula for addition is a rational function (ratio of polynomials) of $\{x_1, y_1\}$ and $\{x_2, y_2\}$ (using Mathematica's curly braces for ordered pairs). We test for failure of the associative law by setting

$$\{g_1, g_2\} = (\{x_1, y_1\} \oplus \{x_2, y_2\}) \oplus \{x_3, y_3\} - \{x_1, y_1\} \oplus (\{x_2, y_2\} \oplus \{x_3, y_3\})$$

We check associativity with the Mathematica command:

$$\text{PolynomialReduce}[\{g_1, g_2\}, \{e_1, e_2, e_3\}, \{x_1, y_1, x_2, y_2, x_3, y_3\}].$$

``I recently finished formalizing your paper on elliptic curves in Edwards form" - Rodrigo Raya 10/9/2019

From Rodrigo Raya <rodrigo.raya@epfl.ch>
Subject **Proof of elliptic Edwards curves**
To Me

Dear Prof. Hales,

I recently finished formalizing your paper on elliptic curves in Edwards form. You can see the formalization here:
https://nam05.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Frjraya%2FIsabelle%2Fblob%2Fmaster%2F
data=02%7C01%7Chales%40pitt.edu%7C84f9434c02724918e96508d74cd83c07%7C9ef9f489e0a04eeb87cc3a526112fd0d%7C1%7C1%7C637
sdata=PY38Mi7R2GPeV6dyPkOigMwc4q1iwwc%2FfjAlflDngj8%3D&reserved=0

I assume that it needs some rewriting to be in a more polished form but the argument is there.

I really appreciated the time you invested in this task, which was fundamental to complete it.

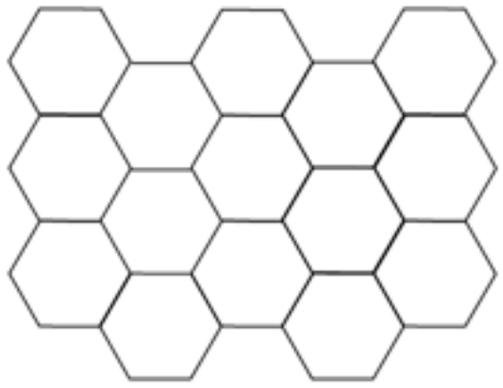
Best regards,

Rodrigo Raya

THE HONEYCOMB CONJECTURE

THOMAS C. HALES

ABSTRACT. This article gives a proof of the classical honeycomb conjecture: any partition of the plane into regions of equal area has perimeter at least that of the regular hexagonal honeycomb tiling.



1. INTRODUCTION

Around 36 B.C., Marcus Terentius Varro, in his book on agriculture, wrote about the hexagonal form of the bee's honeycomb. There were two competing theories of the hexagonal structure. One theory held that the hexagons better accommodated the bee's six feet. The other theory, supported by the mathematicians of the day, was that the structure was explained by an isoperimetric property of the hexagonal honeycomb. Varro wrote, "Does not the chamber in the comb have six angles ... The geometricians prove that this hexagon inscribed in a circular figure encloses the greatest amount of space."

The origin of this problem is somewhat obscure. Varro was aware of it long before Pappus of Alexandria, who mentions it in his fifth book. Much of Book V follows

Theorem 1-A (Honeycomb conjecture). *Let Γ be a locally finite graph in $\mathbb{R}^2$, consisting of smooth curves, and such that $\mathbb{R}^2 \setminus \Gamma$ has infinitely many bounded connected components, all of unit area. Let C be the union of these bounded components. Then*

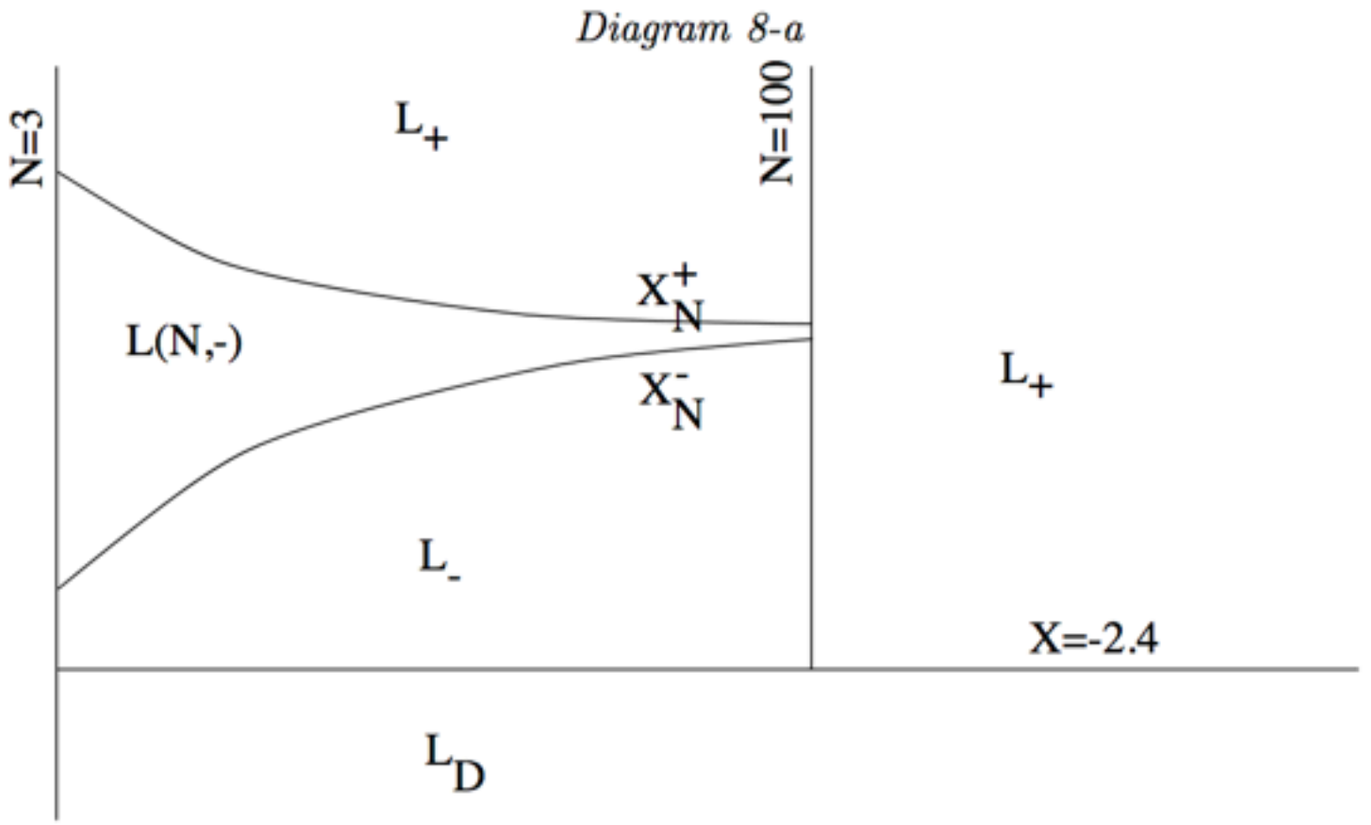
$$\limsup_{r \mapsto \infty} \frac{\text{perim}(C \cap B(0,r))}{\text{area}(C \cap B(0,r))} \geq \sqrt[4]{12}.$$

Equality is attained for the regular hexagonal tile.

Theorem 1-B (Honeycomb conjecture for disconnected regions). *Let K be a compact set in the plane containing disjoint measurable sets $R_1, R_2, \ldots$. Assume that each R_i has a rectifiable current boundary ∂R_i. Set $\alpha_i = \min(1, \text{area}(R_i))$. Set $\Gamma = \cup_i \partial R_i$. Assume $\alpha_i > 0$ for some i. Then*

$$\mathcal{H}^1(\Gamma) > \sqrt[4]{12} \sum \alpha_i.$$

Diagram 8-a



The boundary between $L_+(1)$ and $L(N,\cdot)$ is the curve $X_N^+ = 1 - \pi/\pi_N > 0$, determined by the condition $L_+(1) = L(N, 1, X_N^+)$. (For $N = 3$, we set $X_3^+ = 0.177$,

Other computer proofs

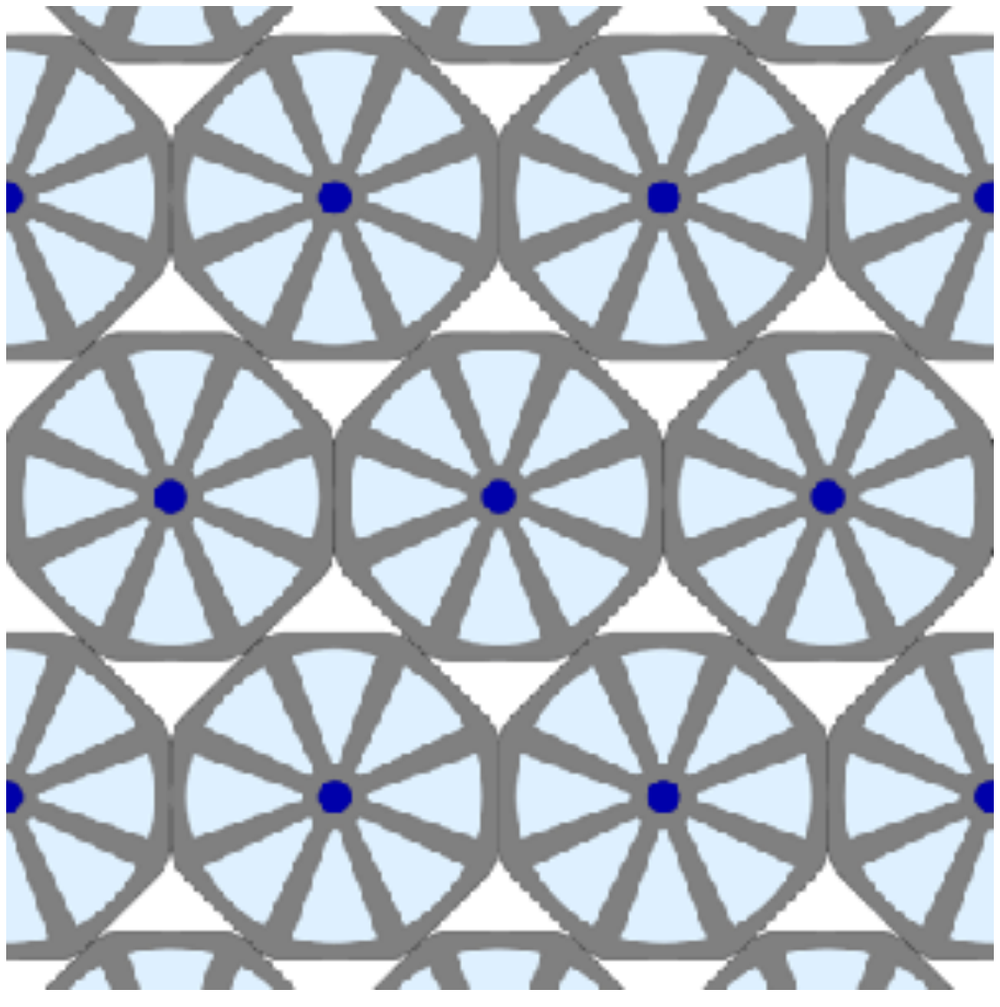
- Fejes-Toth conjecture. In a sphere packing (of congruent balls), if every sphere touches exactly 12 other spheres, then it consists of hexagonal layers.

- Strong dodecahedral conjecture. Every Voronoi cell of a sphere packing (of congruent balls) has volume at least that of a regular dodecahedron (circumscribing a ball).

- Sphere packing problem. No packing of congruent balls has density greater than the cannonball packing.

- Reinhardt conjecture local optimality. The smoothed octagon is a locally optimal for worst-best packing of a centrally symmetric convex disk in the plane.

Reinhardt Conjecture



(Graphics by Egan)

Birch-Swinnerton Dyer conjecture, Sato-Tate Conjecture, Lyons-Sims group of order $2^8 3^7 5^6 7^1 11^1 31^1 37^1 67^1$, original proof of the Calalan conjecture $x^m - x^n = 1$, qWZ proof of the Rogers-Ramunujan identities, conjectural optimal packings of tetrahedra (Chen-Engel-Glotzer), 4-color theorem, finite projective plane of order 10, Smale's 14th problem on strange attractors in the Lorenz oscillator, Mandelbrot's conjectures in fractal geometry, visualization of sphere eversions and Costa surface embeddings, the double bubble conjecture, construction of counterexamples to the Kelvin conjecture, calculation of kissing numbers (Sloane-Odlyzko), the character table for E^8 (Atlas project), Cohn-Kumar proof of the packing optimality of the Leech and E^8 packings among lattices, Viazovska's proof of the optimality of E^8 sphere packing, classification of fake projective planes, weak Goldbach, twin prime problem, Rao's classification of convex pentagon tiles, Boolean pythagorean triples problem, chromatic number of the plane, Dirac-Schwinger conjecture (Fefferman-Seco).

Why should we doubt computer proofs?

- Software has bugs.

- The computer code is generally not refereed as part of the review process. Math journals generally have no standards for code review.

- The computer code is generally not published. Often it cannot be found anywhere. Proofs might rely on proprietary software.

- Mathematicians generally do not read and check computer code.

ANNALS OF MATHEMATICS
Princeton University & Institute for Advanced Study

Statement by the Editors on Computer-Assisted Proofs

Computer-assisted proofs of exceptionally important mathematical theorems will be considered by the Annals.

The human part of the proof, which reduces the original mathematical problem to one tractable by the computer, will be refereed for correctness in the traditional manner. The computer part may not be checked line-by-line, but will be examined for the methods by which the authors have eliminated or minimized possible sources of error: (e.g., round-off error eliminated by interval arithmetic, programming error minimized by transparent surveyable code and consistency checks, computer error minimized by redundant calculations, etc. [Surveyable means that an interested person can readily check that the code is essentially operating as claimed]).

We will print the human part of the paper in an issue of the Annals. The authors will provide the computer code, documentation necessary to understand it, and the computer output, all of which will be maintained on the Annals of Mathematics website online.

Part 2

Formal Proofs

… A formal proof is a mathematical proof that has been checked by computer. The axioms and primitive rules of logic are programmed into a computer, and a proof is not regarded as verified until every step is exhaustively justified by first principles. …

HOL Light

HOL Light has an exquisite minimal design. It has the smallest kernel of any system.

Lean

Lean is ambitious,
and it will be massive.

Lean Theorem Prover

- Lean has a small kernel.
- Its logical foundations are similar to those of Coq.
- Lean is its own metalanguage.

Mathematics > Metric Geometry

A formal proof of the Kepler conjecture

Thomas Hales, Mark Adams, Gertrud Bauer, Dat Tat Dang, John Harrison, Truong Le Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Thang Tat Nguyen, Truong Quang Nguyen, Tobias Nipkow, Steven Obua, Joseph Pleso, Jason Rute, Alexey Solovyev, An Hoai Thi Ta, Trung Nam Tran, Diep Thi Trieu, Josef Urban, Ky Khac Vu, Roland Zumkeller

(Submitted on 9 Jan 2015)

This article describes a formal proof of the Kepler conjecture on dense sphere packings in a combination of the HOL Light and Isabelle proof assistants. This paper constitutes the official published account of the now completed Flyspeck project.

Why formalize mathematics?

- Computer Proofs
- Mathematicians do computer proofs.
- Computer proofs are hard to check.
- Mathematicians do not want to compromise standards in going from paper to computer.

Other reasons for formalization

- Some mathematicians have discovered errors in their own work, and do not want it to happen again. (Voevodsky)

- Some believe that all mathematics will be formalized mathematics in the future, and do not want their work to be forgotten. (Grayson)

- Some believe that the published record leaves out too many details, and that future generations will have trouble reconstructing informal proofs from some branches of mathematics. (Buzzard)

- Proofs are becoming longer and more complex. Better tools are required for long complex proofs (Cambridge Big Proofs I 2017, Edinburgh Big Proofs II 2019)

- Some mathematical techniques (such as zero-knowledge proofs) take a formal proof as input.

- Machine-learning projects aimed at learning to construct computer proofs use formal proofs as training data (Urban, Szegedy)

- Processing of mathematics works better on formal content (search, indexing, transformation).

- Some day referees might be replaced by computers (H.)

Formalizing Mathematics

- Formal Verification has been a research topic in computer science for decades.

- Increasingly mathematicians are becoming involved. Especially, in the past two years there has been a surge in interest among mathematicians.

Year	Theorem	Proof System	Formalizer	Traditional Proof
1986	First Incompleteness	Boyer-Moore	Shankar	Gödel
1990	Quadratic Reciprocity	Boyer-Moore	Russinoff	Eisenstein
1996	Fundamental - of Calculus	HOL Light	Harrison	Henstock
2000	Fundamental - of Algebra	Mizar	Milewski	Brynski
2000	Fundamental - of Algebra	Coq	Geuvers et al.	Kneser
2004	Four Color	Coq	Gonthier	Robertson et al.
2004	Prime Number	Isabelle	Avigad et al.	Selberg-Erdös
2005	Jordan Curve	HOL Light	Hales	Thomassen
2005	Brouwer Fixed Point	HOL Light	Harrison	Kuhn
2006	Flyspeck I	Isabelle	Bauer-Nipkow	Hales
2007	Cauchy Residue	HOL Light	Harrison	classical
2008	Prime Number	HOL Light	Harrison	analytic proof
2012	Odd Order Theorem	Coq	Gonthier	Feit-Thompson

A Formal Proof of the Independence of the Continuum Hypothesis

Jesse Michael Han
Department of Mathematics
University of Pittsburgh
Pittsburgh, PA, USA
jessemichaelhan@gmail.com

Floris van Doorn
Department of Mathematics
University of Pittsburgh
Pittsburgh, PA, USA
fpvdoorn@gmail.com

Abstract

We describe a formal proof of the independence of the continuum hypothesis (CH) in the Lean theorem prover. We use Boolean-valued models to give forcing arguments for both directions, using Cohen forcing for the consistency of ¬CH and a σ-closed forcing for the consistency of CH.

Keywords continuum hypothesis, forcing, Lean, set theory, ZFC

1 Introduction

The continuum hypothesis (CH) states that there is no cardinality between ω, the smallest infinite cardinal and $\mathfrak{c}$, the cardinality of the continuum. It was introduced by Cantor [6] in 1878 and was the first problem on Hilbert's list of twenty-three outstanding problems in mathematics. Gödel [14] proved in 1938 that CH was consistent with ZFC, and later conjectured that CH is independent of ZFC, i.e. neither provable nor disprovable from the ZFC axioms. In 1963, Paul Cohen developed *forcing* [8, 9], which allowed him to prove

Our formalization[2] uses the Lean 3 theorem prover, building on top of mathlib [29]. Lean is an interactive proof assistant under active development at Microsoft Research [10, 44]. It implements the Calculus of Inductive Constructions and has a similar metatheory to Coq, adding definitional proof irrelevance, quotient types, and a noncomputable choice principle. Our formalization makes as much use of the expressiveness of Lean's dependent type theory as possible, using constructions which are impossible or unwieldy to encode in HOL, let alone ZF. The types of cardinals and ordinals in mathlib, which are defined as equivalence classes of (well-ordered) types, live one universe level higher than the types used to construct them, and our models of set theory require as input an entire universe of types. Our encoding of first-order logic also uses parameterized inductive types which ensure that type-correctness implies well-formedness, eliminating the need for separate well-formedness proofs.

The method of forcing with Boolean-valued models was developed by Solovay and Scott [38, 40] as a simplification of Cohen's method. Some of these simplifications were incorpo-

A concrete proposal: mathematical FABSTRACTS (formal abstracts)

Given today's technology, it is not reasonable to ask for all proofs to be formalized. But with today's technology, it seems that it should be possible to create a formal abstract service that

- Gives a statement of the main theorem(s) of each published mathematical paper in a language that is both human and machine readable,
- Links each term in theorem statements to a precise definition of that term (again in human/machine readable form), and
- Grounds every statement and definition is the system in some foundational system for doing mathematics.

The definitions of mathematics

The Oxford English dictionary (2nd edition) has 273,000 headwords and over 600,000 word forms. (The longest entry is for the word set, which continues for 25 pages).

Medicine has a specialized terminology of approximately 250,000 items [Kucharz].

The Math Subject Classification (MSC) lists over 6000 subfields of mathematics.

of Mathematics

Main page | Discussion

Main Page

The Encyclopedia of Mathematics wiki is an open access resource designed specifically for the mathematics community. The
Mathematics, published by Kluwer Academic Publishers in 2002. With more than 8,000 entries, illuminating nearly 50,000 noti
was the most up-to-date graduate-level reference work in the field of mathematics.

Sylvester, "On a theory of Syzygetic Relations"

allotrious, apocapated, Bezoutic, Bezoutoid, co-bezoutiant, cogredient, contragredient, combinant, concomitant, conjunctive, contravariant, covariant, cumulant, determinant, dialytic, discriminant, disjunctive, effluent, emanant, endoscopic, exoscopic, Hessian, hyperdeterminant, inertia, intercalation, invariance, invariant, Jacobian, kenotheme, matrix, minor determinant, monotheme, persymmetrical, quadrinvariant, resultant, rhizoristic, signaletic, semaphoretic, substitution, syrrhizoristic, syzygetic, transform, umbral.

VOCABULARY OF THE KEPLER CONJECTURE

- quoin, negligible, fcc-compatible, decomposition star, score, score adjustment, quasi-regular tetrahedron, contravening, tame graph, pentahedral prism, crown, quarter, upright, flat, quartered octahedron, strict quarter, enclosed vertex, central vertex, corners, isolated quarter, isolated pair, conflicting diagonals, Q-system, S-system, V-cells, barrier, obstructed, face with negative orientation, Delaunay star, colored spaces, compression, quad cluster, mixed quad cluster, standard cluster, standard region, vertex type, quad cluster, Rogers simplex, anchor, anchored simplex, erasing, loops, subcluster, corner cell, truncated corner cell , tame graph, weight assignment, contravening circuit, crowded diagonal, n-crowded, masked, confined, penalties, penalty-free score, exceptional region, special simplex, distinguished edge, nonexternal edge, concave corner, concave vertex, t-cone, partial plane graph, patch, aggregated face,

Lemma 2.73 (Euler triangle) [JLPSDHF] *Let* $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ *be points in* $\mathbb{R}^3$. *Let*

$$(y_1, \ldots, y_6) = (y_{01}, y_{02}, y_{03}, y_{23}, y_{13}, y_{12}), \text{ where } y_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|.$$

Set $x_i = y_i^2$. *and*

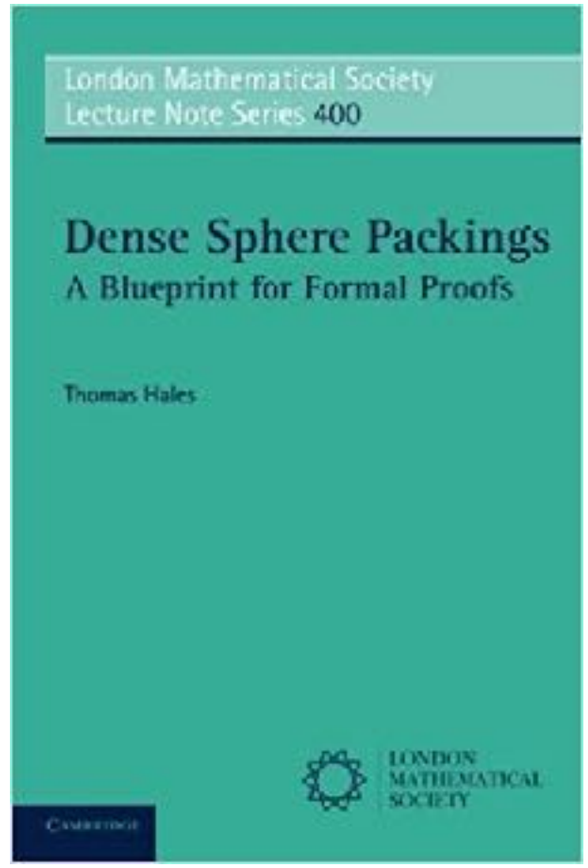
$$p = y_1y_2y_3 + y_1(\mathbf{w}_2 \cdot \mathbf{w}_3) + y_2(\mathbf{w}_1 \cdot \mathbf{w}_3) + y_3(\mathbf{w}_1 \cdot \mathbf{w}_2).$$

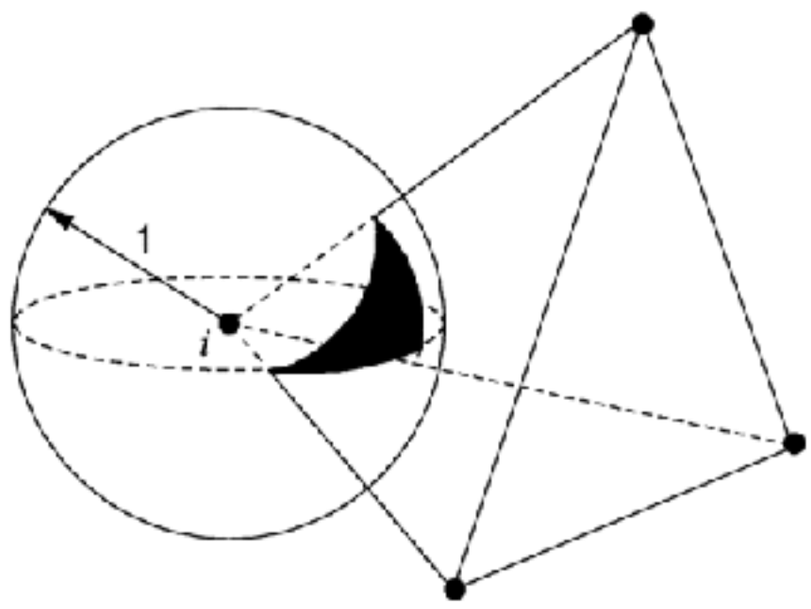
where $\mathbf{w}_i = \mathbf{v}_i - \mathbf{v}_0$. *Let*

$$\alpha_i = \mathrm{dih}_V(\{\mathbf{v}_0, \mathbf{v}_i\}, \{\mathbf{v}_j, \mathbf{v}_k\})$$

where $\{i, j, k\} = \{1, 2, 3\}$. *Assume that* $\Delta(x_1, \ldots, x_6) > 0$. *Then*

$$\alpha_1 + \alpha_2 + \alpha_3 - \pi = \pi - 2\arctan_2(\Delta(x_1, \ldots, x_6)^{1/2}, 2p).$$





```
let JLPSDHF = Euler_main_theorem.EULER_TRIANGLE;; (* euler_triangle ;; *)
```

```
let euler_triangle_t = `!v0 v1 v2 v3:real^3.
    let p = euler_p v0 v1 v2 v3 in
    let (x1,x2,x3,x4,x5,x6) = xlist v0 v1 v2 v3 in
    let alpha1 = dihV v0 v1 v2 v3 in
    let alpha2 = dihV v0 v2 v3 v1 in
    let alpha3 = dihV v0 v3 v1 v2 in
    let d = delta_x x1 x2 x3 x4 x5 x6 in
    ((&0 < d) ==>
      (alpha1 + alpha2 + alpha3 - pi = pi - &2 * atn2(sqrt(d), (&2 * p))))`;;
```

```
let EULER_TRIANGLE = prove_by_refinement (euler_triangle_t ,
```

Lemma 2.73 (Euler triangle) [JLPSDHF] *Let* $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ *be points in* $\mathbb{R}^3$. *Let*

$$(y_1, \ldots, y_6) = (y_{01}, y_{02}, y_{03}, y_{23}, y_{13}, y_{12}), \text{ where } y_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|.$$

Set $x_i = y_i^2$. *and*

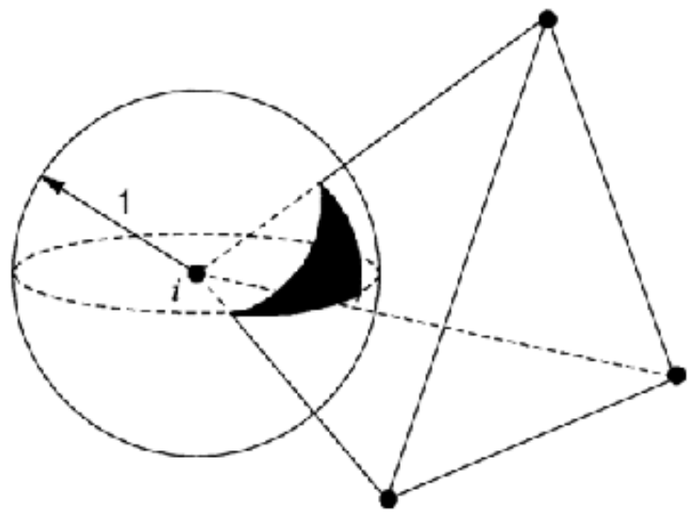
$$p = y_1y_2y_3 + y_1(\mathbf{w}_2 \cdot \mathbf{w}_3) + y_2(\mathbf{w}_1 \cdot \mathbf{w}_3) + y_3(\mathbf{w}_1 \cdot \mathbf{w}_2).$$

where $\mathbf{w}_i = \mathbf{v}_i - \mathbf{v}_0$. *Let*

$$\alpha_i = \mathrm{dih}_V(\{\mathbf{v}_0, \mathbf{v}_i\}, \{\mathbf{v}_j, \mathbf{v}_k\})$$

where $\{i, j, k\} = \{1, 2, 3\}$. *Assume that* $\Delta(x_1, \ldots, x_6) > 0$. *Then*

$$\alpha_1 + \alpha_2 + \alpha_3 - \pi = \pi - 2\arctan_2(\Delta(x_1, \ldots, x_6)^{1/2}, 2p).$$



Don't repeat yourself

From Wikipedia, the free encyclopedia

Don't repeat yourself (DRY, or sometimes **do not repeat yourself)** is a principle of software development aimed at reducing repetition of software patterns,[1] replacing it with abstractions or using data normalization to avoid redundancy.

The DRY principle is stated as "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system". The principle has been formulated by Andy Hunt and Dave Thomas in their book *The Pragmatic Programmer*.[2] They apply it quite broadly to include "database schemas, test plans, the build system, even

DRY vs WET solutions [edit]

Violations of DRY are typically referred to as WET solutions, which is commonly taken to stand for "write every time", "write everything twice", "we enjoy typing" or "waste everyone's time". WET solutions are common in multi-tiered architectures

A concrete proposal: mathematical FABSTRACTS (formal abstracts)

Given today's technology, it is not reasonable to ask for all proofs to be formalized. But with today's technology, it seems that it should be possible to create a formal abstract service that

- Gives a statement of the main theorem(s) of each published mathematical paper in a language that is both human and machine readable,
- Links each term in theorem statements to a precise definition of that term (again in human/machine readable form), and
- Grounds every statement and definition is the system in some foundational system for doing mathematics.

Jordan Curve Theorem

```
‘∀C . simple_closed_curve top2 C ⇒
  (∃A B . top2 A ∧ top2 B ∧
    connected top2 A ∧ connected top2 B ∧
    (A ≠ ∅) ∧ (B ≠ ∅) ∧
    (A ∩ B = ∅) ∧ (A ∩ C = ∅) ∧ (B ∩ C = ∅) ∧
    (A ∪ B ∪ C = euclid 2))‘
```

A rather literal translation of this HOL Light code into English is as follows: [Here, *top2* is the standard metric space topology on $\mathbb{R}^2$.] Let C be a simple closed curve, with respect to *top2*. Then there exist sets A and B with the following properties: A and B are open in the topology *top2*; A and B are connected with respect to the topology *top2*; A and B are nonempty; the sets A, B, and C are pairwise disjoint; and the union of A, B, and C is $\mathbb{R}^2$. Or more idiomatically, a Jordan curve C partitions the plane into the three sets A, B, and C itself, where A and B are nonempty, connected, and open.

Recent and Current Projects

Controlled Natural Language (CNL)

- It is based on a single natural language (such as English).
- It has restricted syntax and semantics. Its design is deliberate and explicit.
- Speakers of the natural language can largely understand the controlled language at least intuitively. (see Tobias Kuhn)
- The definition is intended to exclude artificial languages such as Esperanto and programming languages.

The argument for a controlled natural language for mathematics

(1) Technology is still far from being able to make a semantic reading of mathematics as it is currently written.
  (a) Machine learning techniques (in particular, deep neural networks) are still far from a semantic reading of mathematics.
  (b) Linguistic approaches are still far from a semantic reading of mathematics as it is currently written.
(2) Mathematicians are still far from the mass adoption of proof assistants.
  (a) Adoption has been gradual.
  (b) Structural reasons hinder the adoption of proof assistants.
(3) There is value in bridging the gap between (1) and (2).
(4) CNL technology works now and can help to bridge the gap.

Examples of CNLs for Mathematics

Recent and Current Projects

- Naproche-SAD (and variants Forthel, Naproche, EA,…). (Paskevich, 2007) (Koepke, Cramer, Frerix, 2018) The target is first-order logic.

- MathNat (and variants CLM controlled language of mathematics). (Humayoun’s thesis) The target is first-order logic.

- FMathL (formal mathematical language, CONCISE). The target is a graphical representation (sems).

Recent and Current Projects
Peter Koepke's example

Theorem 1 (text in Naproche-SAD system)**.** *If* $x \in \mathbb{R}$ *and* $y \in \mathbb{R}$ *and* $x > 0$ *then there is a positive integer* n *such that* $n \cdot x > y$.

Proof. Define $A = \{n \cdot x \mid n \text{ is a positive integer}\}$. Assume the contrary. Then y is an upper bound of A. Take a least upper bound α of A. $\alpha - x < \alpha$ and $\alpha - x$ is not an upper bound of A. Take an element z of A such that not $z \leq \alpha - x$. Take a positive integer m such that $z = m \cdot x$. Then $\alpha - x < m \cdot x$ (by 15b).

$$\alpha = (\alpha - x) + x < (m \cdot x) + x = (m + 1) \cdot x.$$

$(m + 1) \cdot x$ is an element of A. Contradiction. Indeed α is an upper bound of A. $\square$

Adapting CNLs to Type Theory

The basic idea of the controlled natural language

- Take all the syntax of Lean.
- Take all the syntax of TEX.
- Take all the syntax of ForTheL CNL.
- Throw it all together and identify all the common parts.
- Translate to Lean by expanding TEXand CNL (remove syntactic sugar).

CNL as PDF

Definition 6.11 (greatest element)**.** *We say that y is a* **greatest element** *in R iff for all x, $x \leq y$.*

We introduce synonyms least/minimum/bottom.

Definition 6.12 (least element)**.** *We say that y is a* **least element** *in R iff for all x, $y \leq x$.*

Let $x < y$ stand for $x \leq y$ and $x \neq y$.

Definition 6.13 (maximal element)**.** *We say that y is a* **maximal element** *in R iff there exists no x such that $y < x$.*

Definition 6.14 (minimal element)**.** *We say that y is a* **minimal element** *in R iff there exists no x such that $x < y$.*

Definition 6.15 (irreflexive)**.** *We say that R is* **irreflexive** *iff there exists no x such that $x < x$.*

Definition 6.16 (asymmetric)**.** *We say that R is* **asymmetric** *iff for all x y, $x < y$ implies not $y < x$.*

CNL as TeX

```
\deflabel{greatest element} We say that $y$ is a
\df{greatest\~element} in $R$ iff for all\ $x,\ x \le y$.
\end{definition}

We introduce synonyms least/minimum/bottom.

\deflabel{least element} We say that $y$ is a \df{least\~element} in
$R$ iff for all\ $x,\ y \le x$.
\end{definition}

Let $x < y$ stand for $x \le y$ and $x \ne y$.

\deflabel{maximal element} We say that $y$ is a \df{maximal\~element}
in $R$ iff there exists no $x$ such that $y < x$.
\end{definition}

\deflabel{minimal element} We say that $y$ is a \df{minimal\~element}
in $R$ iff there exists no $x$ such that $x < y$.
\end{definition}
```

CNL as converted TeX

```
Definition Label_greatest_element . We say that y is a
greatestelement in R iff for all x , x \le y .


We introduce synonyms least / minimum / bottom .

Definition Label_least_element . We say that y is a leastelement in
  R iff for all x , y \le x .
```

```
Let x < y stand for x \le y and x \ne y .

Definition Label_maximal_element . We say that y is a maximalelement
in R iff there exists no x such that y < x .


Definition Label_minimal_element . We say that y is a minimalelement
in R iff there exists no x such that x < y .


Definition Label_irreflexive . We say that R is irreflexive iff there
exists no x such that x < x .
```

We believe that some complexity is justified (and even required) to capture widespread mathematical idioms and formulas, the syntax of type theory, and their interactions. Our grammar is recursive to an extraordinary degree. The grammar has about 350 nonterminals and about 550 production rules. The grammar contains about 150 English words (such as *all, any, are, case, define, exists, if, iff, is, no, not, of, or, over, proof, the, theorem*, etc.) with a fixed grammatical function. User syntax extensions build on that base.

We keep most features of Forthel, such as its handling of synonyms, noun phrases, verbs, and adjectives; and its grammar extension mechanisms. We have added many additional features such as plural formation for nouns and verbs, operator precedence parsing (with user-specified precedence levels and associativities); scoping of variables; syntax for LaTeX macros; and dependent type theory including inductive and mutual inductive types, structures, and lambda terms.