

Formalising New Mathematics in Isabelle: Diagonal Ramsey

Lawrence C. Paulson   

Computer Laboratory, University of Cambridge, UK

Abstract

The formalisation of mathematics is becoming routine, but its value to research mathematicians remains unproven. There are few examples of using proof assistants to verify new work. This paper reports the formalisation – inspired by a Lean one by Bhavik Mehta – of a major new result [4] about Ramsey numbers. One unexpected finding was a heavy role for computer algebra techniques.

2012 ACM Subject Classification Theory of computation → Logic and verification; Mathematics of computing → Combinatorics; Computing methodologies → Symbolic and algebraic manipulation

Keywords and phrases Isabelle, formalisation of mathematics, Ramsey’s theorem, computer algebra

Digital Object Identifier 10.4230/LIPIcs.ITP.2025.18

Related Version *Full Version:* <https://arxiv.org/abs/2501.10852>

Supplementary Material *Other (Formal proof development):* https://www.isa-afp.org/entries/Diagonal_Ramsey.html

Acknowledgements Many thanks to Mantas Baksys, Simon Griffiths, Bhavik Mehta and Andrew Thomason for mathematical advice. Simon provided a detailed plan for fixing the proofs of lemma 9.1 and 10.1, while Andrew sketched a probabilistic proof of the lower bound for Ramsey numbers. Help with specialised Isabelle proof methods was provided by Manuel Eberl (**real_asymp**) and Fabian Immler (**approximation**). Sophie Turret and Bhavik read a draft of the paper and provided detailed, thoughtful comments; thanks also to the referees and the PC chairs.

1 Introduction

In the summer of 2017, a six week programme entitled *Big Proof* was held at Cambridge’s Isaac Newton Institute, “directed at the challenges of bringing proof technology into mainstream mathematical practice”. It attracted leading figures in the formalisation of mathematics, and marked the start of the present era of large formalised libraries. But some questioned whether the formalisation community’s previous achievements (such as the odd order theorem [10]) bore any useful relationship to present-day mathematics. The challenge was *brand-new mathematics*. And finally, we are seeing brand-new results verified prior to journal publication:

- The *Liquid Tensor Experiment*: an effort to verify a lemma by Scholze and Clausen in condensed mathematics. They were unsure about their large and complex proof, appealed to the research community, and the key result was formally checked within six months.
- A breakthrough in combinatorics: an exponential reduction in the upper bound for Ramsey numbers [4], announced in March 2023. A Cambridge PhD student, Bhavik Mehta, formalised this sensational result in months (see <https://tinyurl.com/46d2fm3d>).

Both of these were done using Lean, a proof assistant based on a dependent type theory. That raised the question of whether Isabelle, based on simple type theory, could also handle such proofs. I decided to tackle the Ramsey paper. Bhavik’s proof provided confidence that the work could be formalised, although I would work from the original article.

I should be clear at the outset about the relationship between my proof and Bhavik’s (see <https://github.com/b-mehta>). I have not borrowed any of the logical reasoning in Bhavik’s development: I can’t read Lean code, though I can just about decipher assertions.



© Lawrence C. Paulson;

licensed under Creative Commons License CC-BY 4.0

16th International Conference on Interactive Theorem Proving (ITP 2025).

Editors: Yannick Forster and Chantal Keller; Article No. 18; pp. 18:1–18:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are many gaps in the Ramsey paper and in my knowledge. When a lemma seemed to be missing or the authors' meaning was unclear, I asked others for help and sometimes looked at Bhavik's proof, which often deviated from the text. Although his formalisation was extremely helpful, I did not always follow his approach. My chief borrowings are as follows:

- the convexity argument in Lemma 4.1, discussed in §5 below (gap/error in the paper)
- an unusual Ramsey lower bound for Lemma 5.6, as discussed in §6.1 (gap in my knowledge)
- a solution to the dependence on k in $F(x, y)$, as discussed in §6.7 (error in the paper)

Minor borrowings are indicated by comments in my formalisation [16].

It is easy to port a proof from one formal language to another. I have ported hundreds of thousands of lines from the HOL Light libraries. It is great to acquire these libraries, but porting an existing formalisation has little scientific interest. The point of this project is to encode a large, technically difficult mathematical work into Isabelle and observe how it went.

Isabelle and Lean are proof assistants in which types, functions and other entities can be defined and theorems proved about them. Descriptions and documentation are widely available elsewhere (e.g. for Isabelle [14] [17, §3]). The chief difference between Isabelle and Lean is that types in the latter can depend on arbitrary parameters and not merely upon other types. The advantage of Isabelle's less expressive formalism is better automation and the avoidance of dependency issues: for example, in Lean, $i = j$ does not necessarily imply that the types $T(i)$ and $T(j)$ are the same. The Ramsey proof has nothing suggestive of dependent types, but it relies on a surprising diversity of mathematical techniques and benefits from Isabelle's advanced automation.

The paper introduces Ramsey's theorem (§2), then sketches the new algorithm reported in the Ramsey paper (§3). In the next three sections this algorithm is formalised (§4), one proof is explored in detail (§5) and the rest of the construction is surveyed (§6). There follows a discussion of computer algebra reasoning (§7) and conclusions (§8).

2 Ramsey's theorem

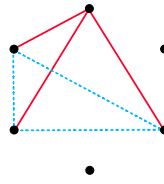
In its simplest form, Ramsey's theorem is concerned with graphs. We imagine the edges to be coloured and are interested in monochromatic (single-coloured) complete subgraphs, or *cliques*. A k -clique is a clique of k vertices; an n -graph is a graph of n vertices.

► **Theorem (Ramsey).** *For all k and ℓ there exists a number $R(k, \ell)$ such that if the edges of a complete $R(k, \ell)$ -graph G are coloured red or blue, then G will contain either a red k -clique or a blue ℓ -clique.*

Abbreviate $R(k, k)$ as $R(k)$, for Ramsey numbers on the *diagonal*. Let's prove that $R(3) \leq 6$. Since each vertex of a complete 6-graph is connected to five others, there must be at least three of the same colour, say red, from the top vertex as shown (Fig. 1). A 3-clique is a triangle, and the only way to avoid having a red triangle is to colour the dashed edges blue, creating a blue triangle. This also suggests how to colour the edges of a 5-graph, so $R(3) > 5$. We know that $R(4) = 18$ and $43 \leq R(5) \leq 46$. (The claim that $R(5) \leq 46$ is new and unconfirmed [1].) Many off-diagonal values have been determined [11], such as $R(3, 9) = 36$. In fact, $R(4, 5) = 25$ has been formally verified using HOL4 [9].

This theorem can be generalised to possibly infinite hypergraphs and much else. One focus of Ramsey theory is to establish lower and upper bounds for Ramsey numbers. Already in 1935, Erdős and Szekeres found an upper bound; Erdős later proved a lower bound, giving

$$\sqrt{2}^k \leq R(k) \leq \binom{2k-2}{k-1} < 4^k.$$



■ **Figure 1** An example to illustrate $R(3, 3) = 6$. Solid lines are red; dashed are blue.

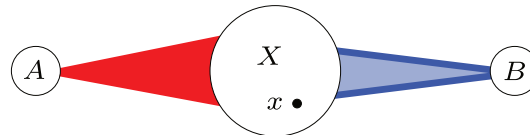
These bounds are far apart, and subsequent refinements over decades failed to improve the bases of the exponents. Hence the enthusiasm when Campos et al. [4] announced a proof that $R(k) \leq (4 - \epsilon)^k$, where ϵ was small but positive.

There is a short proof of Ramsey's theorem that establishes the 4^k bound, actually $R(k, \ell) < 2^{k+\ell}$, while also motivating the new work. It involves the execution of a simple non-deterministic algorithm operating on sets of vertices X , A , B . First, some notation: $N_R(x)$ denotes the *red neighbours* of x (the set of all y connected to x by a red edge). Similarly, $N_B(x)$ is the *blue neighbours* of x . Begin by putting all the vertices of the graph into X . Then, repeat the following as long as X is nonempty and $|A| < k$, $|B| < \ell$:

1. Pick an arbitrary vertex $x \in X$.
2. If x has no fewer red neighbours than blue in X , set $X \rightarrow N_R(x) \cap X$ and $A \rightarrow A \cup \{x\}$.
3. Otherwise, set $X \rightarrow N_B(x) \cap X$ and $B \rightarrow B \cup \{x\}$.

Step 2 leaves in X only the red neighbours of x . Because the same had been done for all the elements previously added to A , the edges between x and elements of A are all red, and by induction, A is a red clique. Analogous remarks hold for step 3: the algorithm builds a red clique in A and a blue clique in B . After at most $k + \ell - 1$ steps, we must have either a red k -clique or a blue ℓ -clique. Because each iteration (step 2 or step 3) removes (at most) approximately half the elements of X , a clique will be found if initially $|X| \geq 2^{k+\ell}$.

Figure 2 shows the algorithm in operation. We see the set X of vertices and the arbitrarily chosen element x . The red expresses that all edges between X and A are red (as are all edges within A); similarly, all edges between X and B are blue. This property is an invariant that holds initially (because A and B are empty) and is preserved by steps 2 and 3.



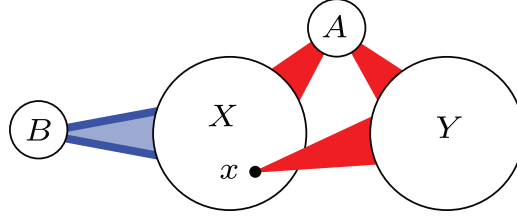
■ **Figure 2** An algorithm for proving Ramsey's theorem. (The hollow colour is blue.)

3 The improved Ramsey algorithm

The algorithm of Fig. 2 is unsophisticated in that the choice of x is arbitrary. Campos et al. [4] get a sharper bound from a better algorithm. They break the symmetry between red and blue, adding vertices to A one at a time but adding huge chunks of vertices to B . And they divide the vertices of the original graph into two separate sets, X and Y .

Key to both algorithms is the notion of a *book*: a pair of sets (S, T) , where all the edges from $S \cup T$ into S have the same colour (so S is a monochromatic clique).

The setup for their *book algorithm* includes integers k, ℓ , with $\ell \leq k$. A complete n -graph G is coloured red/blue with no red k -clique or blue ℓ -clique. The algorithm manipulates the disjoint sets X, Y, A, B , where initially $A = B = \emptyset$ while X and Y are an arbitrary



■ **Figure 3** The Book Algorithm. (The hollow colour is blue.)

partition of the vertices of G into equal (± 1) parts. A further parameter is the constant $\mu \in (0, 1)$. Figure 3 also shows some $x \in X$ being chosen on the basis of the density of red edges linking its neighbours with Y . (The *density* of edges between X and Y equals the number of such edges divided by $|X||Y|$.)

The process runs until either $|X| \leq R(k, \lceil \ell^{3/4} \rceil)$ or $p \leq 1/k$, where p denotes the density of red edges between X and Y . Here is a sketch of the possible execution steps:

- *Degree regularisation* involves removing from X all vertices having “relatively few” red neighbours in Y . This is performed at every even-numbered execution step.
- An odd-numbered step is taken as follows:
 - A *big blue step*, if there exist $R(k, \lceil \ell^{2/3} \rceil)$ vertices $x \in X$ such that

$$|N_B(x) \cap X| \geq \mu |X|,$$

then a “large” blue book (S, T) is chosen and the sets updated by

$$X \rightarrow T, \quad Y \rightarrow Y, \quad A \rightarrow A, \quad B \rightarrow B \cup S.$$

- But if not, then a *central vertex* $x \in X$ exists satisfying $|N_B(x) \cap X| \leq \mu |X|$ and a second technical condition. If the density of red edges between $N_R(x) \cap X$ and $N_R(x) \cap Y$ is sufficiently high, a *red step* is performed:

$$X \rightarrow N_R(x) \cap X, \quad Y \rightarrow N_R(x) \cap Y, \quad A \rightarrow A \cup \{x\}, \quad B \rightarrow B.$$

Otherwise, as in the previous algorithm, x is moved to B , in a *density-boost* step:

$$X \rightarrow N_B(x) \cap X, \quad Y \rightarrow N_R(x) \cap Y, \quad A \rightarrow A, \quad B \rightarrow B \cup \{x\}.$$

The paper [4] presents further definitions specifying aspects of execution traces. Its next 20 or so pages are devoted to proving that big blue steps really are big, density-boost steps really boost the density of red edges between X and Y , and providing lower bounds for Y and X as execution proceeds; the point is to show that they do not decrease too quickly as a function of execution steps of various kinds. Next comes the so-called *zigzag lemma*, which bounds the number of density-boost steps (which decrease both X and Y). It emerges through their work that one of the termination conditions, $p \leq 1/k$, can never happen: the red density remains high enough.

The authors need a further 10 pages to obtain the headline result. Here they typically take a candidate Ramsey number, construct an execution of the algorithm by defining X , Y and so forth, and use the lemmas proved previously to deduce their claims. The authors first find exponentially improved bounds for $R(k, \ell)$ when $\ell \leq k/9$ (“far from the diagonal”), then when $\ell \leq k/4$ (“closer to the diagonal”) and finally for $\ell = k$.

The proofs rely on a variety of elementary techniques for estimating real numbers, as well as asymptotic reasoning and finding upper/lower bounds for real functions. The probabilistic method pioneered by Erdős is used twice. The Isabelle formalisation [16] of the paper comprises some 11,000 lines (for the Lean one, 13,870 lines). A further 1500 lines were devoted to formalising background mathematics. The proofs are too long, difficult and intricate to present in full, but let's look at a few representative examples.

4 Defining the book algorithm in Isabelle

The diagonal paper can be seen as an abstract example of program verification. The book algorithm is indexed by a simple step counter and operates on its four-component state. The state variables at step i are written X_i, Y_i, A_i, B_i , with $i = 0$ denoting the initial state.

A succession of properties about the state are proved in terms of measures of the execution itself, primarily the numbers of steps of the various kinds. The authors assign $\mathcal{R}, \mathcal{B}, \mathcal{S}$ and $\mathcal{D}$ to the sets of indices, corresponding to an execution of a red step, a big blue step, a density-boost step and a degree regularisation step respectively, and they write $t = |\mathcal{R}|$ and $s = |\mathcal{S}|$ for the number of red steps and density-boost steps in a full execution. I found it convenient to introduce also $|\mathcal{H}|$ for the set of indices after the algorithm effectively halts.

The paper is concerned with the book algorithm's execution state but also its context: a graph of n vertices, the edges coloured red/blue and many other details. We don't want to repeat this boilerplate in every lemma statement. Luckily, Isabelle's *locale* mechanism [2] lets us declare these parameters and their constraints in one place, along with associated proofs and even dedicated syntax. A locale encapsulates a local scope, but internally it is simply a predicate.

The locale declarations for the diagonal Ramsey development give an idea what we are dealing with. For starters, although it only becomes evident much later, one parameter of the construction is $p_0 \in (0, 1)$: the minimum density of red edges between X_0 and Y_0 .

```
locale P0_min =
  fixes p0_min :: real
  assumes p0_min: "0 < p0_min" "p0_min < 1"
```

Locales can be combined flexibly to build more complicated structures. Here we combine a formalisation of finite simple graphs [6] with the previous locale, adding the assumptions that our graph is complete and the requirement that the underlying type is infinite.

```
locale Book_Basis = fin_sgraph + P0_min +
  assumes complete: "E = all_edges V"
  assumes infinite_UNIV: "infinite (UNIV::'a set)"
```

We further extend it with a colouring that has no red k -cliques or blue ℓ -cliques.

```
locale No_Cliques = Book_Basis +
  fixes Red Blue :: "'a set set"
  assumes Red_E: "Red  $\subseteq$  E"
  assumes Blue_def: "Blue = E-Red"
  fixes l::nat — blue limit
  fixes k::nat — red limit
  assumes l_le_k: "l  $\leq$  k"
  assumes no_Red_clique: " $\neg (\exists K. \text{size\_clique } k \ K \ \text{Red})$ "
  assumes no_Blue_clique: " $\neg (\exists K. \text{size\_clique } l \ K \ \text{Blue})$ "
```

We need two separate locales for the book algorithm itself. In the preliminary proofs (sections 4–8 of the diagonal paper), it is governed by a given fixed $\mu \in (0, 1)$:

```

locale Book = Book_Basis + No_Cliques +
  fixes  $\mu :: \text{real}$ 
  assumes  $\mu 01$ : " $0 < \mu$ " " $\mu < 1$ "
  fixes  $X0 :: \text{'a set}$ " and  $Y0 :: \text{'a set}$ " — initial values
  assumes  $XY0$ : " $\text{disjnt } X0 \ Y0$ " " $X0 \subseteq V$ " " $Y0 \subseteq V$ "
  assumes  $\text{density\_ge\_p0\_min}$ : " $\text{gen\_density Red } X0 \ Y0 \geq p0\_min$ "

```

In the final proofs (sections 9–11), the parameter is called γ and is defined by $\gamma = \frac{\ell}{k+\ell}$:

```

locale Book' = Book_Basis + No_Cliques +
  fixes  $\gamma :: \text{real}$ 
  assumes  $\gamma\_def$ : " $\gamma = \text{real } 1 / (\text{real } k + \text{real } 1)$ "
  fixes  $X0 :: \text{'a set}$ " and  $Y0 :: \text{'a set}$ "
  assumes  $XY0$ : " $\text{disjnt } X0 \ Y0$ " " $X0 \subseteq V$ " " $Y0 \subseteq V$ "
  assumes  $\text{density\_ge\_p0\_min}$ : " $\text{gen\_density Red } X0 \ Y0 \geq p0\_min$ "

```

Much of the formalisation takes place within locale *Book*, with the parameters specified above immediately accessible. Towards the end, while proving the headline claims, we start outside the locale. Then we create a graph, colour its edges, choose X_0 and Y_0 , thereby creating an instance of *Book*. These steps correspond to the authors' remark "we now apply the book algorithm".

Here is a glimpse at the algorithm itself. Recall that an odd-numbered step can be a big blue step, a red step or a density-boost step.

```

definition next_state :: " $\text{'a config} \Rightarrow \text{'a config}$ " where
  "next_state  $\equiv \lambda(X,Y,A,B).$ 
    if many_bluish  $X$ 
    then let  $(S,T) = \text{choose\_blue\_book } (X,Y,A,B)$  in  $(T, Y, A, B \cup S)$ 
    else let  $x = \text{choose\_central\_vx } (X,Y,A,B)$  in
      if reddish  $k \ X \ Y$  ( $\text{red\_density } X \ Y$ )  $x$ 
      then  $(\text{Neighbours Red } x \cap X, \text{Neighbours Red } x \cap Y, \text{insert } x \ A, B)$ 
      else  $(\text{Neighbours Blue } x \cap X, \text{Neighbours Red } x \cap Y, A, \text{insert } x \ B)$ "

```

The execution stepper itself must also take care of the even-numbered steps (degree regularisation) and the termination condition (when it leaves the next state unchanged):

```

primrec stepper :: " $\text{nat} \Rightarrow \text{'a config}$ " where
  "stepper 0 =  $(X0,Y0,\{\},\{\})$ "
| "stepper (Suc n) =
  (let  $(X,Y,A,B) = \text{stepper } n$  in
    if termination_condition  $X \ Y$  then  $(X,Y,A,B)$ 
    else if even  $n$  then degree_reg  $(X,Y,A,B)$  else next_state  $(X,Y,A,B)$ )"

```

Functions like *many_bluish* and *choose_blue_book* refer to details of the algorithm omitted here. Many other definitions and basic lemmas are also omitted.

5 A detailed peek at formalising a proof

The first substantial proof is Lemma 4.1, stated by the authors as follows [4, p. 11]:

► **Lemma (4.1).** *Set $b = \ell^{1/4}$. If there are $R(k, \ell^{2/3})$ vertices $x \in X$ such that $|N_B(x) \cap X| \geq \mu|X|$, then X contains either a red k -clique, or a blue book (S, T) with $S \geq b$ and $|T| \geq \mu^{|S|}|X|/2$.*

Here X is any set, not necessarily a state component of the book algorithm, though that's how the lemma will be applied. The proof is under a page, but it presents many complications. For starters, some trivia: it is already assumed that there are no red k -cliques in the given graph, so we can already rule out the possibility of such a clique in X . And by convention, expressions like $R(k, \ell^{2/3})$ don't indicate whether the floor or the ceiling of $\ell^{2/3}$ is intended. The omission looks strange to me; in most cases, I have assumed the ceiling.

A more serious omission is that the theorem only holds for sufficiently large ℓ and all $k \geq \ell$. This ℓ may depend upon μ alone; moreover, it must be constant as μ ranges over a closed subinterval of $(0, 1)$. The need for this becomes apparent only 19 pages later, in Lemma 9.3. Constraints on ℓ (or, more usually, k) need to be picked up here and there in the proof. It is quite common that a claimed inequality does not hold in general, but only for sufficiently large k .

So we need to decide how to formalise the “sufficiently large” condition. The elegant approach uses the *eventually* topological filter [13], which is available in both Isabelle/HOL and Lean. Initially, I followed Bhavik and used *eventually*, making the various constraints local to each proof and thus hidden. I later adopted the approach of making these constraints explicit: to define a *bigness predicate* for each theorem, justifying each predicate by proving a lemma (expressed using *eventually*) to check that the predicate indeed holds for all sufficiently large ℓ . This seemed more natural, while also retaining access to the specific conditions used. Eventually it allowed k, ℓ, μ and their constraints to be migrated from the theorem statements to the *Book* locale, to be written only once rather than many times.

The bigness predicate *Big_Blue_4_1*, for Lemma 4.1, is a conjunction of six inequalities, such as $l \geq (6/\mu)^{12/5}$. Recall that the property may depend upon μ alone and the obtained ℓ must be good over a closed interval. The corresponding *bigness lemma* refers to the closed interval $[\mu_0, \mu_1]$, where $0 < \mu_0$. Here there is no need to assume $\mu_1 < 1$, so the precondition could be simplified to $\mu \geq \mu_0$. Uniformity in the lemma statements will be convenient however, since there will be lots of them. The proof, some 40-odd lines, is routine and deals with the six inequalities separately.

lemma *Big_Blue_4_1*:

assumes “ $0 < \mu_0$ ”

shows “ $\forall \mu. \mu \in \{\mu_0.. \mu_1\} \longrightarrow \text{Big_Blue_4_1 } \mu \ 1$ ”

Here is the formal statement of Lemma 4.1. It's easy to state, because the most technical parts of the theorem statement (*many_bluish* and *good_blue_book*) have already been formalised for the specification of the book algorithm.

lemma *Blue_4_1*:

assumes “ $X \subseteq V$ ” **and** *manyb*: “*many_bluish* X ”

and *big*: “*Big_Blue_4_1* $\mu \ 1$ ”

shows “ $\exists S \ T. \text{good_blue_book } X \ (S, T) \wedge \text{card } S \geq 1 \text{ powr } (1/4)$ ”

Before proving this, it was necessary to build a basic library of Ramsey numbers, including the Erdős lower bound $2^{k/2} \leq R(k, k)$ by the famous probabilistic argument. The formal proof of Lemma 4.1 is some 350 lines; let's look at some of the details.

Early in the proof we obtain a blue m -clique, $U \subseteq X$, where $m = \ell^{2/3}$. The text continues:

Let σ be the density of blue edges between U and $X \setminus U$, and observe that

$$\sigma = \frac{e_B(U, X \setminus U)}{|U| \cdot |X \setminus U|} \geq \frac{\mu|X| - |U|}{|X| - |U|} \geq \mu - \frac{1}{k}.$$

The first equality holds by definition, but the two subsequent inequalities require heavy manipulations of sums: more than 70 proof lines even to yield the weaker claim $\sigma \geq \mu - 2/k$. (The weakening, a hint from Bhavik, does no harm: eventually we'll need $2\ell^{-5/12} \leq \mu - 2/k$, which holds for large enough ℓ . It simply becomes part of the bigness predicate.) The first inequality follows by careful counting of edges, while the second relies on $|X| \geq R(k, m)$ and properties of Ramsey numbers. This painful example gives a good impression of how dense the presentation is throughout the diagonal paper. The proof continues:

Let $S \subset U$ be a uniformly-chosen random subset of size b , and let $Z = |N_B(S) \cap (X \setminus U)|$ be the number of common blue neighbours of S in $X \setminus U$. By convexity, we have

$$\mathbb{E}[Z] = \binom{m}{b}^{-1} \sum_{v \in X \setminus U} \binom{|N_B(v) \cap U|}{b} \geq \binom{m}{b}^{-1} \binom{\sigma m}{b} \cdot |X \setminus U|.$$

The first equality requires a probabilistic argument. The expectation of $Z(S)$ for the randomly chosen subset S can be formalised by setting up a probability space [8]. “Convexity” is doing a lot in the inequality. The function f is said to be *convex* if $f(\sum_{i \in S} a_i y_i) \leq \sum_{i \in S} a_i f(y_i)$ where S is nonempty and finite, $a_i \geq 0$ for $i \in S$ and the a_i sum to 1. In our proof, each $a_i = |X \setminus U|^{-1}$ and the convex function is the real-valued generalised binomial $\lambda x \cdot \binom{x}{b}$, which unfortunately is not actually convex. I adopted Bhavik’s elaborate fix for this problem, which was to define an alternative real-valued binomial that was convex and agreed with the original where it mattered, on the integers.

The proof continues:

Now, by Fact 4.2, and recalling (10), and that $b = \ell^{1/4}$ and $m = \ell^{2/3}$, it follows that

$$\mathbb{E}[Z] \geq \sigma^b \exp\left(-\frac{b^2}{\sigma m}\right) \cdot |X \setminus U| \geq \frac{\mu^b}{2} \cdot |X|,$$

and hence there exists a blue clique $S \subset U$ of size b with at least this many common blue neighbours in $X \setminus U$, as required. $\square$

Several dozen detailed and tedious lines of inequality calculations take care of the probability and counting arguments making up this final part. Lemma 4.1 is among the trickier theorems but its proof is typical of the sort of reasoning used throughout the paper: estimation and calculations involving edge density, edges and vertices. On the other hand, the paper has only one further probabilistic argument: to derive a non-standard lower bound for Ramsey numbers.

6 A progression of theorems

It isn’t possible to go through any other proofs in such detail, but let’s have a glimpse at the key milestones. To begin, let’s establish some conventions. Recall that the book algorithm operates on state variables X , Y , A and B . Some other variables are defined in terms of them, notably p_i , the density of red edges between X_i and Y_i .

We have an issue with the subscripts: the Isabelle and Lean formalisations both write X_i for the value of X after i execution steps, when Campos et al. [4] write X_{i-1} . Complicating matters further, everyone agrees to write x_i for the value of the “central vertex” after i execution steps.

In Isabelle, X_i is written $Xseq\ i$ and similarly for Y , A , B and p_i . Note that $\epsilon = k^{-1/4}$, so ϵ depends on k : it is formalised as $\epsilon ps\ k$ but within locales where k is available (`No_Cliques` and beyond) it’s abbreviated to simply ϵ . Some theorems refer to the situation after the algorithm halts; the step number at which it halts is written `halted_point`.

6.1 On red and density-boost steps

The headline result of section 5 is entitled “density-boost steps boost the density”. This refers to the density of red edges between X and Y : to be precise, the current values of those state variables. The local definitions in the theorem statement correspond to abbreviations introduced by the authors. The first assumption is that the current execution step is red or density-boost, $i \in \mathcal{R} \cup \mathcal{S}$ in the diagonal paper. *Step_class* is a convenient way to refer to multiple sets of steps.

```

lemma Red_5_1:
  assumes  $i: "i \in \text{Step\_class } \{\text{red\_step}, \text{dboost\_step}\}"$ 
    and  $\text{Big}: " \text{Big\_Red\_5\_1 } \mu \ 1 "$ 
  defines  $"p \equiv \text{pseq } i"$ 
  defines  $"x \equiv \text{cvx } i"$ 
  defines  $"X \equiv X\text{seq } i"$  and  $"Y \equiv Y\text{seq } i"$ 
  defines  $"NBX \equiv \text{Neighbours Blue } x \cap X"$ 
  defines  $"NRX \equiv \text{Neighbours Red } x \cap X"$ 
  defines  $"NRY \equiv \text{Neighbours Red } x \cap Y"$ 
  defines  $"\beta \equiv \text{card } NBX / \text{card } X"$ 
  shows  $"\text{red\_density } NRX \ NRY \geq p - \alpha \ (\text{hgt } p)$ 
     $\vee \text{red\_density } NBX \ NRY \geq p + (1-\epsilon) * ((1-\beta)/\beta) * \alpha(\text{hgt } p) \wedge \beta > 0"$ 

```

One of the more interesting “moments” in the buildup to its proof was Lemma 5.6, where we find a little throwaway comment:

On the other hand, if we colour the edges independently at random, with probability $k^{-1/8}$ of being blue, then a simple (and standard) 1st moment argument shows that $R(k, \ell^{3/4}) \geq \exp(c \ell^{3/4} \log k)$ for some absolute constant $c > 0$.

I turned to Bhavik’s development to decipher this impossibly cryptic remark, arriving at a claim formalised in Isabelle as follows:

```

lemma Ramsey_number_lower_simple:
  fixes  $p::\text{real}$ 
  assumes  $n: "n^k * p^{\text{powr } (k^2 / 4)} + n^1 * \exp (-p * 1^2 / 4) < 1"$ 
  assumes  $p01: "0 < p" \ "p < 1" \ \text{and} \ "k > 1" \ "1 > 1"$ 
  shows  $"\neg \text{is\_Ramsey\_number } k \ 1 \ n"$ 

```

This actually has a fairly simple proof, appealing to a more general theorem about Ramsey numbers already proved (thanks to a sketch given me by Andrew Thomason). The claim “density-boost steps boost the density” is only evident in this corollary of Lemma 5.1:

```

corollary Red_5_2:
  assumes  $i: "i \in \text{Step\_class } \{\text{dboost\_step}\}"$ 
    and  $\text{Big}: " \text{Big\_Red\_5\_1 } \mu \ 1 "$ 
  shows  $"\text{pseq } (\text{Suc } i) - \text{pseq } i \geq$ 
     $(1-\epsilon) * ((1 - \text{beta } i) / \text{beta } i) * \alpha(\text{hgt } (\text{pseq } i)) \wedge \text{beta } i > 0"$ 

```

The conclusion above is clearer in mathematical notation: $p_{i+1} - p_i \geq (1 - \epsilon) \left(\frac{1 - \beta_i}{\beta_i} \right) \alpha_h$. The right-hand side above is strictly positive. Space prohibits saying much about α_h , but $\beta_i \in (0, 1)$ is defined by $\beta_i = |N_B(x_i) \cap X_i| / |X_i|$.

6.2 Bounding the size of Y

The book algorithm discards vertices from X and/or Y at each step, so we need to know that these sets don't get too small. In particular, both density-boost steps ($\mathcal{S}$) and red steps ($\mathcal{R}$) discard from Y . Writing s for $|\mathcal{S}|$ and t for $|\mathcal{R}|$, here is section 6's headline result.

► **Lemma (6.1).** *If $p_0 = \Omega(1)$, then $|Y| \geq 2^{o(k)} p_0^{s+t} \cdot |Y_0|$.*

The use of Landau symbols is perplexing. The actual meaning of $p_0 = \Omega(1)$ is seemingly just that there is some positive floor P_0 on possible values of p_0 , specified in the formalisation by the locale `P0_min` presented in §4. In Isabelle, $2^{o(k)}$ is formalised with reference to the $o(k)$ function $2k \log_2(1 - 2\sqrt{\epsilon}/P_0)$, which serves the purpose if $P_0 > 2\sqrt{\epsilon}$. Since $\epsilon = k^{-1/4}$, that condition holds for a large enough k and becomes part of the “bigness assumption” of this lemma. An $o(k)$ function is one whose absolute value grows more slowly than the identity; this particular function is actually negative. As it is a function of k , independent of the k fixed in our locales, ϵ is formalised as the function `eps k`.

definition `"ok_fun_61 ≡`
`λk. (2 * real k / ln 2) * ln (1 - 2 * eps k powr (1/2) / p0_min)"`

Now it is possible to formalise the statement of Lemma 6.1.

lemma `(in Book) Y_6_1:`
`assumes big: "Big_Y_6_1 μ 1"`
`defines "st ≡ Step_class {red_step, dboost_step}"`
`shows "card (Yseq halted_point) / card Y0 ≥ 2 powr (ok_fun_61 k) * p0 ^ card st"`

6.3 Bounding the size of X

As in the previous section, our task is to show that density-boost ($\mathcal{S}$) and red ($\mathcal{R}$) steps don't cause too much damage. Recall that $\mu \in (0, 1)$ is a parameter of the book algorithm and that $s = |\mathcal{S}|$ and $t = |\mathcal{R}|$. The headline result of section 7 refers to a further parameter $\beta \in (0, 1)$, whose complicated definition is omitted here.

► **Lemma (7.1).** $|X| \geq 2^{o(k)} \mu^\ell (1 - \mu)^t \left(\frac{\beta}{\mu}\right) |X_0|$.

And the same statement, formalised in Isabelle/HOL.

lemma `(in Book) X_7_1:`
`assumes big: "Big_X_7_1 μ 1"`
`defines "D ≡ Step_class {dreg_step}"`
`defines "R ≡ Step_class {red_step}" and "S ≡ Step_class {dboost_step}"`
`shows "card (Xseq halted_point) ≥ 2 powr ok_fun_71 μ k *`
`μ ^ 1 * (1 - μ) ^ card R * (bigbeta/μ) ^ card S * card X0"`

In this case, the construction of the $o(k)$ function is less straightforward. Lemma 7.1 follows from a succession of other lemmas, such as Lemma 7.2:

$$\prod_{i \in \mathcal{R}} \frac{|X_{i+1}|}{|X_i|} \geq 2^{o(k)} (1 - \mu)^t.$$

So `ok_fun_71`, the $o(k)$ function for Lemma 7.1, is defined as the sum of four other $o(k)$ functions used in the other lemmas building up to it. This section was painful to formalise.

6.4 The zigzag lemma

This lemma bounds the number t of density-boost steps, limiting their wastage of vertices in X . It refers to the central vertex x_i and to $\mathcal{S}^*$, a subset of the density-boost steps with a highly technical definition. Now Lemma 8.1 is stated as $\sum_{i \in \mathcal{S}^*} \frac{1-\beta_i}{\beta_i} \leq t + o(k)$.

In the Isabelle version, the $o(k)$ function is written explicitly as $k^{19/20}$:

```
lemma (in Book) ZZ_8_1:
  assumes big: "Big_ZZ_8_1  $\mu$  1"
  defines "R  $\equiv$  Step_class {red_step}"
  defines "sum_SS  $\equiv$  ( $\sum i \in \text{dboost\_star. (1 - beta i) / beta i}$ )"
  shows "sum_SS  $\leq$  card R + k powr (19/20)"
```

The section is only three pages long and yielded a quite short formalisation.

6.5 An exponential improvement far from the diagonal

Sections 9–12 of the diagonal paper use the accumulated knowledge about the book algorithm to achieve significant new results. “Far from the diagonal” means $\ell \leq k/9$. The headline result of section 9 is stated as follows.

► **Theorem (9.1).** *If $\gamma \leq 1/10$ and $\delta = \gamma/20$, then $R(k, \ell) \leq e^{-\delta k + o(k)} \binom{k+\ell}{\ell}$ for all $k, \ell \in \mathbb{N}$ with $\gamma = \frac{\ell}{k+\ell}$.*

The formal version below is weaker, imposing a “for all big enough ℓ ” condition not present in the original. It also assumes that $p0_min$, the minimum possible p_0 , must not exceed approximately 0.89. On the plus side, the $o(k)$ function has been replaced by simply 1.

```
theorem Far_9_1:
  fixes l k::nat
  fixes  $\delta$   $\gamma$ ::real
  defines " $\gamma \equiv \text{real } l / (\text{real } k + \text{real } l)$ "
  defines " $\delta \equiv \gamma/20$ "
  assumes  $\gamma$ : " $\gamma \leq 1/10$ "
  assumes big: "Big_Far_9_1  $\gamma$  1"
  assumes p0_min_91: " $p0\_min \leq 1 - (1/10) * (1 + 1/15)$ "
  shows "RN k l  $\leq \exp (-\delta*k + 1) * (k+l \text{ choose } l)$ "
```

Also notable is that all the lemmas presented before now were proved within the *Book* locale, providing the full environment and assumptions of the book algorithm. Theorem 9.1 is proved in locale *p0_min*, which specifies nothing other than $p0_min$.

It is instructive to see how the pieces fit together. Theorem 9.1 requires the following:

► **Lemma (9.2).** *Let $\gamma \leq 1/10$, and let $k, \ell \in \mathbb{N}$ be sufficiently large integers with $\gamma = \frac{\ell}{k+\ell}$. Let $\delta = \gamma/20$ and $\eta \leq \gamma/15$, and suppose that $n \geq e^{-\delta k} \binom{k+\ell}{\ell}$. Then every red-blue colouring of the edges of the complete n -graph in which the density of red edges is at least $1 - \gamma - \eta$ contain either a red k -clique or a blue ℓ -clique.*

The proof relies on the lemmas proved earlier within the locale *Book*: they refer to the book algorithm and its implicit assumption that there exists **no** red k -clique or blue ℓ -clique, so we must replace the conclusion above by falsity. We formalise the lemma in a series of steps. First, we prove a modified form of Lemma 9.2 under the assumptions of the book algorithm¹ and concluding *False*.

¹ Recall that locale *Book'* is like *Book* but replacing the parameter μ by $\gamma = \ell/(k + \ell)$.

18:12 Formalising New Mathematics

```

lemma (in Book') Far_9_2_aux:
  fixes  $\delta \ \eta :: \text{real}$ 
  defines " $\delta \equiv \gamma/20$ "
  assumes "real (card  $X_0$ )  $\geq nV/2$ " "card  $Y_0 \geq nV \text{ div } 2$ " " $p_0 \geq 1-\gamma-\eta$ "
  assumes " $\gamma \leq 1/10$ " and  $\eta: "0 \leq \eta"$  " $\eta \leq \gamma/15$ "
  assumes "real  $nV \geq \exp(-\delta * k) * (k+1 \text{ choose } 1)$ "
  assumes big: "Big_Far_9_2  $\gamma \ 1$ "
  shows False

```

Its proof formalises that of Lemma 9.2 in the diagonal paper. But a little more work is needed to get the lemma statement into the correct form: specifically, to set up the invocation of the book algorithm given a complete graph with a red-blue colouring of its edges having no red k -clique or blue ℓ -clique. Assuming the density of red edges is large enough and that $\mu \in (0, 1)$, it is straightforward to partition the vertex set into X_0 and Y_0 with a sufficiently large red-edge density between them. The conclusion is expressed using the *Book* predicate, so this lemma makes it easy to move up from the *No_Cliques* locale to the *Book* locale.

```

lemma (in No_Cliques) to_Book:
  assumes " $p_{0\_min} \leq \text{graph\_density Red}$ "
  assumes " $0 < \mu$ " " $\mu < 1$ "
  obtains  $X_0 \ Y_0$  where " $1 \geq 2$ " "card  $X_0 \geq \text{real } nV / 2$ " "card  $Y_0 = \text{gorder div } 2$ "
    and " $X_0 = V \setminus Y_0$ " " $Y_0 \subseteq V$ "
    and " $\text{graph\_density Red} \leq \text{gen\_density Red } X_0 \ Y_0$ "
    and " $\text{Book } V \ E \ p_{0\_min} \ \text{Red Blue } 1 \ k \ \mu \ X_0 \ Y_0$ "

```

Note that **obtains** is a convenient way of expressing an existential conclusion. Here it is logically equivalent to $\exists X_0 Y_0. \dots$. A corollary (similar and omitted) is *to_Book'*, for instantiating the *Book'* locale.

And so we obtain Lemma 9.2 in its final form, proved in the *No_Cliques* locale. This manipulation of locales is not difficult and provides a uniform treatment of the framework of the book algorithm and associated notation.

```

lemma (in No_Cliques) Far_9_2:
  fixes  $\delta \ \gamma \ \eta :: \text{real}$ 
  defines " $\gamma \equiv 1 / (\text{real } k + \text{real } 1)$ "
  defines " $\delta \equiv \gamma/20$ "
  assumes " $\text{graph\_density Red} \geq 1-\gamma-\eta$ " and " $p_{0\_min} \leq 1-\gamma-\eta$ "
  assumes " $\gamma \leq 1/10$ " and " $0 \leq \eta$ " " $\eta \leq \gamma/15$ "
  assumes "real  $nV \geq \exp(-\delta * k) * (k+1 \text{ choose } 1)$ "
  assumes big: "Big_Far_9_2  $\gamma \ 1$ "
  shows False

```

From this, formalising the proof of Theorem 9.1 takes another 450+ lines. The proof is by contradiction, assuming $R(k, \ell) > e^{-\delta k + o(k)} \binom{k+\ell}{\ell}$ and defining n (the number of vertices in the graph) to be $R(k, \ell) - 1$. We immediately obtain a clique-free red/blue colouring of the edges, and eventually use Lemma 9.2 (and therefore the book algorithm) in one case, for this graph restricted to a subset U of the vertices.

There is a subtle error in the proof of 9.1 that I was able to patch with the help of Simon Griffiths, one of the authors. He sent me a lengthy message in which he described an “inelegant” solution, which he had earlier sent to Bhavik, along with a more speculative “elegant” solution. I decided to go for the latter, but had to do quite a bit of further work to complete the argument. Here, Isabelle was helping to develop the mathematics itself.

6.6 An exponential improvement a little closer to the diagonal

For section 10, “a little closer to the diagonal” means $\ell \leq k/4$, and the headline result is stated as follows.

► **Theorem (10.1).** *If $\gamma \leq 1/5$ and $\delta = \gamma/40$, then $R(k, \ell) \leq e^{-\delta k + o(k)} \binom{k+\ell}{\ell}$ for all $k, \ell \in \mathbb{N}$ with $\gamma = \frac{\ell}{k+\ell}$.*

The similarity to the previous theorem should be obvious, and it is proved in a somewhat similar fashion from the following lemma.

► **Lemma (10.2).** *Let $k, \ell \in \mathbb{N}$ be sufficiently large, and set $\gamma = \frac{\ell}{k+\ell}$. If $1/10 \leq \gamma \leq 1/5$ and $n \geq e^{-k/200} \binom{k+\ell}{\ell}$, then every red-blue colouring of the edges of the complete n -graph in which the density of red edges is at least $1 - \gamma$ contain either a red k -clique or a blue ℓ -clique.*

The proof of the theorem from the lemma has much in common with the analogous proof for section 9 and even an analogous error, but some differences are worth noting. Here is the formalisation of the headline result:

```
theorem Closer_10_1:
  fixes l k::nat
  fixes δ γ::real
  defines "γ ≡ real 1 / (real k + real l)"
  defines "δ ≡ γ/40"
  defines "γ0 ≡ min γ (0.07)"
  assumes big: "Big_Closer_10_1 γ0 l"
  assumes γ: "γ ≤ 1/5"
  assumes "p0_min ≤ 1 - 1/5"
  shows "RN k l ≤ exp (-δ*k + 3) * (k+l choose l)"
```

As before, it looks weaker than it should be. Once again it assumes, contrary to the statement of 10.1, that ℓ needs to be sufficiently large – and worse, the relevant “bigness lemma” refers to the mysterious parameter γ_0 , defined to be $\min(\gamma, 0.07)$. In fact $0.07 < 1/10 - 1/36$ and is concerned with an application of Theorem 9.1 (when $\gamma \leq 1/10$) when ℓ has been replaced by a somewhat smaller value arising from the restriction of our graph to a subgraph. It turns out we can deduce $k \geq 36$ from our bigness assumptions (I suspect that it must be vastly bigger). The other point is that once again the claimed $o(k)$ function is degenerate, in this case simply the value 3.

An unconditional (bigness-free) version of Theorem 10.1 seems necessary to complete the formalisation. We obtain one using a suitable $o(k)$ function: the one that returns 3 if the given k is big enough (we incorporate bigness into the function) and otherwise returns δk , which works since $R(k, \ell) \leq \binom{k+\ell}{\ell}$. This version still has a stray assumption about $p0_min$.

```
theorem Closer_10_1_unconditional:
  fixes l k::nat
  fixes δ γ::real
  defines "γ ≡ real 1 / (real k + real l)"
  defines "δ ≡ γ/40"
  assumes γ: "0 < γ" "γ ≤ 1/5"
  assumes p0_min_101: "p0_min ≤ 1 - 1/5"
  shows "RN k l ≤ exp (-δ*k + ok_fun_10_1 γ k) * (k+l choose l)"
```

6.7 Concluding the proof of the main theorem

The proof ends in a final blaze of technical complexity, of which only hints can be shown. Section 11 defines two functions, for $x, y \in (0, 1)$ and (eventually) $\mu = 2/5$:

$$F(x, y) = \frac{1}{k} \log_2 R(k, k - xk) + x + y$$

$$G_\mu(x, y) = \log_2 \left(\frac{1}{\mu} \right) + x \cdot \log_2 \left(\frac{1}{1 - \mu} \right) + y \cdot \log_2 \left(\frac{\mu(x + y)}{y} \right).$$

The objective of section 11, which is entitled “from off-diagonal to diagonal”, is this:

► **Theorem (11.1).** *Fix $\mu \in (0, 1)$ and $\eta > 0$. We have*

$$\frac{\log_2 R(k)}{k} \leq \max_{\substack{0 \leq x \leq 1 \\ 0 \leq y \leq \mu x / (1 - \mu) + \eta}} \min \{ F(x, y), G_\mu(x, y) \} + \eta$$

for all sufficiently large $k \in \mathbb{N}$.

It looks like we are nearing our destination, because an upper bound on $\log_2 R(k)/k$ looks like giving an upper bound on $R(k)$. Unfortunately, this doesn't quite work because $F(x, y)$ hides a dependence on k that shouldn't be there. The correct version of Theorem 11.1 is stated in terms of other functions, which are actually defined in section 12. We need the binary entropy function $h(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)$ to define the following:

$$f_1(x, y) = x + y + (2 - x) \cdot h \left(\frac{1}{2 - x} \right)$$

$$f_2(x, y) = f_1(x, y) - \frac{\log_2 e}{40} \left(\frac{1 - x}{2 - x} \right)$$

$$f(x, y) = \begin{cases} f_1(x, y) & \text{if } x < 3/4 \\ f_2(x, y) & \text{if } x \geq 3/4 \end{cases}$$

The definitions above are formalised in the obvious way, where ff is f and GG is G . But we seemingly need to modify the theorem statement by imposing a lower bound of 1.9 on the minimum:

definition *"ffGG $\equiv \lambda \mu \ x \ y. \max 1.9 \ (\min (ff \ x \ y) \ (GG \ \mu \ x \ y))"$*

Here is Theorem 11.1 as formalised. The additional conditions make it weaker.

theorem (in *P0_min*) *From_11_1:*

assumes μ : *"0 < μ " " $\mu \leq 2/5$ " and "0 < η " " $\eta \leq 1/12$ "*

and *p0_min12: "p0_min $\leq 1/2$ " and big: "Big_From_11_1 $\eta \ \mu \ k$ "*

shows *"log 2 (RN k k) / k $\leq$ (SUP x $\in$ {0..1}. SUP y $\in$ {0..3/4}. ffGG $\mu \ x \ y + \eta$)"*

Note that this theorem is situated in the trivial locale *P0_min*. The proof, somewhat like others we have discussed, begins by setting $n = R(k, k) - 1$, obtaining a clique-free red/blue colouring of the edges of the complete n -graph. There is a case analysis on whether the density of red or blue edges is greater; either way, the *Book* locale is interpreted, allowing the use of lemmas proved earlier. We are finally close to our objective of proving

► **Theorem (1.1).** *There exists $\epsilon > 0$ such that $R(k) \leq (4 - \epsilon)^k$ for all sufficiently large $k \in \mathbb{N}$.*

The sting in the tail comes at the very end of the argument. The formal version of this nasty lemma is similar but uses the function `ffGG`, as in the formalisation of Theorem 11.1.

► **Lemma (12.3).** *If $\delta \leq 2^{-11}$, then $\max_{\substack{0 \leq x \leq 1 \\ 0 \leq y \leq 3/4}} \min\{f(x, y), g(x, y)\} < 2 - \delta$.*

The authors give a two page proof, with a diagram showing the regions where either f or g is too large (where g is $G_{2/5}$). The two regions must not overlap, but they are separated only by a hair's breadth. The authors remark that the proof is “a simple calculation” that could be checked by computer, which takes us to the topic of verified calculations.

7 Formal reasoning about computer algebra

During the 1990s, researchers sought to combine the capabilities of computer algebra systems (such as Maple and Mathematica) with interactive theorem provers. Computer algebra systems could deliver wrong answers, due to overly aggressive simplification or outright bugs. A student of mine, Clemens Ballarin, worked on this problem [3]. A difficulty was finding an application, since most computer algebra users are not performing proofs. But the diagonal Ramsey proof needs computer algebra everywhere: derivatives (typically for identifying maxima), limits, continuity, Landau symbols, high precision exact real arithmetic. We are frequently forced to perform precise numerical calculations or to prove the existence of a large enough ℓ satisfying some complicated condition.

Isabelle/HOL provides a number of automatic tools to carry out such tasks:

- Eberl's `real_asymp` proof method [5], to prove theorems about limits and Landau symbols for real-valued functions
- Hölzl's `approximation` method [12], to do exact real calculations using interval arithmetic
- the rule set `derivative_eq_intros`, useful for performing symbolic differentiation

The last item needs further explanation, because it is not packaged as a neat proof method. Taking the symbolic derivative of a function by repeatedly applying syntactic rules is an easy programming exercise dating back nearly 60 years [18]. Theorems stating the rules for differentiating functions such as multiplication, division, sine or cosine, packaged in the identifier `derivative_eq_intros`, can take derivatives of quite large expressions when interleaved with calls to the simplifier.

Here is an example, from Lemma 9.3, proving that the function $1 - 1/200x$ is concave on $[1/10, 1/5]$ by calculating its second derivative and noting that $-1/(100x^3) \leq 0$ if $x \in (0, 1)$.

```
show "concave_on {1/10..1/5} ( $\lambda x. 1 - 1/(200*x)$ )"
proof (intro f'_le0_imp_concave)
  fix x::real
  assume "x  $\in$  {1/10..1/5}"
  then have x01: "0 < x" "x < 1" by auto
  show "(( $\lambda x. (1 - 1/(200*x))$ ) has_real_derivative 1/(200*x^2)) (at x)"
    using x01 by (intro derivative_eq_intros | force simp: eval_nat_numeral)+
  show "(( $\lambda x. 1/(200*x^2)$ ) has_real_derivative -1/(100*x^3)) (at x)"
    using x01 by (intro derivative_eq_intros | force simp: eval_nat_numeral)+
  show "-1/(100*x^3)  $\leq$  0"
    using x01 by (simp add: divide_simps)
qed auto
```


18:16 Formalising New Mathematics

The diagonal Ramsey formal development includes 16 symbolic differentiations, 30 calls to **approximation** and 60 calls to **real_asymp**. Many of these would be difficult to establish without those tools. Let's look at the formal proof that

$$\frac{k}{\log 2} \cdot \log \left(1 - \frac{1}{k(1-\mu)} \right) = o(k)$$

This is the $o(k)$ function mentioned in Lemma 7.2, which we saw in §6.3 above. The formal proof requires merely unfolding the definition, then calling **real_asymp**. (The complexity class $o(k)$ is formalised as $o(\text{real})$, where **real** is the injection from $\mathbb{N}$ into $\mathbb{R}$.)

definition "ok_fun_72 $\equiv \lambda \mu k. (\text{real } k / \ln 2) * \ln (1 - 1 / (k * (1-\mu)))$ "

lemma ok_fun_72:

assumes " $\mu < 1$ " **shows** "ok_fun_72 $\mu \in o(\text{real})$ "

using **assms** **unfolding** ok_fun_72_def **by** real_asymp

Many of the asymptotic lemmas claim that some property $P(\mu, \ell)$ holds for all sufficiently large ℓ and all μ ranging over some closed subinterval $[\mu_0, \mu_1] \subset (0, 1)$. The **real_asymp** method does not work for intervals, but monotonicity helps: if $P(\mu, \ell)$ is preserved as μ increases, then it is enough to check $P(\mu_0)$, and if it is preserved as μ decreases, it is enough to check $P(\mu_1)$. And one or the other held every time.

The calls to **approximation** come in two forms. Most are simple calculations, such as $835.81 \leq 3 \cdot \log_2(5/3)/5 \cdot \log_2 2727$, which requires high precision. However, the proof of Lemma 12.3, mentioned above, twice requires reasoning over a closed interval.

```
have ?thesis if "y ∈ {1/10 .. 3/4}"
  using that unfolding gg_eq x_of_def
  by (approximation 24 splitting: y = 12)
```

The *splitting* keyword controls the interval splitting in the search. The claim proved is

$$\log 2 (5/2) + (3 * y / 5 + 5454 / 10^4) * \log 2 (5/3) + y * \log 2 (2 * (3 * y / 5 + 5454 / 10^4 + y) / (5 * y)) \leq 2 - 1/2^{11}$$

The two **approximation** calls with splitting use 27 seconds of CPU time in total. In comparison, the entire diagonal Ramsey proof (including those calls) takes 218 seconds on an M2 MacBook Pro.

8 Discussion and conclusions

This project was a huge amount of work: 251 days, just over eight months. Its de Bruijn index (the ratio between the formal and informal text, after compression) is relatively high, at 5.6, a sign of the work's difficulty. (Common is 4 [7].) Here is the proof of the main theorem, which includes limit reasoning using filters and a call to **approximation**.

theorem Main_1_1:

obtains $\varepsilon :: \text{real}$ **where** " $\varepsilon > 0$ " " $\forall^\infty k. \text{RN } k \ k \leq (4-\varepsilon)^k$ "

proof

let $\varepsilon = "0.00134 :: \text{real}"$

have " $\forall^\infty k. k > 0 \wedge \log 2 (\text{RN } k \ k) / k \leq 2 - \text{delta}'$ "

unfolding eventually_conj_iff **using** Aux_1_1 eventually_gt_at_top **by** blast

then have " $\forall^\infty k. \text{RN } k \ k \leq (2^{\text{powr } (2-\text{delta}'))} \wedge k$ "

proof (eventually_elim)

case (elim k)

```

then have "log 2 (RN k k) ≤ (2-delta') * k"
  by (meson of_nat_0_less_iff pos_divide_le_eq)
then have "RN k k ≤ 2 powr ((2-delta') * k)"
  by (smt (verit, best) Transcendental.log_le_iff powr_ge_zero)
then show "RN k k ≤ (2 powr (2-delta')) ^ k"
  by (simp add: mult.commute powr_power)
qed
moreover have "2 powr (2-delta') ≤ 4 - ?ε"
  unfolding delta'_def by (approximation 25)
ultimately show "∀∞k. real (RN k k) ≤ (4-?ε) ^ k"
  by (smt (verit) power_mono powr_ge_zero eventually_mono)
qed auto

```

A particular advantage of Isabelle is that proofs, even really big ones, are easy to modify. The structured proof language localises the points at which proofs fail, and the strong automation can often correct those failures with a mouse click. My many mistakes sometimes required fundamental changes, which could be pushed through easily enough. Once the first version of the proof was done, I noticed that k , ℓ , μ and their associated constraints could be moved into the locales (as opposed to being explicit arguments throughout), yielding a transformative improvement in readability.

Bhavik's Lean proof is slightly longer. He adopted a different fix to the errors in theorems 9.1 and 10.1 and to the issues in sections 11 and 12. Lacking the **approximation** and **real_asymp** tools, he obtained the required results manually. The splitting example above cost him “over 200 lines of analytic arguments, and at least 150 lines of interval arithmetic calculations” (private communication). Because he used filters rather than bigness predicates, a lot of context had to be repeated many times. Given all that, his proof is admirably concise.

What do we learn from this project? Mainly, it provides more evidence that proof assistants – including Isabelle/HOL – can cope with serious new mathematics. Although the techniques in the paper hardly go beyond the undergraduate level, it is an intricate construction combining probability, graph theory, asymptotics and high precision numerical calculations. It is the sort of proof that would surely attract doubters, but with two largely independent formalisations, in different systems, there can be little room for doubt anymore.

With so much mathematics formalised, both in both Lean and in Isabelle [15], it's reasonable to say that the point has been made. But still: so-called proof assistants should actually be *assisting* mathematicians in their work, as they already assist computer scientists. One line of attack would begin with the Isabelle proof just finished. The diagonal paper is full of seemingly arbitrary numerical choices. The Isabelle proof is a living document that you can tinker with, making a change and immediately seeing what goes wrong. Somebody who knows what they are doing could surely find a sharper bound or a simpler proof.

References

- 1 Vigleik Angeltveit and Brendan D. McKay. $R(5, 5) \leq 46$, 2024. URL: <https://arxiv.org/abs/2409.15709>.
- 2 Clemens Ballarin. Exploring the structure of an algebra text with locales. *Journal of Automated Reasoning*, 64(6):1093–1121, 2020. doi:10.1007/s10817-019-09537-9.
- 3 Clemens Ballarin and Lawrence C. Paulson. A pragmatic approach to extending provers by computer algebra — with applications to coding theory. *Fundamenta Informaticae*, 39:1–20, 1999. doi:10.3233/FI-1999-391201.

- 4 Marcelo Campos, Simon Griffiths, Robert Morris, and Julian Sahasrabudhe. An exponential improvement for diagonal Ramsey, 2023. URL: <https://arxiv.org/abs/2303.09521>.
- 5 Manuel Eberl. Verified real asymptotics in Isabelle/HOL. In James Davenport and Dongming Wang, editors, *International Symposium on Symbolic and Algebraic Computation (ISAAC 2019)*, pages 147–154. ACM, 2019. doi:10.1145/3326229.3326240.
- 6 Chelsea Edmonds. Undirected graph theory. *Archive of Formal Proofs*, September 2022. , Formal proof development. URL: https://isa-afp.org/entries/Undirected_Graph_Theory.html.
- 7 Chelsea Edmonds, Angeliki Koutsoukou-Argyaki, and Lawrence C. Paulson. Formalising Szemerédi’s regularity lemma and Roth’s theorem on arithmetic progressions in Isabelle/HOL. *Journal of Automated Reasoning*, 67(1), 2023. doi:10.1007/s10817-022-09650-2.
- 8 Chelsea Edmonds and Lawrence C. Paulson. Formal probabilistic methods for combinatorial structures using the Lovász local lemma. In Brigitte Pientka and Sandrine Blazy, editors, *13th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2024, pages 132–146. Association for Computing Machinery, 2024. doi:10.1145/3636501.3636946.
- 9 Thibault Gauthier and Chad E. Brown. A formal proof of $R(4,5)=25$. In Yves Bertot, Temur Kutsia, and Michael Norrish, editors, *15th International Conference on Interactive Theorem Proving (ITP 2024)*, volume 309 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITP.2024.16.
- 10 Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O’Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. A machine-checked proof of the odd order theorem. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving*, LNCS 7998, pages 163–179. Springer, 2013. doi:10.1007/978-3-642-39634-2_14.
- 11 Charles M Grinstead and Sam M Roberts. On the Ramsey numbers $R(3, 8)$ and $R(3, 9)$. *Journal of Combinatorial Theory, Series B*, 33(1):27–51, 1982. doi:10.1016/0095-8956(82)90055-7.
- 12 Johannes Hölzl. Proving inequalities over reals with computation in Isabelle/HOL. In Gabriel Dos Reis and Laurent Théry, editors, *ACM SIGSAM 2009 International Workshop on Programming Languages for Mechanized Mathematics Systems (PLMMS’09)*, pages 38–45, 2009.
- 13 Johannes Hölzl, Fabian Immler, and Brian Huffman. Type classes and filters for mathematical analysis in Isabelle/HOL. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *ITP*, LNCS 7998, pages 279–294. Springer, 2013. doi:10.1007/978-3-642-39634-2_21.
- 14 Tobias Nipkow. A tutorial introduction to structured Isar proofs. Online documentation. URL: <http://isabelle.in.tum.de/dist/Isabelle/doc/isar-overview.pdf>.
- 15 Lawrence C. Paulson. Large-scale formal proof for the working mathematician—lessons learnt from the ALEXANDRIA project. In *Intelligent Computer Mathematics (CICM 2023)*, pages 3–15. Springer, 2023. doi:10.1007/978-3-031-42753-4_1.
- 16 Lawrence C. Paulson. An exponential improvement for diagonal Ramsey. *Archive of Formal Proofs*, September 2024. , Formal proof development. URL: https://isa-afp.org/entries/Diagonal_Ramsey.html.
- 17 Lawrence C. Paulson. A formalised theorem in the partition calculus. *Annals of Pure and Applied Logic*, 175(1, Part B):103246, 2024. Kenneth Kunen (1943-2020). doi:10.1016/j.apal.2023.103246.
- 18 Clark Weissman. *Lisp 1.5 Primer*. Dickenson Publishing Company, 1967.