$R(3,10) \leqslant 41$

Vigleik Angeltveit

Submitted: May 14, 2024; Accepted: Apr 29, 2025; Published: Nov 3, 2025

Abstract

We improve the upper bound on the Ramsey number $R(3,10)$ from 42 to 41. Hence $R(3,10)$ is equal to 40 or 41.

Mathematics Subject Classifications: 05D10

1 Introduction

The Ramsey number $R(s,t)$ is defined to be the smallest n such that every graph of order n contains either a clique of s vertices or an independent set of t vertices. See [5] for a survey on the currently known bounds for small Ramsey numbers. The smallest Ramsey numbers that are currently unknown are $R(4,6)$ and $R(3,10)$.

In this paper we prove the following result:

Theorem 1. *The Ramsey number $R(3,10)$ is less than or equal to 41.*

Let $\mathcal{R}(s,t,n)$ denote the set of isomorphism classes of Ramsey graphs of type (s,t) with n vertices. Similarly, let $\mathcal{R}(s,t,n,e \leqslant e_0)$ denote the set of such graphs with at most e_0 edges. In [1, Theorem 1], Exoo proved that $R(3,10) \geqslant 40$ by finding an explicit graph in $\mathcal{R}(3,10,39)$. Combined with Theorem 1 this means that $R(3,10)$ is either 40 or 41. The proof of Theorem 1 uses extensive computer calculations, and can be thought of as a follow-up to [2] where Goedgebeur and Radziszowski proved that $R(3,10) \leqslant 42$.

The project described in [2] took a total of about 50 CPU years, and lowering the upper bound on $R(3,10)$ from 42 to 41 requires several orders of magnitude more calculations. For example, Goedgebeur and Radziszowski had to consider about 80 million $\mathcal{R}(3,8)$-graphs and we had to consider approximately 150 billion such graphs. Despite this, we were able to complete the proof of Theorem 1 in about 3 CPU years, and all of the calculations were done on two standard desktop computers over a period of a few months.

To complete this project, the main challenge was to come up with algorithms that are several orders of magnitude faster than those used in [2] to go with the several orders of magnitude more graphs.

Mathematical Sciences Institute, Australian National University, Canberra, ACT 2601, Australia (`vigleik.angeltveit@anu.edu.au`)

2 Outline of proof

For a graph G and a vertex v in G, we write G_v^+ for the induced subgraph on the vertices adjacent to v and refer to this as the *neighbourhood* of v. Similarly, we write G_v^- for the induced subgraph on the vertices not adjacent to v and refer to this as the *dual neighbourhood* of v. Note that G_v^- does not include v as one of its vertices.

Given a hypothetical graph Γ in $\mathcal{R}(3, 10, 41)$, the basic idea is to consider a vertex v of degree d. Then the neighbourhood of v is an independent set of size d, while the dual neighbourhood is a graph $G \in \mathcal{R}(3, 9, 41 - d - 1)$. We can then use a version of the Neighbourhood Gluing Extension Method from [2] (which we review in Section 4.0) to reconstruct Γ, and it suffices to show that this kind of gluing does not produce any output.

Proposition 2. *Any graph Γ in $\mathcal{R}(3, 10, 41)$ must have a vertex v with dual neighbourhood in one of the following sets:*

1. $\mathcal{R}(3, 9, 32, e \leqslant 112)$
2. $\mathcal{R}(3, 9, 33, e \leqslant 121)$
3. $\mathcal{R}(3, 9, 34, e \leqslant 130)$

What the proof will show is that such a graph Γ must have a vertex v with dual neighbourhood in one of those sets or in $\mathcal{R}(3, 9, 35, e \leqslant 139)$. We note that by [2, Table 14] the set $\mathcal{R}(3, 9, 35, e \leqslant 139)$ is empty, since there is a unique $\mathcal{R}(3, 9, 35)$-graph and it has 140 edges, so we do not include it in the statement.

Proof. Since $R(3, 9) = 36$, every vertex of Γ must have degree in $\{5, 6, 7, 8, 9\}$. Because 41 is odd, Γ cannot be regular of degree 9 and must have at least one vertex of degree in $\{5, 6, 7, 8\}$.

This result makes intuitive sense, since if Γ has 40 vertices of degree 9 and a single vertex v of degree 8 then Γ has a total of $\frac{40 \cdot 9 + 1 \cdot 8}{2} = 184$ edges. After removing the degree 8 vertex v and its 8 neighbours we are left with $184 - 8 \cdot 9 = 112$ edges, so the dual neighbourhood of v is in $\mathcal{R}(3, 9, 32, e = 112)$. We want to argue that this is the worst-case scenario: If we start with some Γ with fewer than 184 edges then we still have a vertex whose dual neighbourhood is sparse.

Let n_i denote the number of vertices of degree i in Γ. First suppose $n_5 = n_6 = n_7 = 0$, so every vertex of Γ has degree 8 or 9. Then Γ has $\frac{369 - n_8}{2}$ edges, and the dual neighbourhood of a degree 8 vertex v has $\frac{369 - n_8}{2} - \sum_{i=1}^{8} d(w_i)$ edges, where the sum is over the 8 vertices in Γ adjacent to v. Because the induced subgraph of Γ on the degree 8 vertices is triangle-free and n_8 is odd, one of the degree 8 vertices is adjacent to at most $\frac{n_8 - 1}{2}$ other degree 8 vertices. Hence the sum of the degree of the neighbours of v is at least $72 - \frac{n_8 - 1}{2}$.

It follows that the number of edges in the dual neighbourhood Γ_v^- of that degree 8 vertex v is

$$e(\Gamma_v^-) = e(\Gamma) - \sum_{i=1}^{8} d(w_i) \leqslant \frac{369 - n_8}{2} - (72 - \frac{n_8 - 1}{2}) = 112.$$

Hence we have established that if every vertex of Γ has degree 8 or 9 then the dual neighbourhood of some vertex of Γ is in $\mathcal{R}(3, 9, 32, e \leqslant 112)$.

Now we consider the general case. First note that Γ has $\frac{369-N}{2}$ edges, where $N = n_8 + 2n_7 + 3n_6 + 4n_5$. Let $nb_n(v)$ denote the number of neighbours of v of degree n in Γ. For a vertex v of degree at most 8, let

$$\epsilon(v) = nb_8(v) + 2nb_7(v) + 3nb_6(v) + 4nb_5(v).$$

Then the dual neighbourhood of v is in one of the sets in the lemma if and only if $\epsilon(v) \leqslant \frac{N-1}{2}$. To see that, we assume that $\epsilon(v) \leqslant \frac{N-1}{2}$ and compute

$$\begin{aligned} e(\Gamma_v^-) = e(\Gamma) - \sum d(w_i) = \frac{369 - N}{2} - 9d(v) + \epsilon(v) \\ \leqslant \frac{369 - N}{2} - 9d(v) + \frac{N-1}{2} = 184 - 9d(v), \end{aligned}$$

where once again the sum is over the neighbours of v and we note that $184 - 9d(v)$ gives us exactly the numbers in the proposition.

Hence it suffices to prove that there exists a vertex v of degree at most 8 with $\epsilon(v) \leqslant \frac{N-1}{2}$. We can compute $\epsilon(v)$ as the degree of a vertex in another graph $\widetilde{\Gamma}$ of order N constructed as follows: For each vertex v of Γ, the graph $\widetilde{\Gamma}$ has $9 - d(v)$ distinct vertices, and two vertices of $\widetilde{\Gamma}$ are adjacent if and only if the corresponding vertices in Γ are adjacent. In other words, delete the degree 9 vertices, keep the degree 8 vertices, keep two copies of the degree 7 vertices, and so on. Then $\epsilon(v)$ is precisely the degree of any of the vertices in $\widetilde{\Gamma}$ corresponding to v, since a single neighbour contributing k to $\epsilon(v)$ has been replaced by k neighbours. This process preserves the property of being triangle-free, so $\widetilde{\Gamma}$ is a triangle-free graph of order N. As such, it must have a vertex of degree at most $\frac{N-1}{2}$, and the result follows. $\square$

Corollary 3. *To prove Theorem 1, it suffices to glue the graphs in Proposition 2 using the Neighbourhood Gluing Extension Method.*

This leaves us with two things to do:

1. Generate all the graphs in Proposition 2.
2. Show that none of them extend to an $\mathcal{R}(3, 10, 41)$-graph.

3 A census of graphs

To generate the graphs in Proposition 2, we start with a partial census of $\mathcal{R}(3, 7)$ and work our way up.

3.1 A partial census of $\mathcal{R}(3,7)$

We used the following sets of graphs:

1. $\mathcal{R}(3,7,16,e \leqslant 24)$ (46 514 graphs)

2. $\mathcal{R}(3,7,17,e \leqslant 30)$ (3 131 580 graphs)

3. $\mathcal{R}(3,7,18,e \leqslant 36)$ (23 149 358 graphs)

4. $\mathcal{R}(3,7,19,e \leqslant 41)$ (2 173 527 graphs)

5. $\mathcal{R}(3,7,20,e \leqslant 46)$ (10 613 graphs)

These are not that difficult to compute, and we computed them using a one-point extender that takes as input an $\mathcal{R}(3,7,n)$-graph and outputs all ways to add a single vertex to produce an $\mathcal{R}(3,7,n+1)$-graph. Our results agree with those in [2, Table 12], and we extend their counts for $\mathcal{R}(3,7,18,e \leqslant e_0)$ from a maximum of $e_0 = 35$ to 36.

It is possible use the much smaller set $\mathcal{R}(3,7,16,e \leqslant 21)$ rather than $\mathcal{R}(3,7,16,e \leqslant 24)$ to generate the necessary $\mathcal{R}(3,8)$-graphs, but see Section 5 for a place where we found it convenient to use the larger set of graphs.

3.2 A partial census of $\mathcal{R}(3,8)$

We need the following sets of graphs:

1. $\mathcal{R}(3,8,23,e \leqslant 53)$ (238 854 716 graphs)

2. $\mathcal{R}(3,8,24,e \leqslant 63)$ (approximately 150 billion graphs)

3. $\mathcal{R}(3,8,25,e \leqslant 70)$ (2 120 846 970 graphs)

4. $\mathcal{R}(3,8,26,e \leqslant 77)$ (1 767 543 graphs)

5. $\mathcal{R}(3,8,27)$ (477 142 graphs)

While we do not have an accurate count of $|\mathcal{R}(3,8,24,e \leqslant 63)|$ we do have that $|\mathcal{R}(3,8,24,e \leqslant 62)| = 14\,645\,288\,701$. See Section 5 below for details on how to deal with $\mathcal{R}(3,8,24,e = 63)$. Our results agree with those in [2, Table 13], and we extend their counts for $\mathcal{R}(3,8,23,e \leqslant e_0)$ from $e_0 \leqslant 52$ to $e_0 \leqslant 53$ and for $\mathcal{R}(3,8,24,e \leqslant e_0)$ from $e_0 \leqslant 60$ to $e_0 \leqslant 62$.

The graphs in $\mathcal{R}(3,8,27)$ are not logically necessary, but we found it convenient to glue them as well in order to make an additional assumption about the minimum degree of any vertex for the remaining gluing operations.

In each case it is a small linear programming exercise to show that gluing the above $\mathcal{R}(3,7)$-graphs suffices to generate these $\mathcal{R}(3,8)$-graphs. These linear programming arguments are all similar, so we do one in detail. Recall e.g. from [4, Theorem 2.2] that for any graph G of order n we have

$$\sum 2e(G_v^-) = \sum \big(d(v)(n-2d(v)) + 2e(G_v^+)\big).$$

Here the sum is over the vertices of G.

In the case of a triangle-free graph, $e(G_v^+) = 0$. Let $\bar{d} = \frac{2e(G)}{n}$ be the average degree of a vertex in G. Since $\sum(\bar{d} - d(v)) = 0$, we can add any multiple of that to each term. In other words, for any constant c we have

$$\sum 2e(G_v^-) = \sum \big(d(v)(n - 2d(v)) + 2c(\bar{d} - d(v))\big).$$

For example, consider some $G \in \mathcal{R}(3, 8, 24, e = 63)$. Then this simplifies to

$$\sum \big(e(G_v^-) - \big(d(v)(12 - d(v)) + c(5.25 - d(v))\big)\big) = 0.$$

Now we choose an appropriate value of c, and prove that if G_v^- is not in one of the above sets of $\mathcal{R}(3, 7)$-graphs (because $e(G_v^-)$ is too large) then the corresponding term is positive. Since not all the terms can be positive, some G_v^- has to be in one of those sets of graphs.

In this case we can choose $c = 7.5$. Then this shows that for at least one vertex we must have G_v^- in one of $\mathcal{R}(3, 7, 16, e \leqslant 21.875)$, $\mathcal{R}(3, 7, 17, e \leqslant 30.375)$, $\mathcal{R}(3, 7, 18, e \leqslant 36.875)$ or $\mathcal{R}(3, 7, 19, e \leqslant 41.375)$. (We also get $\mathcal{R}(3, 7, 20, e \leqslant 43.875)$, $\mathcal{R}(3, 7, 21, e \leqslant 44.375)$ or $\mathcal{R}(3, 7, 22, e \leqslant 42.875)$, but those are all empty.)

3.3 A partial census of $\mathcal{R}(3, 9)$

As explained in Proposition 2 above, we need the following sets of graphs

1. $\mathcal{R}(3, 9, 32, e \leqslant 112)$ (1 554 928 360 graphs)
2. $\mathcal{R}(3, 9, 33, e \leqslant 121)$ (14 395 graphs)
3. $\mathcal{R}(3, 9, 34, e \leqslant 130)$ (5 graphs)

Our results agree with those in [2, Table 14], and here we extend their counts for $\mathcal{R}(3, 9, 32, e \leqslant e_0)$ from $e_0 \leqslant 108$ to $e_0 \leqslant 112$ and for $\mathcal{R}(3, 9, 33, e \leqslant e_0)$ from $e_0 \leqslant 119$ to $e_0 \leqslant 121$. The same table says that they found $|\mathcal{R}(3, 9, 33, e \leqslant 121)| \geqslant 14\,378$, so they were missing just 17 such graphs.

In addition we considered the single graph in $\mathcal{R}(3, 9, 35)$. This is not logically necessary, but we found it convenient to glue it in order to make an additional assumption about the minimum degree of any vertex for the remaining gluing operations.

Again, in each case it is a small linear programming exercise to show that gluing the above $\mathcal{R}(3, 8)$-graphs suffices to generate these $\mathcal{R}(3, 9)$-graphs.

We remark that $\mathcal{R}(3, 8, 24, e = 63)$ is only needed to generate the regular degree 7 graphs in $\mathcal{R}(3, 9, 32, e = 112)$. For all the remaining graphs it suffices to consider $\mathcal{R}(3, 8, 24, e \leqslant 62)$. We explain how to generate those regular degree 7 graphs in Section 5.

3.4 Checking the data

For each of the sets of graphs described above, we checked for completeness by deleting a single vertex from each graph in all possible ways and then using a one-point extender. No additional graphs were found. For the smaller sets we also deleted multiple vertices in all possible ways and used a one-point extender repeatedly, and once again no additional graphs were found. We did the same with a random sample of the larger sets of graphs.

We also compared the output of the gluing algorithm described in Section 4, with and without each of the modifications listed in Subsections 4.1 through 4.4 (except that for the largest graphs we did not try using a precomputed table of independence numbers), on a random sample of graphs to verify that these optimisations had no effect other than on the running time.

This is in addition to the comparisons to [2, Table 12, 13, 14] and the additional checks of $\mathcal{R}(3, 9, 32, e \leqslant 112)$ by Brendan McKay.

4 Gluing algorithms

The main algorithm used by Goedgebeur and Radziszowski is the Neighbourhood Gluing Extension Method. Similar algorithms are pervasive in the subject. We made a few modifications to increase the performance.

4.0 The Neighbourhood Gluing Extension Method

The basic algorithm is as follows. Given $G' \in \mathcal{R}(3, t, n)$ and some natural number d, one can compute all graphs $G \in \mathcal{R}(3, t+1, n+1+d)$ or $G \in \mathcal{R}(3, t+1, n+1+d, e \leqslant e_0)$ with a vertex v of degree d with neighbours $v_1, \ldots, v_d$ and dual neighbourhood G' as follows. First, make a list of all independent sets in G'. If S is an independent set in G' then assigning $S \cup \{v\}$ as the set of neighbours of some v_i does not create any triangles, and the only thing to check is if assigning $S_1, \ldots, S_d$ (and v) as the set of neighbours to $v_1, \ldots, v_d$ creates any independent $(t+1)$-sets.

We say the *independence number* of a subset $W \subset V(G')$ is the size of the largest independent set with vertices in W. Then the above assignment is a valid assignment if and only if for each subset $K \subseteq \{1, \ldots, d\}$ the independence number of $V(G') \setminus \bigcup_{k \in K} S_k$ is at most $t - |K|$. (This is automatic for $|K| = 1$.) If so, we call $S_1, \ldots, S_d$ *compatible.* Based on this one can implement an inductive search algorithm, where given a list $(S_1, \ldots, S_i)$ of compatible independent sets one checks each independent set S to see if $(S_1, \ldots, S_i, S)$ is also compatible.

Since the isomorphism class of the resulting graph in $\mathcal{R}(3, t+1, n+1+d)$ is independent of the ordering of $v_1, \ldots, v_d$, and hence by the ordering of $S_1, \ldots, S_d$, we can order the independent sets in some way and insist that $(S_1, \ldots, S_d)$ appear in non-increasing order. Suppose $S_1, \ldots, S_N$ is a complete list of independent sets in G' (which we can precompute). A basic version of this algorithm which does not impose a restriction on the number of edges is described in Algorithm 1. This algorithm assumes that we already have another algorithm `compatible` which checks if the sets in a list $(S_1, \ldots, S_i)$ are compatible.

Algorithm 1 `NGEM`(list $(i_1, \ldots, i_k)$ of indices)

if $k = d$ **then**
  Output graph G with $V(G) = V(G') \cup \{v, v_1, \ldots, v_d\}$ where v is adjacent to $\{v_1, \ldots, v_d\}$ and v_k is adjacent to $S_{i_k} \cup \{v\}$ for each $k = 1, \ldots, d$.
else
  for $i_{k+1} \in \{1, \ldots, i_k\}$ **do**
    if `compatible`$(S_{i_1}, \ldots, S_{i_{k+1}})$ **then**
      `NGEM`$(i_1, \ldots, i_{k+1})$
    end if
  end for
end if

A version of this algorithm is given in [2, Algorithm 2]. By ordering the independent sets by size one can also prune by checking when adding independent sets from earlier in the list will produce a graph with too many edges.

We used their algorithm without change to compute the parts of $\mathcal{R}(3, 8, n)$ we needed, but we made some changes when computing $\mathcal{R}(3, 9, n)$ and when showing that $\mathcal{R}(3, 10, 41)$ is empty.

4.1 Our first modification: Using only maximal independent sets

This modification is responsible for the largest performance increase. To describe it, we use the following result:

Lemma 4. *If the independent sets $S_1, \ldots, S_d$ are compatible and $S_i \subseteq S'_i$ for each i then the independent sets $S'_1, \ldots, S'_d$ are also compatible.*

In other words, if we can add an edge from v_i to G' without introducing any triangles then we can add that same edge to G without introducing any triangles. Adding an edge does not introduce any additional independent sets, so the lemma follows. Hence it suffices to consider independent sets that are *maximal* in the sense that they are not contained in any larger independent sets. This has a massive advantage. In a typical case this reduces the number of independent sets by an order of magnitude, and the number of tuples by several orders of magnitude.

This modification also has some disadvantages. The first is that we can no longer prune by number of edges, at least not at this stage of the algorithm.

The second downside is that such a maximal solution might represent a large number of non-maximal solutions. So we need another algorithm to extract those non-maximal solutions.

We do this as follows: Before searching for compatible d-tuples of maximal independent sets, assign each independent set S to a maximal independent set S' with $S \subset S'$. This way each maximal independent set comes with an allowed list of subsets.

Given compatible maximal independent sets $S'_1, \ldots, S'_d$, we inductively try replacing S'_i by each allowed subset S_i while pruning on the total number of edges.

4.2 Our second modification: Not using a precomputed table of independence numbers

Goedgebeur and Radziszowski used a precomputed table of the independence number of any subset of $V(G')$. Precomputing this simply takes too long. If the number of vertices is small (at most 27) we initialise a table of independence numbers to 0 and compute independence numbers as needed. If the number of vertices is large, we use a HashMap instead.

Note that there is a fast way to check if a pair of independent sets is compatible without using a precomputed table of independence numbers: For each independent set S we precompute which independent $(t-1)$-sets are contained in $V(G') \setminus S$ and store the result in a bitvector. Then we can use a bitwise *and* operation to check if S_i and S_j are compatible. We then store this result for each pair in another bitvector, so we can check with another bitwise *and* operation which independent sets are pairwise compatible with $S_1, \ldots, S_k$.

4.3 Our third modification: Computing independence numbers as late as possible

Suppose, for example, that we start with $G' \in \mathcal{R}(3, 8, 24)$ and $d = 7$, and that we are looking for graphs in $\mathcal{R}(3, 9, 32, e \leqslant 112)$. Then we are looking for compatible independent 7-tuples $S_1, \ldots, S_7$ of maximal independent sets. Because bitwise operations are fast and computing the independence number of some subset of $V(G')$ is slow, we check for pairwise compatibility only (which we compute once and then store in a bitvector) until we find a potential solution $S_1, \ldots, S_7$. Only at this point do we start computing independence numbers, starting with that of $V(G') \setminus (S_1 \cup S_2 \cup S_3)$. If we do find that $\{S_1, S_2, S_3\}$ are incompatible, we can jump straight to the next assignment of S_3.

4.4 Our fourth modification: Ordering the independent sets

We order the maximal independent sets $S_1, \ldots, S_N$ as follows: First, we find the independent $(t-1)$-set which is contained in $V(G') \setminus S_i$ for the largest number of maximal independent sets, and consider those maximal independent sets last. The reason for doing so is that none of them are compatible with each other, so we can choose at most one of them and they only need to be considered when choosing the last independent set S_{i_d}. Hence we can work with a smaller collection of independent sets until the tail end of the recursive search (or, as we implemented it, in the inner-most loop). This can be repeated.

When aiming for $\mathcal{R}(3, 10, 41)$ this gives us an especially large saving, as we almost never have to consider the last 3 or 4 subsets of maximal independent sets.

4.5 Performance

For example, these modifications allow us to glue approximately 250 graphs in $\mathcal{R}(3,8,24)$ per second (per CPU core) on a modern computer to produce graphs in $\mathcal{R}(3,9,32,e \leqslant 112)$, and a similar number of graphs in $\mathcal{R}(3,9,32)$ per second to produce (hypothetical) graphs in $\mathcal{R}(3,10,41)$.

5 The special case of extending the set $\mathcal{R}(3,8,24,e=63)$ to $\mathcal{R}(3,9,32,e=112)$

There are approximately 150 billion graphs in $\mathcal{R}(3,8,24,e=63)$, and these are needed to find the regular degree 7 graphs in $\mathcal{R}(3,9,32,e=112)$ only. With the obvious algorithm this would take much longer than any of the other calculations, but we can take advantage of the fact that we only need them to find the regular degree 7 graphs to produce a much more efficient algorithm.

If $\Gamma \in \mathcal{R}(3,9,32,e=112)$ is regular of degree 7 then *every* vertex of Γ has dual neighbourhood in $\mathcal{R}(3,8,24,e=63)$.

Suppose in addition that the dual neighbourhood of some vertex v of Γ in $\mathcal{R}(3,8,24,e=63)$ has a degree 7 vertex w. That means the neighbourhoods of v and w in Γ are disjoint, and the intersection of the dual neighbourhoods of v and w is a graph $G' \in \mathcal{R}(3,7,16)$. Hence Γ is given by gluing two graphs $G_1, G_2 \in \mathcal{R}(3,8,24,e=112)$ along $G' \in \mathcal{R}(3,7,16)$.

For a fixed $G' \in \mathcal{R}(3,7,16)$ we first find all $G \in \mathcal{R}(3,8,24,e=63)$ with G' as a dual neighbourhood using the usual Neighbourhood Gluing Extension Method. For each pair $G_1, G_2 \in \mathcal{R}(3,8,24,e=63)$ intersecting in G' we can then compute the degree of each vertex $x \in V(G')$ considered in Γ, and each must have degree exactly 7. That is a very strong condition, so very few pairs are compatible. For a fixed G' there might be several hundred million extensions to $\mathcal{R}(3,8,24,e=63)$, so that leaves approximately 10^{17} pairs in that case. To avoid having to check every pair, we put the graphs in a HashMap based on the degree sequence of the vertices in $V(G')$ considered in G. Then, for each $G_1 \in \mathcal{R}(3,8,24,e=63)$ extending G' we can simply look up the compatible G_2 in the HashMap.

After finding a pair that is compatible, we do the following: For each neighbour v_i of v and neighbour w_j of w, we add an edge between v_i and w_j if this does not introduce a triangle. This can be checked with a single bitwise *and* for each pair (v_i, w_j). If every vertex now has degree at least 7, we check if this is an $R(3,9,32)$-graph. If it is, we then remove edges in all possible ways between degree $\geqslant 8$ vertices and record the $R(3,9,32)$-graphs we get.

A similar algorithm works for $G' \in \mathcal{R}(3,7,17)$ or $G' \in \mathcal{R}(3,7,18)$. The restriction on the degree of the vertices in G' considered in Γ is weaker, but we also have fewer pairs.

Because this has to work for *every neighbourhood*, we can exclude any graph $G \in \mathcal{R}(3,8,24,e=63)$ that has already been considered. It is possible to do this without storing $\mathcal{R}(3,8,24,e=63)$, for example as follows:

We first ran this program for $G' \in \mathcal{R}(3,7,16,e=24)$. Next, we ran it for $G' \in$

$\mathcal{R}(3,7,17,e=30)$ while excluding any output in $\mathcal{R}(3,8,24,e=63)$ with some dual neighbourhood in $\mathcal{R}(3,7,16,e=24)$. Next we considered $\mathcal{R}(3,7,18,e=36)$ while excluding any output in $\mathcal{R}(3,7,16,e=24)$ or $\mathcal{R}(3,7,17,e=30)$. After that we considered $\mathcal{R}(3,7,16,e=23)$ while excluding the graphs with a dual neighbourhood in one of the previous cases, and so on until covering all of $\mathcal{R}(3,7,16,e\leqslant 24)$, $\mathcal{R}(3,7,17,e\leqslant 30)$ and $\mathcal{R}(3,7,18,e\leqslant 36)$. Finally we also considered $\mathcal{R}(3,7,19,e\leqslant 40)$ (which is small and only required a simple one-point extender.)

This produced a total of 506 regular degree 7 graphs in $\mathcal{R}(3,9,32,e=112)$, in about 6 months of CPU time.

6 Extending the necessary $\mathcal{R}(3,9)$-graphs

In the end we had about 1.6 billion graphs in $\mathcal{R}(3,9,32,e\leqslant 112)$, 14 395 graphs in $\mathcal{R}(3,9,33,e\leqslant 121)$ and 5 graphs in $\mathcal{R}(3,9,34,e\leqslant 130)$. The latter two cases were very fast. The first case took about 3 months of CPU time to extend to $\mathcal{R}(3,10,41)$. Since this produced no outputs, this finishes the proof of Theorem 1.

Since running a program that produces no output is not very satisfying we had our program produce a large number of graphs in $\mathcal{R}(3,10,38)$ instead, and used a one-point extender to verify that none of them extended to $\mathcal{R}(3,10,41)$. In fact, none of the graphs we produced extended to $\mathcal{R}(3,10,40)$. (Otherwise this would have determined $R(3,10)$ exactly.)

7 A partial census of $\mathcal{R}(3,10,39)$

This project produced a large number of graphs in $\mathcal{R}(3,10,39)$, and we were able to produce many more by deleting a single vertex from each such graph in all possible ways and then using a one-point extender. In this way we produced 39 745 077 such graphs. Most of these had already been found by Goedgebeur and Radziszowski [2] who found 43 117 868 such graphs. By taking the union (and deleting a single vertex and using a one-point extender) we extended this to 43 146 537 graphs. By deleting two or three vertices and extending back up in all possible ways we further extended this to 43 225 483 graphs.

One might consider the set $\mathcal{R}(s,t,n)$ for $n=R(s,t)-1$ of *maximal* Ramsey graphs for various values of s and t. We see that $|\mathcal{R}(3,10,39)|$ is much larger than the number of maximal Ramsey graphs in other confirmed or conjectured cases:

The obvious conjecture is that $R(3,10)=40$. But because $\mathcal{R}(3,10,39)$ is so large, we are not confident in this prediction. If, indeed, $R(3,10)=40$ then this is going to be quite difficult to prove with current techniques. For example, a potential $\mathcal{R}(3,10,40)$-graph might be regular of degree 9 and then the dual neighbourhood of any vertex will be in $\mathcal{R}(3,9,30,e=99)$. This contains many orders of magnitude more graphs than $\mathcal{R}(3,9,32,e\leqslant 112)$.

Table 1: The number of maximal Ramsey graphs in some cases

$\lvert\mathcal{R}(3,5,13)\rvert$	1
$\lvert\mathcal{R}(3,6,17)\rvert$	7
$\lvert\mathcal{R}(3,7,22)\rvert$	191
$\lvert\mathcal{R}(3,8,27)\rvert$	477 142
$\lvert\mathcal{R}(3,9,35)\rvert$	1
$\lvert\mathcal{R}(4,4,17)\rvert$	1
$\lvert\mathcal{R}(4,5,24)\rvert$	352 366
$\lvert\mathcal{R}(4,6,35)\rvert$	$\geqslant 37$
$\lvert\mathcal{R}(5,5,42)\rvert$	$\geqslant 656$
$\lvert\mathcal{R}(3,10,39)\rvert$	$\geqslant 43\,225\,483$

Acknowledgements

The author would like to thank Brendan McKay for doing some sanity checks on some of the sets of graphs generated in this project and for commenting on an early draft. In particular, he spent some CPU time verifying that the census of $\mathcal{R}(3,9,32,e \leqslant 112)$ graphs is complete. I would also like to thank Jan Goedgebeur for some useful comments and for sharing a file with 43 117 868 $\mathcal{R}(3,10,39)$-graphs he and Radziszowski found in [2]. Finally I would like to thank the anonymous referees for their constructive comments.

This project also used the software package *nauty* extensively. See [3].

References

[1] Geoffrey Exoo. On two classical Ramsey numbers of the form $R(3,n)$. *SIAM J. Discrete Math.*, 2(4):488–490, 1989.

[2] Jan Goedgebeur and Stanisław P. Radziszowski. New computational upper bounds for Ramsey numbers $R(3,k)$. *Electron. J. Combin.*, 20(1):#P30, 2013.

[3] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symbolic Comput.*, 60:94–112, 2014.

[4] Brendan D. McKay and Stanisław P. Radziszowski. Subgraph counting identities and Ramsey numbers. *J. Combin. Theory Ser. B*, 69(2):193–209, 1997.

[5] Stanisław P. Radziszowski. Small Ramsey numbers, *Electron. J. Combin.*, #DS1. 1994–2021.