RAMSEY PROBLEMS IN MEGALODON AND PROOFGOLD

CHAD E. BROWN

1. Introduction

The finite version of Ramsey's Theorem states that for positive integers m, n there is some number K such that every graph with K vertices must have a clique of size m or an anticlique of size n. $R(m, n)$ is defined to be the least such K. The easiest nontrivial example is $R(3, 3) = 6$. The fact that $R(3, 3) > 5$ follows by considering the graph with 5 vertices shown in Figure 1. We encode this graph as a formal object to witness $R(3, 3) > 5$ in Section 3.

The fact that $R(3, 3) \leq 6$ can be proven by proving every graph with 6 vertices has the desired property. This can be done by considering various cases. We describe a formal proof of $R(3, 3) \leq 6$ in Section 4.

Typically the Ramsey property is presented in a way that allows for multiple colors of edges. Here we are considering the two-color case where an edge is "black" if there is an edge and "white" if there is no edge.

2. The Two Color Ramsey Property in HOTG

We will be working in Higher Order Tarski-Grothendieck (HOTG) axiomatized in Egal style. The details of this theory can be found in [1]. Here we will only need a handful of objects and known propositions.

Let ι be the type of sets and o be the type of propositions. In Megalodon ι is written as `set` and o is written as `prop`. Since HOTG is a classical extensional theory, there are two distinct propositions $\top, \bot : o$ where all propositions are either equal to $\top$ or equal to $\bot$. That is, o has exactly two elements. In general if σ and τ are types, $(\sigma \to \tau)$ is

Date: December 29, 2020.
Czech Technical University in Prague.

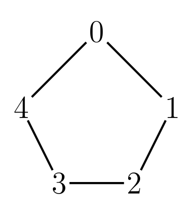

Figure 1. Graph demonstrating $R(3, 3) > 5$

also a type, intended to represent the type of functions from σ to τ. Here we only need to consider the following types in addition to ι and o:

- $\iota \to \iota$ is the type of functions from the universe of sets to itself. (Note that these are not functions from a set A to a set B encoded as sets. We could call functions of type $\iota \to \iota$ "operators" or "meta-functions" to avoid potential confusion, but we will simply say "functions" and rely on the reader not to be confused. It is common for functions of type $\iota \to \iota$ to restrict to be a function from a particular set A to a particular set B and we will see cases of this below.)
- $\iota \to o$ is the type of functions from sets to propositions. These functions can be thought of as predicates over sets, or (the characteristic functions of) classes of sets.
- $\iota \to \iota \to o$ (i.e., $\iota \to (\iota \to o)$) is the Curried form of a binary function from sets to propositions. These functions correspond to binary relations over the universe of sets. We will use objects of this type to represent the edge relation of a graph.

The basic terms of Megalodon and Proofgold include the following terms:

- If s is a term of type $\sigma \to \tau$ and t is a term of type σ, then (st) is a term of type τ.
- If x is a variable of type σ and t is a term of type τ, then $(\lambda x : \sigma.t)$ is a term of type $\sigma \to \tau$.
- If s and t are terms of type o, then $(s \to t)$ is a term of type o.
- If x is a variable of type σ and t is a term of type o, then $(\forall x : \sigma.t)$ is a term of type o.

We omit σ and simply write $(\lambda x.t)$ or $(\forall x.t)$ if σ is ι. Application associates to the left so that stu means $((st)u)$. Implication associates to the right so that $s \to t \to u$ means $(s \to (t \to u))$. Terms of type o are called *propositions*.

Two easy cases of objects of type $\iota \to \iota \to o$ are $\in$ and $\subseteq$. The membership relation $\in$ is primitive and satisfies certain set theory axioms. In Megalodon $\in$ is written in infix using ASCII as `:e`. The inclusion relation $\subseteq$ is defined by $\lambda AB.\forall x.x \in A \to x \in B$. In Megalodon this definition is declared as follows:

```
Definition Subq : set -> set -> prop := fun A B => forall x :e A, x :e B.
```

The inclusion relation is written in infix using ASCII as `c=`.

Although the basic term language only contains the logical connective $\to$ and quantifier $\forall$, the other logical connectives can be defined following the Russell-Prawitz style [5, 4]. In the cases of $\bot, \neg, \wedge, \vee$ we show the definitions in Megalodon below:

```
Definition False : prop := forall p:prop, p.
Definition not : prop -> prop := fun A:prop => A -> False.
Definition and : prop -> prop -> prop
 := fun A B:prop => forall p:prop, (A -> B -> p) -> p.
Definition or : prop -> prop -> prop
 := fun A B:prop => forall p:prop, (A -> p) -> (B -> p) -> p.
```

We will write $\wedge$ and $\vee$ in infix. In Megalodon the corresponding ASCII infix are `/\` and `\/`.

At each type σ we write $\exists x : \sigma.\varphi$ as notation for

$$\forall p : o.(\forall x : \sigma.\varphi \to p) \to p$$

where p is a variable not free in φ. Again, when σ is ι we leave it implicit.

At each type σ we write $s = t$ for $\forall q : \sigma \to \sigma \to o.qst \to qts$. This is a variant of Leibniz's equality used by Church [2]. We write `=` in ASCII in infix Megalodon for equality and `<>` for its negation ($\neq$).

We often use $\in$ and $\subseteq$ as restrictions on quantifiers. In particular, we adopt the following conventions:

- $\forall x \in A.\varphi$ means $\forall x.x \in A \to \varphi$,
- $\exists x \in A.\varphi$ means $\exists x.x \in A \wedge \varphi$,
- $\forall x \subseteq A.\varphi$ means $\forall x.x \subseteq A \to \varphi$ and
- $\exists x \subseteq A.\varphi$ means $\exists x.x \subseteq A \wedge \varphi$.

In order to say that set of vertices has a certain number of elements we will need natural numbers and a notion of equipotence (two sets having the same number of elements). We will follow the common convention of using finite ordinals as natural numbers.

The empty set $\emptyset$ of type ι is a primitive of the HOTG theory. The ordinal successor function ordsucc if type $\iota \to \iota$ is not a primitive of the theory but can be defined once one defines binary unions and a singleton operator. The idea is simply to define ordsucc so that the elements of ordsucc α are precisely the elements of α with the one additional element: α itself. Since this has already been defined in a document published into the Proofgold blockchain, we can simply declare ordsucc opaquely by giving a reference to its Merkle root. Technically there are two different Merkle roots: one for Megalodon and one for Proofgold. We must give both. In Megalodon the declaration looks as follows:

```
(* Parameter ordsucc
    "9db634daee7fc36315ddda5f5f694934869921e9c5f55e8b25c91c0a07c5cbec"
    "65d8837d7b0172ae830bed36c8407fcd41b7d875033d2284eb2df245b42295a6"
 *)
Parameter ordsucc : set->set.
```

The document only knows the type of ordsucc, but the Merkle root allows the document to be propertly integrated into the Proofgold chain when published.

We can now use natural numbers n as notation for ordsucc applied n times to $\emptyset$. That is, n is notation for a unary representation denoting the corresponding finite ordinal. In Megalodon the notation is declared as follows:

```
Notation Nat Empty ordsucc.
```

This unary representation of numbers is not realistic for large numbers, but we will only consider small numbers here. For numbers of the form 2^n we can give a different set with 2^n elements using the power set operator. The power set operator $\wp$ of type $\iota \to \iota$ is a primitive of the theory and axioms of the theory enforce that $\wp X$ is the set of subsets of X. If X is a finite set with n elements then $\wp X$ will be a finite set with 2^n elements.

The next step is to define what it means for two sets X and Y to be equipotent (have the same number of elements). The idea is simply to say X and Y are equipotent if there is a bijection from X to Y. In a first-order setting we would need a representation of functions as sets to make this definition, but in the higher-order setting we can make use of the higher type $\iota \to \iota$. In particular we say f of type $\iota \to \iota$ is *a bijection from X to Y* if f maps elements of X to elements of Y, f is injective on X and f maps X onto Y. In Megalodon the corresponding definition of bij looks as follows:

```
Definition bij : set->set->(set->set)->prop :=
  fun X Y f =>
  (forall u :e X, f u :e Y)
  /\
  (forall u v :e X, f u = f v -> u = v)
  /\
  (forall w :e Y, exists u :e X, f u = w).
```

We can then define X and Y to be *equipotent* if there exists such a bijection f.

```
Definition equip : set -> set -> prop
 := fun X Y : set => exists f : set -> set, bij X Y f.
```

This gives us enough infrastructure to define a tertiary relation between sets M, N and V recording whether or not the two-color Ramsey Property holds for cliques of size M and anticliques of size N when there are V vertices. The idea is to say that for every edge relation on the set V of vertices there is either a clique $X \subseteq V$ equipotent to M or an anticlique $Y \subseteq V$ equipotent to N. A set X is a clique if all distinct pairs of elements of X are edge-related. A set Y is an anticlique if no distinct pairs of elements of Y are edge-related. Since we are considering undirected graphs, we only quantify over symmetric relations. In Megalodon the definition looks as follows:

```
Definition TwoRamseyProp : set -> set -> set -> prop
 := fun M N V =>
 forall R:set -> set -> prop,
    (forall x y, R x y -> R y x)
   -> ((exists X c= V, equip M X /\ (forall x y :e X, x <> y -> R x y))
    \/ (exists Y c= V, equip N Y /\ (forall x y :e Y, x <> y -> ~R x y))).
```

We can now form the proposition $R(m, n) \leq K$ simply by writing TwoRamseyProp $m\ n\ K$. This relies on the fact that m has exactly m elements, n has exactly n elements and K has exactly K elements. Note that negating such a proposition corresponds $R(m, n) > K$. In case K is of the form 2^k we can write the (usually) shorter proposition TwoRamseyProp $m\ n\ (\wp\ k)$ for $R(m, n) \leq 2^k$ or its negation for $R(m, n) > 2^k$.

3. A Formal Proof of $R(3, 3) > 5$

In this section we describe two formal proofs of $R(3, 3) > 5$ in Megalodon. The two proofs are the same in essence, but the second version leads to a smaller Proofgold document. For both proofs we take some previously defined objects and previously proven propositions for granted. The two objects natp and ordinal have type $\iota \to o$ and are defined in ways such that we know each n satisfies natp and everything satisfying natp also satisfies ordinal.

The main previous result we will use is the following:

```
Axiom ordinal_equip_3_E_impred : forall X, equip 3 X ->
  (forall alpha :e X, ordinal alpha) ->
  forall p:prop,
    (forall alpha beta gamma :e X,
      alpha :e beta -> beta :e gamma ->
      (forall y :e X, forall q:set -> prop,
          q alpha -> q beta -> q gamma -> q y)
      -> p)
    -> p.
```

This means that if 3 and X are equipotent and every element of X is an ordinal, then there exist $\alpha, \beta, \gamma \in X$ such that $\alpha \in \beta$, $\beta \in \gamma$ and every element of X is either α, β or γ. This allows us to treat subsets of size three as being sorted by the strict ordering $\in$ (which is linear on ordinals). Considering the triples as ordered removes a number of symmetric cases.

We first state the main theorem in Megalodon.

```
Theorem not_TwoRamseyProp_3_3_5 : ~TwoRamseyProp 3 3 5.
```

Once negation is expanded $\neg\varphi$ is the same as $\varphi \to \bot$, so we begin the proof by assuming `TwoRamseyProp 3 3 5`. We label this assumption `H1`. The `assume` tactic makes this new assumption and associates it with the given label. It is optional whether or not to explicitly give the proposition being assumed and here we leave it implicit.

```
assume H1.
```

The current goal is now to prove $\bot$ (e.g., `False` in Megalodon). We can always make the current goal explicit using the `prove` tactic.

```
prove False.
```

In general it is an error when the current goal is not convertible to the proposition given with the `prove` tactic. The special case `prove False` never results in an error since it is possible to infer any goal by proving $\bot$. (There reader should be able to see this by inspecting the definition of $\bot$.)

Essentially we need to use the assumption `H1` to prove a contradiction. Inspecting the definition of `TwoRamseyProp` we see that we can use `H1` by applying it to a binary relation R and a proof that R is symmetric. We already know the edge relation we want to use is depicted in the graph from Figure 1. However, to communicate this edge relation to Megalodon we must write it as a term of type $\iota \to \iota \to o$. There may often be clever ways of giving this relation, but in this case we take the straightforward route of using disjunction, conjunction and equality to simply list all the cases. The term defining this relation will be relatively large so we use the `set` tactic to give it the short name R.

```
set R : set -> set -> prop := fun i j =>
     i = 0 /\ (j = 4 \/ j = 1)
  \/ i = 1 /\ (j = 0 \/ j = 2)
```

```
  \/ i = 2 /\ (j = 1 \/ j = 3)
  \/ i = 3 /\ (j = 2 \/ j = 4)
  \/ i = 4 /\ (j = 3 \/ j = 0).
```

We now prove a number of basic facts about R. For example we prove $R\,0\,4$. The proof will use the `prove` tactic (to expand $R\,0\,4$ into the proposition we wish to prove), the `apply` tactic and the `reflexivity` tactic. The `reflexivity` tactic solves a goal of the form $s = t$ when s and t are convertible. The `apply` tactic takes a proof term (in this case simply a reference to a previously proven proposition) and uses the proposition proven by the proof term to reduce the current goal to a number of subgoals.

As an aside, many of these tactics have very similar counterparts in the Coq system [3]. For example, `apply` and `reflexivity` are Coq tactics that behave in similar ways. Megalodon's `let` and `assume` tactics work like Coq's `intros` tactic. Megalodon's `prove` tactic is similar to Coq's `change` tactic. Although the systems have very different foundations, a tutorial on proving theorems in Coq is likely to help users learn to prove theorems in Megalodon.

We will use `apply` with three different previously proven propositions: `andI`, `orIL` and `or5I1`. We state these three propositions here and leave the reader to interpret their obvious meanings.

```
Axiom andI : forall A B:prop, A -> B -> A /\ B.
Axiom orIL : forall A B:prop, A -> A \/ B.
Axiom or5I1 : forall P1 P2 P3 P4 P5:prop,
                P1 -> P1 \/ P2 \/ P3 \/ P4 \/ P5.
```

Returning to the main proof we make a claim $R\,0\,4$, labeled `LRI04`, and then prove the claim with the proof delimited by `{` and `}`.

```
claim LRI04: R 0 4.
{ prove 0 = 0 /\ (4 = 4 \/ 4 = 1)
     \/ 0 = 1 /\ (4 = 0 \/ 4 = 2)
     \/ 0 = 2 /\ (4 = 1 \/ 4 = 3)
     \/ 0 = 3 /\ (4 = 2 \/ 4 = 4)
     \/ 0 = 4 /\ (4 = 3 \/ 4 = 0).
  apply or5I1. apply andI.
  - reflexivity.
  - apply orIL. reflexivity.
}
```

Note that applying `or5I1` results in one subgoal: $0 = 0 \wedge (4 = 4 \vee 4 = 1)$. Applying `andI` results in two subgoals: $0 = 0$ and $4 = 4 \vee 4 = 1$. We use bullet points `-` to indicate the start of the subproof of each subgoal (but only when there is more than one). There are three kinds of bullet points: `-`, `+` and `*`. These can be varied to make the levels of hierarchy of subproofs visible. The first subgoal $0 = 0$ is solved by `reflexivity`. The second subgoal $4 = 4 \vee 4 = 1$ is reduced to $4 = 4$ by applying `orIL` and then finished by `reflexivity`.

The proof then continues by making and proving claims for each of the nine other pairs where R holds (corresponding to each of the 5 edges in both directions). All the proofs are similar to the one above.

We next prove a general claim allowing us to use an assumption that $R\ i\ j$ holds:

```
claim LRE: forall i j, R i j -> forall p:set -> set -> prop,
       p 0 4 -> p 0 1 -> p 1 0 -> p 1 2 -> p 2 1 -> p 2 3
    -> p 3 2 -> p 3 4 -> p 4 3 -> p 4 0
    -> p i j.
```

This claim essentially states that the 10 pairs where R holds exhaust all the possibilities. So, if we know $R\ i\ j$ holds and we want to prove the goal $p\ i\ j$ holds we can apply `LRE` to reduce the goal to 10 subgoals corresponding to the 10 pairs. The proof of the claim begins by using the `let` tactic to fix i and j, reducing the goal to the proposition underneath the outer quantifiers. We then assume $R\ i\ j$, labeling the assumption `H`. We then let p be given and assume the ten specific cases.

```
{ let i j. assume H.
  let p. assume H04 H01 H10 H12 H21 H23 H32 H34 H43 H40.
```

The goal is now to prove $p\ i\ j$. Since R is defined by 5 disjunctions, we use the following previously proven `or5E` proposition.

```
Axiom or5E : forall P1 P2 P3 P4 P5:prop P1 \/ P2 \/ P3 \/ P4 \/ P5
 -> (forall p:prop,
         (P1 -> p)
      -> (P2 -> p)
      -> (P3 -> p)
      -> (P4 -> p)
      -> (P5 -> p)
     -> p).
```

In this case we do not simply write `apply or5E` but write the larger proof term with `or5E` applied to the five propositions used to define R and then to the assumption `H`:

```
  apply or5E (i = 0 /\ (j = 4 \/ j = 1))
             (i = 1 /\ (j = 0 \/ j = 2))
             (i = 2 /\ (j = 1 \/ j = 3))
             (i = 3 /\ (j = 2 \/ j = 4))
             (i = 4 /\ (j = 3 \/ j = 0)) H.
```

This has the effect of reducing the goal to five subgoals. The first subgoal is $(i = 0 \wedge (j = 4 \vee j = 1)) \rightarrow p\ i\ j$. We begin working on the first subgoal by assuming the conjunction, then applying the assumed conjunction.

```
  - assume H. apply H.
```

The goal is now $i = 0 \rightarrow (j = 4 \vee j = 1) \rightarrow p\ i\ j$. We assume $i = 0$ and then use the `rewrite` tactic.

```
    assume Hi: i = 0. rewrite Hi.
```

The goal is now $(j = 4 \vee j = 1) \rightarrow p\ 0\ j$. We next assume the disjunction and apply the assumed disjunction. This leads to two new subgoals $j = 4 \rightarrow p\ 0\ j$ and $j = 1 \rightarrow p\ 0\ j$. We can complete the first new subgoal by assuming $j = 4$, rewriting with the assumption and then using the `exact` tactic.

```
    + assume Hj: j = 4. rewrite Hj. exact H04.
```

The `exact` tactic requires a proof term that proves the current goal completely. In this case after `assume` and `rewrite` the current goal is $R\ 0\ 4$. Note that `H04` is the label for the assumption $R\ 0\ 4$ from the start of the proof of the claim. Hence `exact H04` completes this subproof. The remaining open subgoals are very similar and so we now assume the claim `LRE` has been proven.

Using `LRE` we can prove claims such as $\neg R\ 0\ 2$. The idea is simply to assume $R\ 0\ 2$ holds and then apply `LRE` with this assumption and an appropriate proposition.

```
claim LNR02: ~R 0 2.
{ assume H2.
  apply LRE 0 2 H2 (fun i j => i = 0 -> j = 2 -> False).
```

Before the `apply` the goal is `False`. After the `apply` there are 12 subgoals. Ten of these subgoals require proving $i = 0 \to j = 2 \to \bot$ for the ten values of (i, j) listed in `LRE`. The last two subgoals require proving $0 = 0$ and $2 = 2$ (both of which follow by reflexivity). Proving each of the first ten subgoals requires knowing the numbers 0, 1, 2, 3 and 4 are all (provably) different. There are previously proven results that allow one to easily prove each pair is different.

After proving `LRE` and the 10 claims enumerating when R holds for members of the ordinal 5, we can prove R is symmetric.

```
claim L2: forall i j, R i j -> R j i.
{ let i j. assume H.
```

The assumption `H` gives us that $R\ i\ j$ holds and our goal is to prove $R\ j\ i$ holds. We can use `LRE` to reduce the subgoal to checking 10 cases. Each case is proven by one of our previous claims. There is no need to use tactics to prove each subgoal and can instead give the exact proof term and complete the proof of the claim.

```
  exact LRE i j H
            (fun i j => R j i)
            LRI40 LRI10 LRI01 LRI21 LRI12 LRI32 LRI23 LRI43 LRI34 LRI04.
}
```

Finally we prove the most important claim. This claim states that every subset of 3 vertices is not a clique and not an anticlique.

```
claim L3: forall X c= 5, equip 3 X
  -> (exists x y :e X, x <> y /\ R x y)
  /\ (exists x y :e X, x <> y /\ ~R x y).
{ let X.
  assume HX1: X c= 5.
  assume HX2: equip 3 X.
```

Our goal is to prove

$$(\exists xy \in X, x \neq y \wedge R\ x\ y) \wedge (\exists xy \in X, x \neq y \wedge \neg R\ x\ y).$$

Since this proposition is relatively large, let us abbreviate it by P.

```
  set P : prop := (exists x y :e X, x <> y /\ R x y)
                /\ (exists x y :e X, x <> y /\ ~R x y).
```

Using previously proven results it is easy to prove that every member of X is a member of 5, hence a natural number and hence an ordinal.

```
claim LXo: forall x :e X, ordinal x.
{ let x. assume Hx: x :e X.
  ...
}
```

Using this claim we can apply our main assumed result above to obtain three ordinals $\alpha, \beta, \gamma \in X$ such that $\alpha \in \beta$, $\beta \in \gamma$ and α, β and γ exhaust X.

```
apply ordinal_equip_3_E_impred X HX2 LXo.
let alpha. assume Ha: alpha :e X.
let beta. assume Hb: beta :e X.
let gamma. assume Hc: gamma :e X.
assume Hab: alpha :e beta.
assume Hbc: beta :e gamma.
assume Hxc: forall y :e X, forall q:set -> prop,
                  q alpha -> q beta -> q gamma -> q y.
```

Since γ is in $X \subseteq 5$, $\gamma \in 5$.

```
claim Lc: gamma :e 5.
{ exact HX1 gamma Hc. }
```

The remainder of the proof of claim `L3` consists of splitting into cases for the value of $\gamma \in 5$ and the various possible values of β and α. The case splits are made possible by applying the following five previously proven lemmas:

```
Axiom cases_1: forall i :e 1, forall p:set->prop, p 0 -> p i.
Axiom cases_2: forall i :e 2, forall p:set->prop, p 0 -> p 1 -> p i.
Axiom cases_3: forall i :e 3, forall p:set->prop, p 0 -> p 1
                                                -> p 2 -> p i.
Axiom cases_4: forall i :e 4, forall p:set->prop, p 0 -> p 1
                                                -> p 2 -> p 3 -> p i.
Axiom cases_5: forall i :e 5, forall p:set->prop, p 0 -> p 1
                                          -> p 2 -> p 3 -> p 4 -> p i.
```

We complete the description of the proof of claim `L3` by summarizing the five cases for γ.

- $\gamma = 0$: Impossible since $\beta \in \gamma$.
- $\gamma = 1$: Impossible since $\alpha \in \beta$ and $\beta \in \gamma$.
- $\gamma = 2$: We must have $\alpha = 0$ and $\beta = 1$. $R\ 0\ 1$ and $\neg R\ 0\ 2$ hold.
- $\gamma = 3$: There are three possibilities for α and β and all are easy.
- $\gamma = 4$: There are six possibilities for α and β and all are easy.

After proving `L3` completing the proof of $\bot$ using the original assumption `H1` is straightforward. Applying `H1` to R and `L2` (the proof R is symmetric) yields two subcases: one in which there is a clique of size 3 and one in which there is an anticlique of size 3. Each case contradicts one of the conjuncts of the conclusion of `L3`.

After successfully completing the proof of the theorem, we would like to translate the Megalodon file into a Proofgold document to publish into the chain. Doing so with the proof described above leads to a document that would be roughly the Proofgold block size limit, so that the document would (probably) not fit into a block.

One reason the Proofgold document would be large is that the definition of R is repeated several times throughout the proof. We can make the document smaller by first proving a lemma giving the existence of a symmetric relation R satisfying the properties we want and then using that relation abstractly in the main proof. We describe this second approach briefly.

We first prove the following theorem stating a relation R with the desired properties exists.

```
Theorem exists_TwoRamseyProp_3_3_5 : exists R:set -> set -> prop,
  forall p:prop,
    ((forall i j, R i j -> R j i)
     -> R 0 1 -> R 1 2 -> R 2 3 -> R 3 4 -> R 4 0
     -> ~R 0 2 -> ~R 0 3 -> ~R 1 3 -> ~R 1 4 -> ~R 2 4
     -> p)
    -> p.
```

The proof of this is similar to the first half of the previous proof. We begin by explicitly giving R using `set` as before.

```
set R : set -> set -> prop := fun i j =>
     i = 0 /\ (j = 4 \/ j = 1)
  \/ i = 1 /\ (j = 0 \/ j = 2)
  \/ i = 2 /\ (j = 1 \/ j = 3)
  \/ i = 3 /\ (j = 2 \/ j = 4)
  \/ i = 4 /\ (j = 3 \/ j = 0).
```

We now use the `witness` tactic to reduce the subgoal from proving an existential goal to proving it for the given witness.

```
witness R.
```

We next let the proposition p be given and assume the property of p.

```
let p. assume Hp.
```

At this point we repeat the proofs of all the claims above before claim `L3`. We can use these claims along with the assumption `Hp` to finish the proof of this existential theorem.

We now turn to the proof of the main theorem again.

```
Theorem not_TwoRamseyProp_3_3_5 : ~TwoRamseyProp 3 3 5.
assume H1.
prove False.
```

This time instead of giving the relation R we use the existence theorem to have a variable R (of type $\iota \to \iota \to o$) satisfying sufficient properties.

```
apply exists_TwoRamseyProp_3_3_5.
let R. assume HR. apply HR.
assume HRsym HRI01 HRI12 HRI23 HRI34 HRI40 HNR02 HNR03 HNR13 HNR14 HNR24.
```

We then prove the claim L3 from above with mostly cosmetic changes and use the claim to complete the proof.

The resulting Proofgold document for this second version is roughly half the size of the first version and fits into a block.

4. A Formal Proof of $R(3,3) \leq 6$

We now describe a formal proof of $R(3,3) \leq 6$ in Megalodon. The two main previous proven results we will use deal with equipotence. First, if M and M' are equipotent, f is a bijection from V to V', and there is a clique $X \subseteq V$ of size M relative to a relation $\lambda xy.R\ (fx)\ (fy)$, then there is a clique $X \subseteq V'$ of size M' relative to R.

```
Axiom TwoRamseyProp_equip_lem : forall M M' V V',
  forall R':set -> set -> prop,
  forall f:set -> set,
    equip M M' ->
    bij V V' f ->
        (exists X c= V, equip M X /\ (forall x y :e X, x <> y -> R' (f x) (f y)))
     -> (exists X c= V', equip M' X /\ (forall x y :e X, x <> y -> R' x y)).
```

The second result we use states that 3 is equipotent to $\{u, v, w\}$ if $u \neq v$, $u \neq w$ and $v \neq w$. Here $\{u, v, w\}$ is notation for $\{u, v\} \cup \{w\}$ where the binary union operator, the unordered pair operator and the singleton operator satisfy the expected properties.

```
Axiom equip_3_I : forall u v w, u <> v -> u <> w -> v <> w -> equip 3 {u,v,w}.
```

We now prove three lemmas before proving the main result.

The first lemma simply allows us to infer we have a clique of size 3 if we have three distinct members of 6 satisfying the (symmetric) edge relation.

```
Theorem TwoRamseyProp_3_3_6_lem1 : forall R:set -> set -> prop,
    (forall x y, R x y -> R y x)
 -> forall u v w :e 6, u <> v -> u <> w -> v <> w -> R u v -> R u w -> R v w
 -> exists X c= 6, equip 3 X /\ (forall x y :e X, x <> y -> R x y).
```

We begin the proof in the usual way.

```
let R.
assume HR: forall x y, R x y -> R y x.
let u. assume Hu: u :e 6.
let v. assume Hv: v :e 6.
let w. assume Hw: w :e 6.
assume Huv Huw Hvw HRuv HRuw HRvw.
```

Our witness to the existence of a clique is $\{u, v, w\}$.

```
witness {u,v,w}.
```

Since the existential quantifier was restricted, we apply andI to split into two subgoals. The first subgoal requires proving $\{u, v, w\} \subseteq 6$. This can be easily accomplished using previous results about binary unions, unordered pairs and singletons. The second subgoal requires proving $\{u, v, w\}$ has size 3 and is a clique, so we apply andI again to split into two more subgoals. The fact that $\{u, v, w\}$ has size 3 follows from equip_3_I.

The fact that $\{u, v, w\}$ is a clique can be proven again using previous results about binary unions, unordered pairs and singletons and the assumptions of the lemma.

The second lemma states that the Ramsey Property holds if we assume there is an edge connecting 0 and 4 and an edge connecting 4 and 5.

```
Theorem TwoRamseyProp_3_3_6_lem2 : forall R:set -> set -> prop,
     R 0 4
  -> R 4 5
  -> (forall x y, R x y -> R y x)
  -> ((exists X c= 6, equip 3 X /\ (forall x y :e X, x <> y -> R x y))
   \/ (exists Y c= 6, equip 3 Y /\ (forall x y :e Y, x <> y -> ~R x y))).
let R.
assume H04: R 0 4.
assume H45: R 4 5.
assume HR: forall x y, R x y -> R y x.
```

We begin by proving the complement of R is also symmetric.

```
claim LRC: forall x y, ~R x y -> ~R y x.
{ let x y. assume H1 H2. apply H1. apply HR. exact H2. }
```

We next prove that if some $u \in 4$ is connected to both 4 and 5, then there is a clique.

```
claim L45: forall u :e 4, R u 4 -> R u 5
 -> exists X c= 6, equip 3 X /\ (forall x y :e X, x <> y -> R x y).
```

We can prove this using the first lemma once we prove $u \in 6$, $u \neq 4$ and $u \neq 5$, all of which follow from $u \in 4$ and previously proven results.

We next make a use of excluded middle, which has been previously proven.

```
Axiom xm : forall P:prop, P \/ ~P.
```

By applying `xm` with the proposition $R\ 0\ 5$ we can split into two subgoals. In the first we assume $R\ 0\ 5$ holds and so we have a clique using 0 and `L45`. In the second we assume $\neg R\ 0\ 5$.

```
apply xm (R 0 5).
- assume H05: R 0 5.
  apply orIL. exact L45 0 In_0_4 H04 H05.
- assume H05: ~R 0 5.
```

We now use excluded middle again with $R\ 1\ 4$.

Assume $R\ 1\ 4$ holds. Use excluded middle with $R\ 1\ 5$. If $R\ 1\ 5$ holds, we have a clique by `L45`. Assume $\neg R\ 1\ 5$. Use excluded middle with $R\ 0\ 1$. If $R\ 0\ 1$ holds, then 0, 1 and 4 give a clique (using the first lemma). Assume $\neg R\ 0\ 1$ holds. In this case 0, 1 and 5 form an anticlique. Note that an anticlique is simply a clique for $\lambda xy.\neg R\ x\ y$ so we can use the first lemma with $\lambda xy.\neg R\ x\ y$ and `LRC` to complete this subcase as well.

Assume $\neg R\ 1\ 4$. Use excluded middle with $R\ 2\ 4$.

Assume $R\ 2\ 4$ holds. Use excluded middle with $R\ 2\ 5$. If $R\ 2\ 5$, then we have a clique by `L45`. Assume $\neg R\ 2\ 5$. Use excluded middle with $R\ 0\ 2$. If $R\ 0\ 2$, then 0, 2 and 4 give a clique. If $\neg R\ 0\ 2$, then 0, 2 and 5 give an anticlique.

Assume $\neg R\ 2\ 4$. Use excluded middle with $R\ 3\ 4$. Assume $R\ 3\ 4$. Use excluded middle with $R\ 3\ 5$. If $R\ 3\ 5$, then we have a clique by `L45`. Assume $\neg R\ 3\ 5$. Use excluded middle with R 0 3. If R 0 3, then 0, 3 and 4 give a clique. If $\neg R$ 0 3, then 0, 3 and 5 give an anticlique.

Assume $\neg R$ 3 4. Use excluded middle on R 1 2. If $\neg R$ 1 2, then 1, 2 and 4 give an anticlique. Assume R 1 2. Use excluded middle on R 1 3. If $\neg R$ 1 3, then 1, 3 and 4 give an anticlique. Assume R 1 3. Use excluded middle on R 2 3. If R 2 3, then 1, 2 and 3 give a clique. If $\neg R$ 2 3, then 2, 3 and 4 give an anticlique. This exhausts all the possibilities and completes the proof of the second lemma.

(The proof of the second lemma could almost certainly be improved.)

We next turn to the third lemma. It is the same as the second lemma except it omits the assumption of an edge connecting 0 and 4.

```
Theorem TwoRamseyProp_3_3_6_lem3 : forall R:set -> set -> prop,
     R 4 5
  -> (forall x y, R x y -> R y x)
  -> ((exists X c= 6, equip 3 X /\ (forall x y :e X, x <> y -> R x y))
   \/ (exists Y c= 6, equip 3 Y /\ (forall x y :e Y, x <> y -> ~R x y))).
let R.
assume H45: R 4 5.
assume HR: forall x y, R x y -> R y x.
claim LRC: forall x y, ~R x y -> ~R y x.
{ let x y. assume H1 H2. apply H1. apply HR. exact H2. }
```

We use excluded middle on R 0 4. If R 0 4 holds, we can use the second lemma. This leaves only the case where $\neg R$ 0 4 holds.

```
apply xm (R 0 4).
- assume H04: R 0 4.
  exact TwoRamseyProp_3_3_6_lem2 R H04 H45 HR.
- assume H04: ~R 0 4.
```

Apply excluded middle with R 1 4. Assume R 1 4. We can prove this subcase using the second lemma and `TwoRamseyProp_equip_lem` by giving a bijection f from 6 to 6 (i.e., a permuation on 6) such that $f\ 0 = 1$, $f\ 1 = 0$ and $f\ x = x$ for other values of x. Such an f can be defined using a simple if-then-else. We omit the details here.

Assume $\neg R$ 1 4. We now follow the same procedure as above for 2. Apply excluded middle with R 2 4. If R 2 4, then we can define a permutation on 6 swapping 0 and 2 and use the second lemma and `TwoRamseyProp_equip_lem` to complete the subproof.

Assume $\neg R$ 2 4. Apply excluded middle with R 0 1, R 0 2 and R 1 2. If all hold, then 0, 1 and 2 give a clique. If $\neg R$ 1 2, then 1, 2 and 4 give an anticlique. If $\neg R$ 0 2, then 0, 2 and 4 give an anticlique. If $\neg R$ 0 1, then 0, 1 and 4 give an anticlique. This exhausts the possibilities and completes the proof of the third lemma.

We finally prove the main result.

```
Theorem TwoRamseyProp_3_3_6: TwoRamseyProp 3 3 6.
let R. assume HR.
set P : prop := (exists X c= 6, equip 3 X /\ (forall x y :e X, x <> y -> R x y))
             \/ (exists Y c= 6, equip 3 Y /\ (forall x y :e Y, x <> y -> ~R x y)).
```

We apply excluded middle on R 4 5 using the third lemma to finish the first case.

```
apply xm (R 4 5).
- assume H1: R 4 5.
  exact TwoRamseyProp_3_3_6_lem3 R H1 HR.
- assume H1: ~R 4 5.
```

In this second case we will let R' be the complement of the relation R. Note that R' 4 5 does hold and R' is symmetric.

```
  set R' : set -> set -> prop := fun x y => ~R x y.
  claim L1: R' 4 5.
  { exact H1. }
  claim L2: forall x y, R' x y -> R' y x.
  { let x y. assume H2 H3. apply H2. apply HR. exact H3. }
```

Applying the third lemma with R' we obtain either a clique or anticlique with respect to R'. The existence of a clique for R' is precisely the same (up to conversion) as the existence of an anticlique for R, so the first case can be completed easily. In the second case we will prove the complement of R' is again R and use this to prove that the existence of an anticlique for R yields a clique for R.

```
  apply TwoRamseyProp_3_3_6_lem3 R' L1 L2.
  + apply orIR.
  + claim L3: (fun x y:set => ~R' x y) = R.
```

To prove these two relations are equal we use functional extensionality (`func_ext`). Functional extensionality is a special proof term construct that expects two types σ and τ and returns a proof of

$$\forall fg:\sigma\to\tau.(\forall x:\sigma.fx=gx)\to f=g.$$

```
    { apply func_ext set (set -> prop).
```

We now need to prove $\forall x.(\lambda y.\neg R'\ x\ y) = (R\ x)$.

```
      let x.
      prove (fun y:set => ~R' x y) = (R x).
```

We could apply functional extensionality again followed by propositional extensionality, but for type $\iota\to o$ we have a previously proven result allowing us to prove two unary predicates are equal if they are included in each other. In this case the two subgoals are easy to complete (using a double negation property called `dneg` here)

```
      apply pred_ext_2.
      - let y. prove ~ ~ R x y -> R x y. apply dneg.
      - let y. assume H2 H3. exact H3 H2.
    }
```

We now use the `prove` tactic to carefully modify the goal so that we can rewrite with `L3`. Recall that P is the local name for our main conclusion.

```
    prove (exists X c= 6, equip 3 X
                      /\ (forall x y :e X, x <> y -> ~R' x y))
      -> P.
    prove (exists X c= 6, equip 3 X
              /\ (forall x y :e X, x <> y -> (fun x y:set => ~R' x y) x y))
```

```
    -> P.
```

After rewriting the `L3` the goal can be completed by `orIL` since the antecedent of the implication now simply states there exists a clique for R of size 3.

```
rewrite L3.
apply orIL.
```

This completes the proof of the main result.

5. Conclusion

The proofs described above demonstrate that it is feasible, though tedious, to prove propositions of the form $R(m,n) > K$ and $R(m,n) \leq K$. The next simplest such propositions to prove would be, e.g., $R(3,4) > 8$ (by giving an appropriate relation on 8), $R(4,4) > 17$ (by giving an appropriate relation on 17), $R(3,4) \leq 9$ and $R(4,4) \leq 18$. The somewhat brute force approach described above would likely need to be refined before being applied with larger numbers (even as "large" as 4). To incentivize creating such refinements Proofgold bounties can be placed on these relatively small Ramsey problems. In addition, since Ramsey numbers are only known for very few values of m and n, it is easy to create propositions that are open problems in combinatorics. Proofgold bounties could be placed on these as well, if only to encourage curious people to look more closely at the Megalodon system and Proofgold network, as it seems unlikely an open problem would be solved via a formal proof. Tables 1, 2, 3, 4, 5, 6 and 7 list several conjectures distributed with Megalodon 1.3 and the corresponding Proofgold address where a bounty could be placed and later claimed by resolving the conjecture with an appropriate proof.

References

[1] Brown, C.E., Pąk, K.: A tale of two set theories. In: Kaliszyk, C., Brady, E.C., Kohlhase, A., Coen, C.S. (eds.) Intelligent Computer Mathematics - 12th International Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11617, pp. 44–60. Springer (2019)

[2] Church, A.: A formulation of the simple theory of types. The Journal of Symbolic Logic **5**, 56–68 (1940)

[3] The Coq development team: The Coq proof assistant reference manual. LogiCal Project (2020), http://coq.inria.fr, version 8.12

[4] Prawitz, D.: Natural deduction: a proof-theoretical study. Dover (2006)

[5] Russell, B.: The Principles of Mathematics. Cambridge University Press (1903)

Proposition	Megalodon Name	Proofgold Address
$R(3,4) > 8$	`not_TwoRamseyProp_3_4_8`	`TMZTRwksZ429x7jwcuaATRxKhUvGEdp5Nyj`
$R(3,5) > 13$	`not_TwoRamseyProp_3_5_13`	`TMcGK8KWhfaCkgZUqXFaa4wSTBr6oTf89PQ`
$R(3,6) > 17$	`not_TwoRamseyProp_3_6_17`	`TMMeqRAd9xzG4jtxa9EUaL64uqRevdds6jW`
$R(3,7) > 22$	`not_TwoRamseyProp_3_7_22`	`TMJKbZjgHwR1R1hCak4EgwPjmA8HNR1AzVe`
$R(3,8) > 27$	`not_TwoRamseyProp_3_8_27`	`TMXv8MmejVaeybNBgqmLP4so3SqWZWxqJE7`
$R(3,9) > 35$	`not_TwoRamseyProp_3_9_35`	`TMdKrj9bPB76LVKYCPjJUYHDSFYdfmXkww3`
$R(3,10) > 39$	`not_TwoRamseyProp_3_10_39`	`TMKUcnY4Z28obHfQWVZn56bFru9zRQdBzzA`
$R(4,4) > 17$	`not_TwoRamseyProp_4_4_17`	`TMFYWunQezDrkhpMUee7saZdayoEVM4SsbP`
$R(4,5) > 24$	`not_TwoRamseyProp_4_5_24`	`TMXaEY6oQ7QdVZuqvs3nB3u1a3PaeREqW4b`
$R(4,6) > 35$	`not_TwoRamseyProp_4_6_35`	`TMTP6yNEuLJgxabKFMpXXG2jXMmGBhXo7LG`
$R(4,7) > 48$	`not_TwoRamseyProp_4_7_48`	`TMWNCQzzBFKBFRj2Vg4QhjCUg9Pno3QpkbU`
$R(4,8) > 58$	`not_TwoRamseyProp_4_8_58`	`TMEoc7MD7M2kuGeqGA7VTD67gWsQvS7QyjS`
$R(4,9) > 72$	`not_TwoRamseyProp_4_9_72`	`TMVDStJK2rAYDiNJCHMNaCGuAU5KY6GpwG1`
$R(5,5) > 42$	`not_TwoRamseyProp_5_5_42`	`TMbu5cNHpJ8J5t435KssbZiaKf7Av2ASGBH`
$R(5,6) > 57$	`not_TwoRamseyProp_5_6_57`	`TMGVzjSdaN6obxzPsnYbvq9nQJ5sH3qYUGL`
$R(5,7) > 79$	`not_TwoRamseyProp_5_7_79`	`TMW6xdLtbxVSHVaUL6epngvPdpDXyzQoeHp`
$R(5,8) > 100$	`not_TwoRamseyProp_5_8_100`	`TMPs8cxp8Zg5ugHirdaPo8qrEuBCkcZ8tcf`
$R(6,6) > 101$	`not_TwoRamseyProp_6_6_101`	`TMLsyJxzLDBGJt4RLuXffhVfBnk72RE9DL4`
$R(6,7) > 114$	`not_TwoRamseyProp_6_7_114`	`TMcWxeWu1w1WJJfzUim5sTbF4UaXzfcJuDa`

TABLE 1. Best Known Lower Bounds for Small Ramsey Numbers

Proposition	Megalodon Name	Proofgold Address
$R(3,5) > 2^3$	not_TwoRamseyProp_3_5_Power_3	TMVHzDbwLEUq7ZdF7SwdxojLSSZrETq2ty7
$R(3,6) > 2^3$	not_TwoRamseyProp_3_6_Power_3	TMU5S3VcCxhSGM557BB2pMPSuPt47UcKSax
$R(3,6) > 2^4$	not_TwoRamseyProp_3_6_Power_4	TMLyvZypyq14D3Q2vyHSse88YzHewwRcCDB
$R(3,7) > 2^3$	not_TwoRamseyProp_3_7_Power_3	TMYhjqt4H2QCQA9Zta5ft1wkbEUoFrtyrHr
$R(3,7) > 2^4$	not_TwoRamseyProp_3_7_Power_4	TMRgsDDE1r3HdbuN5FnQLLbtL4VxT2U4oxm
$R(3,8) > 2^3$	not_TwoRamseyProp_3_8_Power_3	TMdrnqT6ipbhyhibu4PcP9mdhihVk6gdmpW
$R(3,8) > 2^4$	not_TwoRamseyProp_3_8_Power_4	TMEzzwYRHW3oFCPn7Htm5N2aQV4NeasT58S
$R(3,9) > 2^3$	not_TwoRamseyProp_3_9_Power_3	TMH8CuMPvv8wWJ9ctLwnaHum2JwFwWZzhf1
$R(3,9) > 2^4$	not_TwoRamseyProp_3_9_Power_4	TMUYsSwSxE7bzhNET5QLRoGiQNtgyLc2q5s
$R(3,9) > 2^5$	not_TwoRamseyProp_3_9_Power_5	TMPP44pD9m61Ua5QkmuR3pGQpGyy1uVuguG
$R(3,10) > 2^3$	not_TwoRamseyProp_3_10_Power_3	TMY6yrGra8BMSi2im1iBh6Y5hhYyP9C2Mhc
$R(3,10) > 2^4$	not_TwoRamseyProp_3_10_Power_4	TMFWhbv4YPX463UqrwxCVfPnPMgkcn1EwGf
$R(3,10) > 2^5$	not_TwoRamseyProp_3_10_Power_5	TMVPg3DwriFjvu4n82kK8Ka8fkKqzddN4vH
$R(4,4) > 2^3$	not_TwoRamseyProp_4_4_Power_3	TMb2Sy7PW4A7JxWuNHKhF1RyJi1ALA12yEn
$R(4,4) > 2^4$	not_TwoRamseyProp_4_4_Power_4	TMV3LTHyCBnhXvur7GPWJBv9EbT5YUC4S8K
$R(4,5) > 2^3$	not_TwoRamseyProp_4_5_Power_3	TMSiba4Q39wbstpCeDBgTvu9cgj6j7foegZ
$R(4,5) > 2^4$	not_TwoRamseyProp_4_5_Power_4	TMK6infbDwDHS4b8r9xmZohrD5GBwPDg9Y6
$R(4,6) > 2^3$	not_TwoRamseyProp_4_6_Power_3	TMdAvV5dAUEbvvnvouGFbsYA3hNp6AZi2gh
$R(4,6) > 2^4$	not_TwoRamseyProp_4_6_Power_4	TMGspMoqWL7PcaQVkGiQko7y57XzeaECkiZ
$R(4,6) > 2^5$	not_TwoRamseyProp_4_6_Power_5	TMNEbNDi1x9cRwNgQ365q4mQrTA21mcZzpE
$R(4,7) > 2^3$	not_TwoRamseyProp_4_7_Power_3	TMbxdFJCUamZbt2MySrR82XMAYXTTixrrL4
$R(4,7) > 2^4$	not_TwoRamseyProp_4_7_Power_4	TMReAUbFDvWy1AjAZs14R81dre36VoPsfS5
$R(4,7) > 2^5$	not_TwoRamseyProp_4_7_Power_5	TMKi1D84D9M7yXLUMiez2HapYajJVBTbpiu
$R(4,8) > 2^3$	not_TwoRamseyProp_4_8_Power_3	TMMqajLmzgVcjmZyYogunHvh5hWoxHNDAq4
$R(4,8) > 2^4$	not_TwoRamseyProp_4_8_Power_4	TMQtVpgWyNwmDEV4a1phzjan4LZCrtCQ8AV
$R(4,8) > 2^5$	not_TwoRamseyProp_4_8_Power_5	TMGQTLwKsfK567DHtN8vAVcv2UTVvaSG6AA
$R(4,9) > 2^3$	not_TwoRamseyProp_4_9_Power_3	TMHUSbZSpkav3xHNGZ7eqfLj6JfLMimtjNs
$R(4,9) > 2^4$	not_TwoRamseyProp_4_9_Power_4	TMahh56eTuXexupey9e4a2qs5ebXZ1CqUEz
$R(4,9) > 2^5$	not_TwoRamseyProp_4_9_Power_5	TMFCRqFBwt3J6dxcFHn5EgswrUMNWr9nBFZ
$R(4,9) > 2^6$	not_TwoRamseyProp_4_9_Power_6	TMQGT6GGrGNc3fnpNVVyzYp1UfDQkmUx5hh

TABLE 2. Weak Lower Bounds for Small Ramsey Numbers (1)

Proposition	Megalodon Name	Proofgold Address
$R(5,5) > 2^3$	not_TwoRamseyProp_5_5_Power_3	TMK7GHozXXyUDvZ7sWzWvZ2D2rA4QwFie5p
$R(5,5) > 2^4$	not_TwoRamseyProp_5_5_Power_4	TMFG8Vmyysj75BgACYiq1K3RPqwRAgxQupZ
$R(5,5) > 2^5$	not_TwoRamseyProp_5_5_Power_5	TMGmDYbRXrpoiZjRi312cY1fvzjX3vVcMbz
$R(5,6) > 2^3$	not_TwoRamseyProp_5_6_Power_3	TMcxptoPKiJ8Q67vE6chP4KauMz56jXhuHS
$R(5,6) > 2^4$	not_TwoRamseyProp_5_6_Power_4	TMFbYaKNtu4zJfq46WL5ucePjpjFmuGjeLQ
$R(5,6) > 2^5$	not_TwoRamseyProp_5_6_Power_5	TMc5MAkbjb79K9dcjGUEYC76314eQZkiw1c
$R(5,7) > 2^3$	not_TwoRamseyProp_5_7_Power_3	TMJ6qQh3fADYVrttCTFyjugyUroHquv9VWk
$R(5,7) > 2^4$	not_TwoRamseyProp_5_7_Power_4	TMHp76CfXyf9JAxPqM3r8GumwMrRLHTjim5
$R(5,7) > 2^5$	not_TwoRamseyProp_5_7_Power_5	TMRXV6eS9sr6MsLUcUz8gb8uXmdYMxjNmFJ
$R(5,7) > 2^6$	not_TwoRamseyProp_5_7_Power_6	TMNXFB9HwrMCgkiMLxVBRW5nkQVbRbLU9YD
$R(5,8) > 2^3$	not_TwoRamseyProp_5_8_Power_3	TMV4oLWxxL9gduEPEmbPFnaF48mrcfXvy4Z
$R(5,8) > 2^4$	not_TwoRamseyProp_5_8_Power_4	TMccab19u1j95mqYTexxZsq7E8qdHsBL31Q
$R(5,8) > 2^5$	not_TwoRamseyProp_5_8_Power_5	TMPaUGYW3WyPmb41n83C3Z1tqsB6wYa5DQH
$R(5,8) > 2^6$	not_TwoRamseyProp_5_8_Power_6	TMYZ6xiWaWa4LUtduMqzRe1BndeKk1nnKcV
$R(6,6) > 2^3$	not_TwoRamseyProp_6_6_Power_3	TMWJvuKAYvjKtFnHjHUoDnoTzE8kibgeGit
$R(6,6) > 2^4$	not_TwoRamseyProp_6_6_Power_4	TMYNoFmD6Rb2sEwpA7PgB7cRraVw1JDdGHZ
$R(6,6) > 2^5$	not_TwoRamseyProp_6_6_Power_5	TMTYNCqRgcySPyfZGBzZgcqScw5RS6X2hhC
$R(6,6) > 2^6$	not_TwoRamseyProp_6_6_Power_6	TMTxgdseSpFSGHDNR7RSMbhK5Wk3jSstF5Q
$R(6,7) > 2^3$	not_TwoRamseyProp_6_7_Power_3	TMPTEnXyPsjhnTtGm6Lrnj4MzaEboGdfVfR
$R(6,7) > 2^4$	not_TwoRamseyProp_6_7_Power_4	TMaZ4F5ahqctbqyMkTp4YZkmtTKg1JZBdEX
$R(6,7) > 2^5$	not_TwoRamseyProp_6_7_Power_5	TMVnsD3eB7hnvajnroG6dpv4ve85pkroCCz
$R(6,7) > 2^6$	not_TwoRamseyProp_6_7_Power_6	TMExwwLP8cRBB4esj7b44gEfGL8BFAi8H6F

Table 3. Weak Lower Bounds for Small Ramsey Numbers (2)

Proposition	Megalodon Name	Proofgold Address
$R(3,4) \leq 9$	`TwoRamseyProp_3_4_9`	`TMNqnATDwgaDsTFX1M9RmSouPYDn9u67dAZ`
$R(3,5) \leq 14$	`TwoRamseyProp_3_5_14`	`TMbShW7mSiehCqEg6gfAGZXGbBLt5TA2atT`
$R(3,6) \leq 18$	`TwoRamseyProp_3_6_18`	`TMbJ1MogStdKCGN3J3j1hThprkcWjA8ggEB`
$R(3,7) \leq 23$	`TwoRamseyProp_3_7_23`	`TMGKH8FtpMKQL5nSjsU22A45gjtZpd5Yarm`
$R(3,8) \leq 28$	`TwoRamseyProp_3_8_28`	`TMJeasRF8U3ej45EdWxUA85TXjati3kbPyS`
$R(3,9) \leq 36$	`TwoRamseyProp_3_9_36`	`TMavx891tAmc71YDKiSXQBD2Rm932Qp23tk`
$R(3,10) \leq 42$	`TwoRamseyProp_3_10_42`	`TMLgkDpY2A4KZzDRtTk8w1SzySmFN7ygd1R`
$R(4,4) \leq 18$	`TwoRamseyProp_4_4_18`	`TMbeXnozpPPnYPGE5hdooJdVvco3rwg45tS`
$R(4,5) \leq 25$	`TwoRamseyProp_4_5_25`	`TMaGCJ8SmUjESVqqhCFM894vFTpNfi7Zcbo`
$R(4,6) \leq 41$	`TwoRamseyProp_4_6_41`	`TMGyu8j3dnjTxE99spsMGXnkEp8QU5d7ePK`
$R(4,7) \leq 61$	`TwoRamseyProp_4_7_61`	`TMXtyXPKJFL6wYpipDCsjK7fJScGSuFPN1S`
$R(4,8) \leq 84$	`TwoRamseyProp_4_8_84`	`TML8t7q8Wz5HP9YGrvJcp2vGuNDwrRGNP57`
$R(4,9) \leq 115$	`TwoRamseyProp_4_9_115`	`TMRc1h6FhfYCoCofHKNvxpR82dzXF2ndG1R`
$R(5,5) \leq 48$	`TwoRamseyProp_5_5_48`	`TMctie53cBVHdM4Dvc2CpJMCm7QGMPDGRQm`
$R(5,6) \leq 87$	`TwoRamseyProp_5_6_87`	`TMcD2ZaGBnE6sxdoAA2GFAG6mQsKZPZHvdq`
$R(5,7) \leq 143$	`TwoRamseyProp_5_7_143`	`TMVSgM5dy7dbA4eVmKwUK4yUt8vdufNQapi`
$R(5,8) \leq 216$	`TwoRamseyProp_5_8_216`	`TMMm14Wr2mksV4NLhHHbALNnE9PqjBKWHKT`
$R(6,6) \leq 165$	`TwoRamseyProp_6_6_165`	`TMFcryYsFNf2kRz2njXmBfrmXoxJ9Dp7FkP`
$R(6,7) \leq 298$	`TwoRamseyProp_6_7_298`	`TMTj9pJWZUioWqtxHD9EmyauhheXzt3yYRj`

TABLE 4. Best Known Upper Bounds for Small Ramsey Numbers

Proposition	Megalodon Name	Proofgold Address
$R(3,4) \leq 2^4$	TwoRamseyProp_3_4_Power_4	TMa4Lc4AyMhSWuiBQmLra5Cex3wBXf15o16
$R(3,4) \leq 2^5$	TwoRamseyProp_3_4_Power_5	TMQzy35yBcche2UXwdvoZVnt3dyp4LnFDNx
$R(3,4) \leq 2^6$	TwoRamseyProp_3_4_Power_6	TMchT1fHoSCWkPkaNu2nSYwcXiQaD73EHTD
$R(3,4) \leq 2^7$	TwoRamseyProp_3_4_Power_7	TMNHo4B3BUr9TbU35fH9h7nCyQRMLfjAGhA
$R(3,4) \leq 2^8$	TwoRamseyProp_3_4_Power_8	TMW38xrr6tkiKg7v3rwvzcQ3h7M1j5bb3Po
$R(3,5) \leq 2^4$	TwoRamseyProp_3_5_Power_4	TMGMoKMeC2QJu2UhVYot5qJzaXu3dYkTHUb
$R(3,5) \leq 2^5$	TwoRamseyProp_3_5_Power_5	TMG3jmdBxNxSKQWdVCkb63peTMJTqK4n1EP
$R(3,5) \leq 2^6$	TwoRamseyProp_3_5_Power_6	TMGzjbC1rwhqhaSAB4bJquzeoVAfqjbXQ45
$R(3,5) \leq 2^7$	TwoRamseyProp_3_5_Power_7	TMMdpPGZzqSgcaWdRPhoxs8FZagr1Ku3sh8
$R(3,5) \leq 2^8$	TwoRamseyProp_3_5_Power_8	TMTsPkynff2LXJmYGRT2UET2mF6RGJNXZVf
$R(3,6) \leq 2^5$	TwoRamseyProp_3_6_Power_5	TMVhEw2QbtGaDDY51t2cwuHDMUzaETWib2D
$R(3,6) \leq 2^6$	TwoRamseyProp_3_6_Power_6	TMP3JV9tNhcdfcVDZj734eiYJhHwDfegyBB
$R(3,6) \leq 2^7$	TwoRamseyProp_3_6_Power_7	TMPDXGn6qXa3HLYqqLBszZJeUxFMihnMw4B
$R(3,6) \leq 2^8$	TwoRamseyProp_3_6_Power_8	TMSMammDfv4rySY6UrYupgxMbaKDFRTHpRs
$R(3,7) \leq 2^5$	TwoRamseyProp_3_7_Power_5	TMb4eT18NCpi4ejC4ECKxNQaagNAEGTj4LA
$R(3,7) \leq 2^6$	TwoRamseyProp_3_7_Power_6	TMUmwLeejdDfRmrpuF1TaHgjbWn93x6nNLU
$R(3,7) \leq 2^7$	TwoRamseyProp_3_7_Power_7	TMPUFYxWzFBgwayouu4SUU8EzvgByyNgcRD
$R(3,7) \leq 2^8$	TwoRamseyProp_3_7_Power_8	TMHQTw2ojQ2k5rHnX37gyhp5MT9LPqwD5Ri
$R(3,8) \leq 2^5$	TwoRamseyProp_3_8_Power_5	TMXaAN4JK3kL6dtksJvq6QFMvxFqNdDnm74
$R(3,8) \leq 2^6$	TwoRamseyProp_3_8_Power_6	TMS7xacvnKSxxiXgHWv6QKF29XtvuLERhYD
$R(3,8) \leq 2^7$	TwoRamseyProp_3_8_Power_7	TMWq6xRb5gdxucG1V74VfEoc5e8YiEdpg3P
$R(3,8) \leq 2^8$	TwoRamseyProp_3_8_Power_8	TMG2j6KQNEncd1vzvTgFjYmAtyCVQjohd29
$R(3,9) \leq 2^6$	TwoRamseyProp_3_9_Power_6	TMRpVz5Q3k2b8S6a2jGKwBWyUNFh8KbNoNo
$R(3,9) \leq 2^7$	TwoRamseyProp_3_9_Power_7	TMM77b3mnpsLDSBDVUzxoB9SFRd1mopPYzh
$R(3,9) \leq 2^8$	TwoRamseyProp_3_9_Power_8	TMcBQxbdGYVjjofGuDJiHwa928JUDUhzRF2
$R(3,10) \leq 2^6$	TwoRamseyProp_3_10_Power_6	TMHXjGJcjf1kvrEFrD1XP3Jds6hXJyvEFQD
$R(3,10) \leq 2^7$	TwoRamseyProp_3_10_Power_7	TMNxMK9VqypkQxHtcMqNFv1TdqTsc1SmxJj
$R(3,10) \leq 2^8$	TwoRamseyProp_3_10_Power_8	TMP3ZwqJREJLdQZ2JJjCfzTu2gmLShT4zSu

TABLE 5. Weak Upper Bounds for Small Ramsey Numbers (1)

Proposition	Megalodon Name	Proofgold Address
$R(4,4) \leq 2^5$	TwoRamseyProp_4_4_Power_5	TMM1zbu3WmwYVsxxmcESD8kFqbfFQwTcR4f
$R(4,4) \leq 2^6$	TwoRamseyProp_4_4_Power_6	TMJv5tmSnJSKJgVYvRKQVVMBttzVWdfnLJf
$R(4,4) \leq 2^7$	TwoRamseyProp_4_4_Power_7	TMcpdNTccL8xFQ6JC8GfX7sHkqzpvE9KXaR
$R(4,4) \leq 2^8$	TwoRamseyProp_4_4_Power_8	TMYh85Mw7Hngazh6x1sa1XBJgDz7EvtDSmP
$R(4,5) \leq 2^5$	TwoRamseyProp_4_5_Power_5	TMV3YoyvmjyjxRvvRFvzzy1SBNJiVLAPcj7
$R(4,5) \leq 2^6$	TwoRamseyProp_4_5_Power_6	TMNtFFfd9MiGZHZJfJRTckZNh38Pw2NJkh4
$R(4,5) \leq 2^7$	TwoRamseyProp_4_5_Power_7	TMT72t5wVipcDyKEUp2NxiqgKBBeFAzCwTA
$R(4,5) \leq 2^8$	TwoRamseyProp_4_5_Power_8	TMHfWMTzVfcKRiJmWVLEZgC8gKLUfsZsF6J
$R(4,6) \leq 2^6$	TwoRamseyProp_4_6_Power_6	TMd3rAdqu6qzfWvG6z8fctmdsnvMEex6ykJ
$R(4,6) \leq 2^7$	TwoRamseyProp_4_6_Power_7	TMaxYGLoNcZ9npUhCq7eKJZwDrsBerDyapG
$R(4,6) \leq 2^8$	TwoRamseyProp_4_6_Power_8	TMbtUxRqtMgd4aNMTSCvkjGoUv8hpYZp8Nk
$R(4,7) \leq 2^6$	TwoRamseyProp_4_7_Power_6	TMUTpcc8in2onNu2AC3fbS6Q9SegWNaMCmE
$R(4,7) \leq 2^7$	TwoRamseyProp_4_7_Power_7	TMF6AWwpeiybjsgiLTDQnEnRddXAVqdwusM
$R(4,7) \leq 2^8$	TwoRamseyProp_4_7_Power_8	TMSLHY3cxsrK4S1KyUWnWZkGMrWWNc1uQST
$R(4,8) \leq 2^7$	TwoRamseyProp_4_8_Power_7	TMNx667poeYtsT7KTZTGnyLTq5KdQm1nJxa
$R(4,8) \leq 2^8$	TwoRamseyProp_4_8_Power_8	TMSkRiSt5qZMy1KuD4arx7FcdmvDDNJib5f
$R(4,9) \leq 2^7$	TwoRamseyProp_4_9_Power_7	TMYjJ5cgHonhxfDky3NTprMxo36acV14s8x
$R(4,9) \leq 2^8$	TwoRamseyProp_4_9_Power_8	TMKYb87veLxzurwXFUttEMEyFTGChwJXW6Z
$R(5,5) \leq 2^6$	TwoRamseyProp_5_5_Power_6	TMFmfYexZpQQTd5iYiCqFrsY32PBd2EqPzm
$R(5,5) \leq 2^7$	TwoRamseyProp_5_5_Power_7	TMJPAjAu1aKMLh3RpYFM65GzSwohzEYgaA9
$R(5,5) \leq 2^8$	TwoRamseyProp_5_5_Power_8	TMSwfavHeMj9kJLBS2Sc79vbDjTwV2rRpb2
$R(5,6) \leq 2^7$	TwoRamseyProp_5_6_Power_7	TMdu7Jgg2EakX7Nvz69Z1gfVbW2afcXvDVA
$R(5,6) \leq 2^8$	TwoRamseyProp_5_6_Power_8	TMZDhzNsA5pbx1rQ29J1jQEUqKozxpVpnPS
$R(5,7) \leq 2^8$	TwoRamseyProp_5_7_Power_8	TMQjipjMrCSNCqNGoiGJVr3qghrzBSnicBx
$R(5,8) \leq 2^8$	TwoRamseyProp_5_8_Power_8	TMQjEqYWm45Jzke34ekXu4NhZB6ejU4NJKb
$R(6,6) \leq 2^8$	TwoRamseyProp_6_6_Power_8	TMH3bevkcqvh8mvvjxFmnKn1ZN54nPPmNma

TABLE 6. Weak Upper Bounds for Small Ramsey Numbers (2)

Proposition	Megalodon Name	Proofgold Address
$R(3,10) \leq 40$	TwoRamseyProp_3_10_40	TMdvEn8UmrSjZRZwuJBKRu3CDNJVsVpbEZJ
$R(3,10) \leq 41$	TwoRamseyProp_3_10_41	TMFULvhHjCsGFYPpsvnET6iD7oP1RL5JiHW
$R(4,6) \leq 36$	TwoRamseyProp_4_6_36	TMUnmw49fNPNqFFquQctTBy16LwoDRrxTg7
$R(4,6) \leq 40$	TwoRamseyProp_4_6_40	TMN4q5W37NkZMoR6vedSpCh9okznLfUoQbH
$R(4,7) \leq 49$	TwoRamseyProp_4_7_49	TMXZCujSRwE3ypPsErAyadfWgkYTrvzwgKb
$R(4,7) \leq 60$	TwoRamseyProp_4_7_60	TMa9UyXyWCAfApzbofsANmb2xABVQdP4og7
$R(4,8) \leq 59$	TwoRamseyProp_4_8_59	TMdvK7rphvbxbWK3nPKGsaqxfChhJg49aCK
$R(4,8) \leq 83$	TwoRamseyProp_4_8_83	TMVJbMfaXk3HFrKnfAMfXYBJzo57MiwdVe6
$R(4,9) \leq 73$	TwoRamseyProp_4_9_73	TMVRXEJ5cjvMtDZmVkM6JU5ZrWgQHMk4dVn
$R(4,9) \leq 114$	TwoRamseyProp_4_9_114	TMKENiMj8e2EAfjfbMajJn6DVuR7nGTomoB
$R(5,5) \leq 43$	TwoRamseyProp_5_5_43	TMNUq5RbcM6eMfgZZQwN7xZNjZQaoV9H2Vo
$R(5,5) \leq 47$	TwoRamseyProp_5_5_47	TMHQLAu8LNq833psgUYemBdTPiRj2wTeAyb
$R(5,6) \leq 58$	TwoRamseyProp_5_6_58	TMPGnKYdBa3khY844daC3MQL8bRM2LmZ8yv
$R(5,6) \leq 86$	TwoRamseyProp_5_6_86	TMJ7MPMgsZpZQkTaVS352SSVg1mcEGMKtkx
$R(5,7) \leq 80$	TwoRamseyProp_5_7_80	TMbpMY5y2tVTnzo6TLtjiRUusv3eKTSXsKn
$R(5,7) \leq 142$	TwoRamseyProp_5_7_142	TMa7KQh2wjJGHQ2Kyf5VPBowLgAaKWQ1LDa
$R(5,8) \leq 101$	TwoRamseyProp_5_8_101	TMXm5VieR3P3dBUvtSEpesaMqWpwDN1M3Va
$R(5,8) \leq 215$	TwoRamseyProp_5_8_215	TMLZCSMmwK2QPMcAiYXtjNvHzpWoeRrAtn2
$R(6,6) \leq 102$	TwoRamseyProp_6_6_102	TMVADVhyFCzdS62thUm1sCyWzWwqbrZgrjq
$R(6,6) \leq 164$	TwoRamseyProp_6_6_164	TMW9bczVUMdHGUsDKNseVb6Z4KJxYd5HK9R
$R(6,7) \leq 115$	TwoRamseyProp_6_7_115	TMWy6aLQZ8awR6aqVNTTgj3ZLwovY2a39s2
$R(6,7) \leq 297$	TwoRamseyProp_6_7_297	TMZXeYbd2fNjbxnQqC2Fxqgp3AK6JjC9dHT

TABLE 7. Open Problems for Small Ramsey Numbers