Ramsey's Theorem

Tom Ridge

December 12, 2009

Abstract

The infinite form of Ramsey's Theorem is proved following Boolos and Jeffrey, Chapter 26.

Contents

1 Infinite Sets and Related Concepts

theory *Infinite-Set*
imports *Main*
begin

1.1 Infinite Sets

Some elementary facts about infinite sets, mostly by Stefan Merz. Beware! Because "infinite" merely abbreviates a negation, these lemmas may not work well with *blast*.

abbreviation
  infinite :: *'a set* ⇒ *bool* **where**
  infinite S == ¬ *finite S*

Infinite sets are non-empty, and if we remove some elements from an infinite set, the result is still infinite.

lemma *infinite-imp-nonempty*: *infinite* S ==> $S \neq \{\}$
⟨*proof*⟩

lemma *infinite-remove*:
infinite $S \Longrightarrow$ *infinite* $(S - \{a\})$
⟨*proof*⟩

lemma *Diff-infinite-finite*:
assumes T: *finite* T **and** S: *infinite* S
shows *infinite* $(S - T)$
⟨*proof*⟩

lemma *Un-infinite*: *infinite* $S \Longrightarrow$ *infinite* $(S \cup T)$
⟨*proof*⟩

lemma *infinite-super*:
assumes T: $S \subseteq T$ **and** S: *infinite* S
shows *infinite* T
⟨*proof*⟩

As a concrete example, we prove that the set of natural numbers is infinite.

lemma *finite-nat-bounded*:
assumes S: *finite* (S::*nat set*)
shows $\exists\, k.\ S \subseteq \{..<k\}$ (**is** $\exists\, k.$ *?bounded* S k)
⟨*proof*⟩

lemma *finite-nat-iff-bounded*:
finite (S::*nat set*) = $(\exists\, k.\ S \subseteq \{..<k\})$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *finite-nat-iff-bounded-le*:
finite (S::*nat set*) = $(\exists\, k.\ S \subseteq \{..k\})$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *infinite-nat-iff-unbounded*:
infinite (S::*nat set*) = $(\forall\, m.\ \exists\, n.\ m<n \wedge n \in S)$
(**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *infinite-nat-iff-unbounded-le*:
infinite (S::*nat set*) = $(\forall\, m.\ \exists\, n.\ m \leq n \wedge n \in S)$
(**is** *?lhs* = *?rhs*)
⟨*proof*⟩

For a set of natural numbers to be infinite, it is enough to know that for any number larger than some k, there is some larger number that is an element of the set.

lemma *unbounded-k-infinite*:
assumes *k*: $\forall m.\ k<m \longrightarrow (\exists n.\ m<n \wedge n\in S)$
shows *infinite* $(S::nat\ set)$
⟨*proof*⟩

lemma *nat-infinite* [*simp*]: *infinite* (*UNIV* :: *nat set*)
⟨*proof*⟩

lemma *nat-not-finite* [*elim*]: *finite* (*UNIV*::*nat set*) $\Longrightarrow R$
⟨*proof*⟩

Every infinite set contains a countable subset. More precisely we show that a set S is infinite if and only if there exists an injective function from the naturals into S.

lemma *range-inj-infinite*:
inj $(f::nat \Rightarrow 'a) \Longrightarrow$ *infinite* (*range* f)
⟨*proof*⟩

lemma *int-infinite* [*simp*]:
shows *infinite* (*UNIV*::*int set*)
⟨*proof*⟩

The "only if" direction is harder because it requires the construction of a sequence of pairwise different elements of an infinite set S. The idea is to construct a sequence of non-empty and infinite subsets of S obtained by successively removing elements of S.

lemma *linorder-injI*:
assumes *hyp*: !!$x\ y.\ x < (y::'a::linorder)$ ==> $f\,x \neq f\,y$
shows *inj f*
⟨*proof*⟩

lemma *infinite-countable-subset*:
assumes *inf*: *infinite* $(S::'a\ set)$
shows $\exists f.$ *inj* $(f::nat \Rightarrow 'a) \wedge$ *range* $f \subseteq S$
⟨*proof*⟩

lemma *infinite-iff-countable-subset*:
infinite $S = (\exists f.$ *inj* $(f::nat \Rightarrow 'a) \wedge$ *range* $f \subseteq S)$
⟨*proof*⟩

For any function with infinite domain and finite range there is some element that is the image of infinitely many domain elements. In particular, any infinite sequence of elements from a finite set contains some element that occurs infinitely often.

lemma *inf-img-fin-dom*:
assumes *img*: *finite* (f‘A) **and** *dom*: *infinite* A
shows $\exists y \in f$‘A. *infinite* (f −‘ $\{y\}$)
⟨*proof*⟩

lemma *inf-img-fin-domE*:
assumes *finite* (f‘A) **and** *infinite* A
obtains y **where** $y \in f$‘A **and** *infinite* (f −‘ $\{y\}$)
⟨*proof*⟩

1.2 Infinitely Many and Almost All

We often need to reason about the existence of infinitely many (resp., all but finitely many) objects satisfying some predicate, so we introduce corresponding binders and their proof rules.

definition
Inf-many :: ($'a \Rightarrow$ *bool*) $\Rightarrow$ *bool* (**binder** *INFM* *10*) **where**
Inf-many P = *infinite* $\{x.\ P\ x\}$

definition
Alm-all :: ($'a \Rightarrow$ *bool*) $\Rightarrow$ *bool* (**binder** *MOST* *10*) **where**
Alm-all P = ($\neg$ (*INFM* $x.\ \neg\ P\ x$))

notation (*xsymbols*)
Inf-many (**binder** $\exists_\infty$ *10*) **and**
Alm-all (**binder** $\forall_\infty$ *10*)

notation (*HTML* **output**)
Inf-many (**binder** $\exists_\infty$ *10*) **and**
Alm-all (**binder** $\forall_\infty$ *10*)

lemma *INFM-EX*:
$(\exists_\infty x.\ P\ x) \Longrightarrow (\exists x.\ P\ x)$
⟨*proof*⟩

lemma *MOST-iff-finiteNeg*: $(\forall_\infty x.\ P\ x) =$ *finite* $\{x.\ \neg\ P\ x\}$
⟨*proof*⟩

lemma *ALL-MOST*: $\forall x.\ P\ x \Longrightarrow \forall_\infty x.\ P\ x$
⟨*proof*⟩

lemma *INFM-mono*:
assumes *inf*: $\exists_\infty x.\ P\ x$ **and** q: $\bigwedge x.\ P\ x \Longrightarrow Q\ x$
shows $\exists_\infty x.\ Q\ x$
⟨*proof*⟩

lemma *MOST-mono*: $\forall_\infty x.\ P\ x \Longrightarrow (\bigwedge x.\ P\ x \Longrightarrow Q\ x) \Longrightarrow \forall_\infty x.\ Q\ x$
⟨*proof*⟩

lemma *INFM-disj-distrib*:
$(\exists_\infty x.\ P\ x \lor Q\ x) \longleftrightarrow (\exists_\infty x.\ P\ x) \lor (\exists_\infty x.\ Q\ x)$
⟨*proof*⟩

lemma *MOST-conj-distrib*:
$(\forall_\infty x.\ P\ x \wedge Q\ x) \longleftrightarrow (\forall_\infty x.\ P\ x) \wedge (\forall_\infty x.\ Q\ x)$
⟨*proof*⟩

lemma *MOST-rev-mp*:
assumes $\forall_\infty x.\ P\ x$ **and** $\forall_\infty x.\ P\ x \longrightarrow Q\ x$
shows $\forall_\infty x.\ Q\ x$
⟨*proof*⟩

lemma *not-INFM* [*simp*]: $\neg\ (INFM\ x.\ P\ x) \longleftrightarrow (MOST\ x.\ \neg\ P\ x)$
⟨*proof*⟩

lemma *not-MOST* [*simp*]: $\neg\ (MOST\ x.\ P\ x) \longleftrightarrow (INFM\ x.\ \neg\ P\ x)$
⟨*proof*⟩

lemma *INFM-const* [*simp*]: $(INFM\ x{::}'a.\ P) \longleftrightarrow P \wedge infinite\ (UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *MOST-const* [*simp*]: $(MOST\ x{::}'a.\ P) \longleftrightarrow P \vee finite\ (UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *INFM-nat*: $(\exists_\infty n.\ P\ (n{::}nat)) = (\forall m.\ \exists n.\ m<n \wedge P\ n)$
⟨*proof*⟩

lemma *INFM-nat-le*: $(\exists_\infty n.\ P\ (n{::}nat)) = (\forall m.\ \exists n.\ m\le n \wedge P\ n)$
⟨*proof*⟩

lemma *MOST-nat*: $(\forall_\infty n.\ P\ (n{::}nat)) = (\exists m.\ \forall n.\ m<n \longrightarrow P\ n)$
⟨*proof*⟩

lemma *MOST-nat-le*: $(\forall_\infty n.\ P\ (n{::}nat)) = (\exists m.\ \forall n.\ m\le n \longrightarrow P\ n)$
⟨*proof*⟩

1.3 Enumeration of an Infinite Set

The set's element type must be wellordered (e.g. the natural numbers).

consts
enumerate :: $'a{::}wellorder\ set => (nat => 'a{::}wellorder)$
primrec
enumerate-0: $enumerate\ S\ 0 = (LEAST\ n.\ n \in S)$
enumerate-Suc: $enumerate\ S\ (Suc\ n) = enumerate\ (S - \{LEAST\ n.\ n \in S\})\ n$

lemma *enumerate-Suc'*:
$enumerate\ S\ (Suc\ n) = enumerate\ (S - \{enumerate\ S\ 0\})\ n$
⟨*proof*⟩

lemma *enumerate-in-set*: $infinite\ S \Longrightarrow enumerate\ S\ n : S$
⟨*proof*⟩

declare *enumerate-0* [*simp del*] *enumerate-Suc* [*simp del*]

lemma *enumerate-step*: *infinite S* $\Longrightarrow$ *enumerate S n* $<$ *enumerate S* (*Suc n*)
⟨*proof*⟩

lemma *enumerate-mono*: *m*<*n* $\Longrightarrow$ *infinite S* $\Longrightarrow$ *enumerate S m* $<$ *enumerate S n*
⟨*proof*⟩

1.4 Miscellaneous

A few trivial lemmas about sets that contain at most one element. These simplify the reasoning about deterministic automata.

definition
atmost-one :: *'a set* $\Rightarrow$ *bool* **where**
atmost-one S = ($\forall$ *x y*. *x*$\in$*S* $\wedge$ *y*$\in$*S* $\longrightarrow$ *x*=*y*)

lemma *atmost-one-empty*: *S* = {} $\Longrightarrow$ *atmost-one S*
⟨*proof*⟩

lemma *atmost-one-singleton*: *S* = {*x*} $\Longrightarrow$ *atmost-one S*
⟨*proof*⟩

lemma *atmost-one-unique* [*elim*]: *atmost-one S* $\Longrightarrow$ *x* $\in$ *S* $\Longrightarrow$ *y* $\in$ *S* $\Longrightarrow$ *y* = *x*
⟨*proof*⟩

end

2 Ramsey's Theorem

theory *Ramsey*
imports *Main Infinite-Set*
begin

declare [[*simp-depth-limit* = *5*]]

2.1 Library lemmas

lemma *infinite-inj-infinite-image*: *infinite Z* ==> *inj-on f Z* ==> *infinite* (*f* ‘ *Z*)
⟨*proof*⟩

lemma *infinite-dom-finite-rng*: [| *infinite A*; *finite* (*f* ‘ *A*) |] ==> *? b* : *f* ‘ *A*. *infinite* {*a* : *A*. *f a* = *b*}
⟨*proof*⟩

lemma *infinite-mem*: *infinite X* ==> *? x*. *x* : *X*

⟨proof⟩

lemma *not-empty-least*: (Y::nat set) ~= {} ==> ? m. m : Y & (! m'. m' : Y --> m <= m')
⟨proof⟩

2.2 Dependent Choice Variant

—

consts *choice* :: ('a => bool) => ('a => 'a => bool) => nat => 'a
primrec
choice P R 0 = (SOME x. P x)
choice P R (Suc n) = (let x = choice P R n in SOME y. P y & R x y)
—

lemma *dc*:
(! x y z. R x y & R y z --> R x z)
& (? x0. P x0)
& (! x. P x --> (? y. P y & R x y))
--> (? f::nat=>'b. (! n. P (f n)) & (! n m. R (f n) (f (n+m+1))))

⟨proof⟩

2.3 Partitions

definition
part :: nat => nat => 'a set => ('a set => nat) => bool **where**
part r s Y f = (! X. X <= Y & finite X & card X = r --> f X < s)

lemma *part*: [| infinite YY; part (Suc n) s YY f; yy : YY |] ==> part n s (YY − {yy}) (%u. f (insert yy u))
⟨proof⟩

lemma *part-subset*: part (Suc n) s YY f ==> Y <= YY ==> part (Suc n) s Y f
⟨proof⟩

2.4 Ramsey's theorem

lemma *ramsey*:
! (s::nat) (r::nat) (YY::'a set) (f::'a set => nat).
infinite YY
& (! X. X <= YY & finite X & card X = r --> f X < s)
--> (? Y' t'.
Y' <= YY
& infinite Y'
& t' < s
& (! X. X <= Y' & finite X & card X = r --> f X = t'))
⟨proof⟩

end