A Probabilistic Proof of the Girth-Chromatic Number Theorem

Lars Noschinski

September 13, 2023

Abstract

This works presents a formalization of the Girth-Chromatic number theorem in graph theory, stating that graphs with arbitrarily large girth and chromatic number exist. The proof uses the theory of Random Graphs to prove the existence with probabilistic arguments and is based on [1].

Contents

theory *Girth-Chromatic-Misc*
imports
Main
HOL−Library.Extended-Real
begin

1 Auxilliary lemmas and setup

This section contains facts about general concepts which are not directly connected to the proof of the Chromatic-Girth theorem. At some point in time, most of them could be moved to the Isabelle base library.

Also, a little bit of setup happens.

1.1 Numbers

lemma *enat-in-Inf*:
fixes S :: *enat set*
assumes $Inf\ S \neq top$
shows $Inf\ S \in S$
$\langle proof \rangle$

lemma *enat-in-INF*:
fixes f :: $'a \Rightarrow enat$
assumes $(INF\ x{\in}\ S.\ f\ x) \neq top$
obtains x **where** $x \in S$ **and** $(INF\ x{\in}\ S.\ f\ x) = f\ x$
$\langle proof \rangle$

lemma *enat-less-INF-I*:
fixes f :: $'a \Rightarrow enat$
assumes *not-inf*: $x \neq \infty$ **and** *less*: $\bigwedge y.\ y \in S \Longrightarrow x < f\ y$
shows $x < (INF\ y{\in}S.\ f\ y)$
$\langle proof \rangle$

lemma *enat-le-Sup-iff*:
$enat\ k \leq Sup\ M \longleftrightarrow k = 0 \vee (\exists\, m \in M.\ enat\ k \leq m)$ (**is** $?L \longleftrightarrow ?R$)
$\langle proof \rangle$

lemma *enat-neq-zero-cancel-iff*[*simp*]:
$0 \neq enat\ n \longleftrightarrow 0 \neq n$
$enat\ n \neq 0 \longleftrightarrow n \neq 0$
$\langle proof \rangle$

lemma *natceiling-lessD*: $nat(ceiling\ x) < n \Longrightarrow x < real\ n$
$\langle proof \rangle$

lemma *le-natceiling-iff*:
fixes n :: *nat* **and** r :: *real*
shows $n \leq r \Longrightarrow n \leq nat(ceiling\ r)$
$\langle proof \rangle$

lemma *natceiling-le-iff*:
fixes n :: *nat* **and** r :: *real*
shows $r \leq n \Longrightarrow nat(ceiling\ r) \leq n$

$\langle proof \rangle$

lemma *dist-real-noabs-less*:
  fixes $a\ b\ c$ :: *real* **assumes** $dist\ a\ b < c$ **shows** $a - b < c$
$\langle proof \rangle$

lemma *n-choose-2-nat*:
  fixes n :: *nat* **shows** $(n\ choose\ 2) = (n * (n - 1))\ div\ 2$
$\langle proof \rangle$

lemma *powr-less-one*:
  fixes x::*real*
  assumes $1 < x\ y < 0$
  shows $x\ powr\ y < 1$
$\langle proof \rangle$

lemma *powr-le-one-le*: $\bigwedge x\ y{::}real.\ 0 < x \Longrightarrow x \leq 1 \Longrightarrow 1 \leq y \Longrightarrow x\ powr\ y \leq x$
$\langle proof \rangle$

1.2 Lists and Sets

lemma *list-set-tl*: $x \in set\ (tl\ xs) \Longrightarrow x \in set\ xs$
$\langle proof \rangle$

lemma *list-exhaust3*:
  obtains $xs = []$ | x **where** $xs = [x]$ | $x\ y\ ys$ **where** $xs = x \# y \# ys$
$\langle proof \rangle$

lemma *card-Ex-subset*:
  $k \leq card\ M \Longrightarrow \exists N.\ N \subseteq M \wedge card\ N = k$
$\langle proof \rangle$

1.3 Limits and eventually

We employ filters and the *eventually* predicate to deal with the $\exists N.\ \forall n{\geq}N.\ P\ n$ cases. To make this more convenient, introduce a shorter syntax.

abbreviation *evseq* :: $(nat \Rightarrow bool) \Rightarrow bool$ (**binder** $\forall^\infty$ *10*) **where**
  $evseq\ P \equiv eventually\ P\ sequentially$

lemma *eventually-le-le*:
  fixes P :: $'a => ('b :: preorder)$
  assumes *eventually* $(\lambda x.\ P\ x \leq Q\ x)$ *net*
  assumes *eventually* $(\lambda x.\ Q\ x \leq R\ \ x)$ *net*
  shows *eventually* $(\lambda x.\ P\ x \leq R\ x)$ *net*
$\langle proof \rangle$

lemma *LIMSEQ-neg-powr*:
  assumes s: $s < 0$
  shows $(\%x.\ (real\ x)\ powr\ s) \longrightarrow 0$

⟨*proof*⟩

lemma *LIMSEQ-inv-powr*:
  assumes $0 < c$ $0 < d$
  shows $(\lambda n :: nat.\ (c\ /\ n)\ powr\ d) \longrightarrow 0$
⟨*proof*⟩

end
theory *Ugraphs*
imports
  Girth-Chromatic-Misc
begin

2 Undirected Simple Graphs

In this section, we define some basics of graph theory needed to formalize the Chromatic-Girth theorem.

For readability, we introduce synonyms for the types of vertexes, edges, graphs and walks.

type-synonym *uvert* = *nat*
type-synonym *uedge* = *nat set*
type-synonym *ugraph* = *uvert set* × *uedge set*
type-synonym *uwalk* = *uvert list*

abbreviation *uedges* :: *ugraph* ⇒ *uedge set* **where**
  uedges G ≡ *snd G*

abbreviation *uverts* :: *ugraph* ⇒ *uvert set* **where**
  uverts G ≡ *fst G*

fun *mk-uedge* :: *uvert* × *uvert* ⇒ *uedge* **where**
   mk-uedge (*u*,*v*) = {*u*,*v*}

All edges over a set of vertexes S:

definition *all-edges S* ≡ *mk-uedge* ‘ {*uv* ∈ *S* × *S*. *fst uv* ≠ *snd uv*}

definition *uwellformed* :: *ugraph* ⇒ *bool* **where**
  uwellformed G ≡ (∀ *e*∈*uedges G*. *card e* = *2* ∧ (∀ *u* ∈ *e*. *u* ∈ *uverts G*))

fun *uwalk-edges* :: *uwalk* ⇒ *uedge list* **where**
    uwalk-edges [] = []
  | *uwalk-edges* [*x*] = []
  | *uwalk-edges* (*x* # *y* # *ys*) = {*x*,*y*} # *uwalk-edges* (*y* # *ys*)

definition *uwalk-length* :: *uwalk* ⇒ *nat* **where**
  uwalk-length p ≡ *length* (*uwalk-edges p*)

definition *uwalks* :: *ugraph* $\Rightarrow$ *uwalk set* **where**
uwalks G $\equiv$ $\{p.\ set\ p \subseteq uverts\ G \wedge set\ (uwalk\text{-}edges\ p) \subseteq uedges\ G \wedge p \neq []\}$

definition *ucycles* :: *ugraph* $\Rightarrow$ *uwalk set* **where**
ucycles G $\equiv$ $\{p.\ uwalk\text{-}length\ p \geq 3 \wedge p \in uwalks\ G \wedge distinct\ (tl\ p) \wedge hd\ p = last\ p\}$

definition *remove-vertex* :: *ugraph* $\Rightarrow$ *nat* $\Rightarrow$ *ugraph* (- −− - [60,60] 60) **where**
remove-vertex G u $\equiv$ $(uverts\ G - \{u\},\ uedges\ G - \{A \in uedges\ G.\ u \in A\})$

2.1 Basic Properties

lemma *uwalk-length-conv*: $uwalk\text{-}length\ p = length\ p - 1$
⟨*proof*⟩

lemma *all-edges-mono*:
$vs \subseteq ws \Longrightarrow all\text{-}edges\ vs \subseteq all\text{-}edges\ ws$
⟨*proof*⟩

lemma *all-edges-subset-Pow*: $all\text{-}edges\ A \subseteq Pow\ A$
⟨*proof*⟩

lemma *in-mk-uedge-img*: $(a,b) \in A \vee (b,a) \in A \Longrightarrow \{a,b\} \in mk\text{-}uedge\ `\ A$
⟨*proof*⟩

lemma *in-mk-uedge-img-iff*: $\{a,b\} \in mk\text{-}uedge\ `\ A \longleftrightarrow (a,b) \in A \vee (b,a) \in A$
⟨*proof*⟩

lemma *distinct-edgesI*:
assumes *distinct p* **shows** *distinct* (*uwalk-edges p*)
⟨*proof*⟩

lemma *finite-ucycles*:
assumes *finite* (*uverts G*)
shows *finite* (*ucycles G*)
⟨*proof*⟩

lemma *ucycles-distinct-edges*:
assumes $c \in ucycles\ G$ **shows** *distinct* (*uwalk-edges c*)
⟨*proof*⟩

lemma *card-left-less-pair*:
fixes A :: ($'a$ :: *linorder*) *set*
assumes *finite A*
shows $card\ \{(a,b).\ a \in A \wedge b \in A \wedge a < b\}$
$= (card\ A * (card\ A - 1))\ div\ 2$
⟨*proof*⟩

lemma *card-all-edges*:
assumes *finite A*
shows *card* (*all-edges A*) = *card A choose 2*
⟨*proof*⟩

lemma *verts-Gu*: *uverts* (*G* −− *u*) = *uverts G* − {*u*}
⟨*proof*⟩

lemma *edges-Gu*: *uedges* (*G* −− *u*) ⊆ *uedges G*
⟨*proof*⟩

2.2 Girth, Independence and Vertex Colorings

definition *girth* :: *ugraph* ⇒ *enat* **where**
girth G ≡ *INF p*∈ *ucycles G*. *enat* (*uwalk-length p*)

definition *independent-sets* :: *ugraph* ⇒ *uvert set set* **where**
independent-sets Gr ≡ {*vs*. *vs* ⊆ *uverts Gr* ∧ *all-edges vs* ∩ *uedges Gr* = {}}

definition α :: *ugraph* ⇒ *enat* **where**
α *G* ≡ *SUP vs* ∈ *independent-sets G*. *enat* (*card vs*)

definition *vertex-colorings* :: *ugraph* ⇒ *uvert set set set* **where**
vertex-colorings G ≡ {*C*. ⋃ *C* = *uverts G* ∧ (∀ *c1*∈*C*. ∀ *c2*∈*C*. *c1* ≠ *c2* ⟶ *c1* ∩ *c2* = {}) ∧
(∀ *c*∈*C*. *c* ≠ {} ∧ (∀ *u* ∈ *c*. ∀ *v* ∈ *c*. {*u*,*v*} ∉ *uedges G*))}

The chromatic number χ:

definition *chromatic-number* :: *ugraph* ⇒ *enat* **where**
chromatic-number G ≡ *INF c*∈ (*vertex-colorings G*). *enat* (*card c*)

lemma *independent-sets-mono*:
vs ∈ *independent-sets G* ⟹ *us* ⊆ *vs* ⟹ *us* ∈ *independent-sets G*
⟨*proof*⟩

lemma *le-α-iff*:
assumes *0* < *k*
shows *k* ≤ α *Gr* ⟷ *k* ∈ *card* ‘ *independent-sets Gr* (**is** *?L* ⟷ *?R*)
⟨*proof*⟩

lemma *zero-less-α*:
assumes *uverts G* ≠ {}
shows *0* < α *G*
⟨*proof*⟩

lemma *α-le-card*:
assumes *finite* (*uverts G*)
shows α *G* ≤ *card*(*uverts G*)
⟨*proof*⟩

lemma *α-fin*: *finite* (*uverts* G) $\Longrightarrow$ α $G \neq \infty$
⟨*proof*⟩

lemma *α-remove-le*:
shows α (G −− u) $\leq$ α G
⟨*proof*⟩

A lower bound for the chromatic number of a graph can be given in terms of the independence number

lemma *chromatic-lb*:
assumes *wf-G*: *uwellformed* G
and *fin-G*: *finite* (*uverts* G)
and *neG*: *uverts* $G \neq \{\}$
shows *card* (*uverts* G) / α $G \leq$ *chromatic-number* G
⟨*proof*⟩

end
theory *Girth-Chromatic*
imports
Ugraphs
Girth-Chromatic-Misc
HOL−Probability.Probability
HOL−Decision-Procs.Approximation
begin

3 Probability Space on Sets of Edges

definition *cylinder* :: *'a set* ⇒ *'a set* ⇒ *'a set* ⇒ *'a set set* **where**
cylinder S A B = $\{T \in$ *Pow* $S.$ $A \subseteq T \wedge B \cap T = \{\}\}$

lemma *full-sum*:
fixes p :: *real*
assumes *finite* S
shows ($\sum A \in$*Pow* S. p^*card* A $*$ (1 − p)^*card* (S − A)) = 1
⟨*proof*⟩

Definition of the probability space on edges:

locale *edge-space* =
fixes n :: *nat* **and** p :: *real*
assumes *p-prob*: $0 \leq p$ $p \leq 1$
begin

definition *S-verts* :: *nat set* **where**
S-verts ≡ $\{1..n\}$

definition *S-edges* :: *uedge set* **where**
S-edges = *all-edges S-verts*

definition *edge-ugraph* :: *uedge set* ⇒ *ugraph* **where**
edge-ugraph es ≡ (*S-verts*, *es* ∩ *S-edges*)

definition *P* = *point-measure* (*Pow S-edges*) (λ*s*. *p*^*card s* ∗ (*1* − *p*)^*card* (*S-edges* − *s*))

lemma *finite-verts*[*intro*!]: *finite S-verts*
⟨*proof*⟩

lemma *finite-edges*[*intro*!]: *finite S-edges*
⟨*proof*⟩

lemma *finite-graph*[*intro*!]: *finite* (*uverts* (*edge-ugraph es*))
⟨*proof*⟩

lemma *uverts-edge-ugraph*[*simp*]: *uverts* (*edge-ugraph es*) = *S-verts*
⟨*proof*⟩

lemma *uedges-edge-ugraph*[*simp*]: *uedges* (*edge-ugraph es*) = *es* ∩ *S-edges*
⟨*proof*⟩

lemma *space-eq*: *space P* = *Pow S-edges* ⟨*proof*⟩

lemma *sets-eq*: *sets P* = *Pow* (*Pow S-edges*) ⟨*proof*⟩

lemma *emeasure-eq*:
emeasure P A = (*if A* ⊆ *Pow S-edges then* (∑ *edges*∈*A*. *p*^*card edges* ∗ (*1* − *p*)^*card* (*S-edges* − *edges*)) *else 0*)
⟨*proof*⟩

lemma *integrable-P*[*intro*, *simp*]: *integrable P* (*f*::- ⇒ *real*)
⟨*proof*⟩

lemma *borel-measurable-P*[*measurable*]: *f* ∈ *borel-measurable P*
⟨*proof*⟩

lemma *prob-space-P*: *prob-space P*
⟨*proof*⟩

end

sublocale *edge-space* ⊆ *prob-space P*
⟨*proof*⟩

context *edge-space*
begin

lemma *prob-eq*:

prob A = (*if A* ⊆ *Pow S-edges then* ($\sum$ *edges*∈*A*. *p*^*card edges* ∗ (*1* − *p*)^*card* (*S-edges* − *edges*)) *else 0*)
⟨*proof*⟩

lemma *integral-finite-singleton*: *integral*^{L} *P f* = ($\sum$ *x*∈*Pow S-edges*. *f x* ∗ *measure P* {*x*})
⟨*proof*⟩

Probability of cylinder sets:

lemma *cylinder-prob*:
assumes *A* ⊆ *S-edges B* ⊆ *S-edges A* ∩ *B* = {}
shows *prob* (*cylinder S-edges A B*) = *p* ^ (*card A*) ∗ (*1* − *p*) ^ (*card B*) (**is** - = *?pp A B*)
⟨*proof*⟩

lemma *Markov-inequality*:
fixes *a* :: *real* **and** *X* :: *uedge set* ⇒ *real*
assumes *0* < *c* ⋀*x*. *0* ≤ *f x*
shows *prob* {*x* ∈ *space P*. *c* ≤ *f x*} ≤ (∫ *x*. *f x* ∂ *P*) / *c*
⟨*proof*⟩

end

3.1 Graph Probabilities outside of *Edge-Space* locale

These abbreviations allow a compact expression of probabilities about random graphs outside of the *Edge-Space* locale. We also transfer a few of the lemmas we need from the locale into the toplevel theory.

abbreviation *MGn* :: (*nat* ⇒ *real*) ⇒ *nat* ⇒ (*uedge set*) *measure* **where**
MGn p n ≡ (*edge-space.P n* (*p n*))
abbreviation *probGn* :: (*nat* ⇒ *real*) ⇒ *nat* ⇒ (*uedge set* ⇒ *bool*) ⇒ *real* **where**
probGn p n P ≡ *measure* (*MGn p n*) {*es* ∈ *space* (*MGn p n*). *P es*}

lemma *probGn-le*:
assumes *p-prob*: *0* < *p n p n* < *1*
assumes *sub*: ⋀*n es*. *es* ∈ *space* (*MGn p n*) ⟹ *P n es* ⟹ *Q n es*
shows *probGn p n* (*P n*) ≤ *probGn p n* (*Q n*)
⟨*proof*⟩

4 Short cycles

definition *short-cycles* :: *ugraph* ⇒ *nat* ⇒ *uwalk set* **where**
short-cycles G k ≡ {*p* ∈ *ucycles G*. *uwalk-length p* ≤ *k*}

obtains a vertex in a short cycle:

definition *choose-v* :: *ugraph* ⇒ *nat* ⇒ *uvert* **where**
choose-v G k ≡ *SOME u*. ∃ *p*. *p* ∈ *short-cycles G k* ∧ *u* ∈ *set p*

partial-function (*tailrec*) *kill-short* :: *ugraph* ⇒ *nat* ⇒ *ugraph* **where**
kill-short G k = (*if short-cycles G k* = {} *then G else* (*kill-short* (*G* −− (*choose-v G k*)) *k*))

lemma *ksc-simps*[*simp*]:
short-cycles G k = {} ⟹ *kill-short G k* = *G*
short-cycles G k ≠ {} ⟹ *kill-short G k* = *kill-short* (*G* −− (*choose-v G k*)) *k*
⟨*proof*⟩

lemma
assumes *short-cycles G k* ≠ {}
shows *choose-v--in-uverts*: *choose-v G k* ∈ *uverts G* (**is** *?t1*)
and *choose-v--in-short*: ∃ *p*. *p* ∈ *short-cycles G k* ∧ *choose-v G k* ∈ *set p* (**is** *?t2*)
⟨*proof*⟩

lemma *kill-step-smaller*:
assumes *short-cycles G k* ≠ {}
shows *short-cycles* (*G* −− (*choose-v G k*)) *k* ⊂ *short-cycles G k*
⟨*proof*⟩

Induction rule for *kill-short*:

lemma *kill-short-induct*[*consumes 1*, *case-names empty kill-vert*]:
assumes *fin*: *finite* (*uverts G*)
assumes *a-empty*: ⋀*G*. *short-cycles G k* = {} ⟹ *P G k*
assumes *a-kill*: ⋀*G*. *finite* (*short-cycles G k*) ⟹ *short-cycles G k* ≠ {}
⟹ *P* (*G* −− (*choose-v G k*)) *k* ⟹ *P G k*
shows *P G k*
⟨*proof*⟩

Large Girth (after *kill-short*):

lemma *kill-short-large-girth*:
assumes *finite* (*uverts G*)
shows *k* < *girth* (*kill-short G k*)
⟨*proof*⟩

Order of graph (after *kill-short*):

lemma *kill-short-order-of-graph*:
assumes *finite* (*uverts G*)
shows *card* (*uverts G*) − *card* (*short-cycles G k*) ≤ *card* (*uverts* (*kill-short G k*))
⟨*proof*⟩

Independence number (after *kill-short*):

lemma *kill-short-α*:
assumes *finite* (*uverts G*)
shows α (*kill-short G k*) ≤ α *G*
⟨*proof*⟩

Wellformedness (after *kill-short*):

lemma *kill-short-uwellformed*:
  assumes *finite* (*uverts G*) *uwellformed G*
  shows *uwellformed* (*kill-short G k*)
⟨*proof*⟩

5 The Chromatic-Girth Theorem

Probability of Independent Edges:

lemma (**in** *edge-space*) *random-prob-independent*:
  assumes $n \geq k$ $k \geq 2$
  shows $prob\ \{es \in space\ P.\ k \leq \alpha\ (edge\text{-}ugraph\ es)\}$
    $\leq (n\ choose\ k){*}(1{-}p)\widehat{\ }(k\ choose\ 2)$
⟨*proof*⟩

Almost never many independent edges:

lemma *almost-never-le-α*:
  fixes $k :: nat$
    and $p :: nat \Rightarrow real$
  assumes *p-prob*: $\forall^\infty\ n.\ 0 < p\ n \wedge p\ n < 1$
  assumes [*arith*]: $k > 0$
  assumes *N-prop*: $\forall^\infty\ n.\ (6 * k * ln\ n)/n \leq p\ n$
  shows $(\lambda n.\ probGn\ p\ n\ (\lambda es.\ 1/2{*}n/k \leq \alpha\ (edge\text{-}space.edge\text{-}ugraph\ n\ es)))$
$\longrightarrow 0$
    (**is** $(\lambda n.\ ?prob\text{-}fun\ n) \longrightarrow 0$)
⟨*proof*⟩

Mean number of k-cycles in a graph. (Or rather of paths describing a circle of length k):

lemma (**in** *edge-space*) *mean-k-cycles*:
  assumes $3 \leq k$ $k < n$
  shows $(\int es.\ card\ \{c \in ucycles\ (edge\text{-}ugraph\ es).\ uwalk\text{-}length\ c = k\}\ \partial\ P)$
    $= of\text{-}nat\ (fact\ n\ div\ fact\ (n - k)) * p\ \widehat{\ }\ k$
⟨*proof*⟩

Girth-Chromatic number theorem:

theorem *girth-chromatic*:
  fixes $l :: nat$
  shows $\exists\, G.\ uwellformed\ G \wedge l < girth\ G \wedge l < chromatic\text{-}number\ G$
⟨*proof*⟩

end

References

[1] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4 edition, 2010. http://diestel-graph-theory.com.