Properties of Random Graphs – Subgraph Containment

Lars Hupel

April 20, 2020

Abstract

Random graphs are graphs with a fixed number of vertices, where each edge is present with a fixed probability. We are interested in the probability that a random graph contains a certain pattern, for example a cycle or a clique. A very high edge probability gives rise to perhaps too many edges (which degrades performance for many algorithms), whereas a low edge probability might result in a disconnected graph. We prove a theorem about a threshold probability such that a higher edge probability will asymptotically almost surely produce a random graph with the desired subgraph.

Contents

1 Introduction 2

2 Miscellaneous and contributed lemmas 2

3 Lemmas about probabilities 11
   3.1 Indicator variables and valid probability values . . . . . . . . 11
   3.2 Expectation and variance . . . . . . . . . . . . . . . . . . . . 12
   3.3 Sets of indicator variables . . . . . . . . . . . . . . . . . . . . 14

4 Lemmas about undirected graphs 18
   4.1 Subgraphs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
   4.2 Induced subgraphs . . . . . . . . . . . . . . . . . . . . . . . . 21
   4.3 Graph isomorphism . . . . . . . . . . . . . . . . . . . . . . . . 22
   4.4 Isomorphic subgraphs . . . . . . . . . . . . . . . . . . . . . . 26
   4.5 Density . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
   4.6 Fixed selectors . . . . . . . . . . . . . . . . . . . . . . . . . . 31

5 Classes and properties of graphs 32

6 The subgraph threshold theorem 33
1 Introduction

Random graphs have been introduced by Erdős and Rényi in [2]. They describe a probability space where, for a fixed number of vertices, each possible edge is present with a certain probability independent from other edges, but with the same probability for each edge. They study what properties emerge when increasing the number of vertices, or as they call it, “the evolution of such a random graph”. The theorem which we will prove here is a slightly different version from that in the first section of that paper.

Here, we are interested in the probability that a random graph contains a certain pattern, for example a cycle or a clique. A very high edge probability gives rise to perhaps too many edges, which is usually undesired since it degrades the performance of many algorithms, whereas a low edge probability might result in a disconnected graph. The central theorem determines a threshold probability such that a higher edge probability will asymptotically almost surely produce a random graph with the desired subgraph.

The proof is outlined in [1, § 11.4] and [3, § 3]. The work is based on the comprehensive formalization of probability theory in Isabelle/HOL and on a previous definition of graphs in a work by Noschinski [4]. There, Noschinski formalized the proof that graphs with arbitrarily large girth and chromatic number exist. While the proof in this paper uses a different approach, the definition of a probability space on edges turned out to be quite useful.

2 Miscellaneous and contributed lemmas

theory Ugraph-Misc
imports
  HOL−Probability.Probability
  Girth-Chromatic.Girth-Chromatic-Misc
begin

lemma sum-square:
  fixes a :: i ⇒ a :: {monoid-mult, semiring-0}
  shows (∑ i ∈ I. a i) ∗ 2 = (∑ i ∈ I. ∑ j ∈ I. a i ∗ a j)
  by (simp only: sum-product power2-eq-square)

lemma sum-split:
  finite I ⟹
  (∑ i ∈ I. if p i then f i else g i) = (∑ i ∈ I ∧ p i. f i) + (∑ i ∈ I ∧ ¬ p i. g i)
  by (simp add: sum.If-cases Int-def)

lemma sum-split2:
  assumes finite I
  shows (∑ i ∈ I ∧ P i. if Q i then f i else g i) = (∑ i ∈ I ∧ P i ∧ Q i. f i) + (∑ i ∈ I ∧ P i ∧ ¬ Q i. g i)
proof (subst sum.If-cases)
  show finite \{i ∈ I. P i\}
  using assms by simp

  have \{i ∈ I. P i\} ∩ Collect Q = \{i ∈ I. P i ∧ Q i\} \{i ∈ I. P i\} ∩ − Collect Q = \{i ∈ I. P i ∧ ∼ Q i\}
    by auto
  thus sum f \{i ∈ I. P i ∧ Q i\} + sum g \{i ∈ I. P i ∧ ∼ Q i\}
    by presburger
qed

lemma sum-upper:
  fixes f :: 'i ⇒ 'a :: ordered-comm-monoid-add
  assumes finite I i. i ∈ I ⇒ 0 ≤ f i
  shows (∑ i | i ∈ I ∧ P i. f i) ≤ sum f I
proof —
  have sum f I = (∑ i ∈ I. if P i then f i else f i)
    by simp
  hence sum f I = (∑ i | i ∈ I ∧ P i. f i) + (∑ i | i ∈ I ∧ ∼P i. f i)
    by (simp only: sum-split[OF ⟨finite I⟩])
  moreover have 0 ≤ (∑ i | i ∈ I ∧ ∼P i. f i)
    by (rule sum-nonneg) (simp add: assms)
  ultimately show ?thesis
    by (metis (full-types) add.comm-neutral add-left-mono)
qed

lemma sum-lower-or-eq:
  fixes f :: 'i ⇒ 'a :: ordered-comm-monoid-add
  assumes finite I i. i ∈ I ⇒ 0 ≤ f i x ≤ f i
  shows x ≤ sum f I
proof —
  have x < f i by fact
  also have ... ≤ sum f I
    by (simp add: assms)
  finally show ?thesis.
qed

lemma sum-lower-or-eq:
  fixes f :: 'i ⇒ 'a :: ordered-comm-monoid-add
  assumes finite I i. i ∈ I ⇒ 0 ≤ f i x ≤ f i
  shows x ≤ sum f I
proof —
  have x ≤ f i by fact
  also have ... ≤ sum f I
    by (simp add: assms)
  finally show ?thesis.
qed
lemma sum-left-div-distrib:
  fixes f :: 'i ⇒ real
  shows (∑ i ∈ I. f i / x) = sum f I / x
proof
  have (∑ i ∈ I. f i / x) = (∑ i ∈ I. f i * (1 / x))
    by simp
  also have ... = sum f I * (1 / x)
    by (rule sum-distrib-right[symmetric])
  also have ... = sum f I / x
    by simp
finally show ?thesis.
qed

lemma powr-mono3:
  fixes x :: real
  assumes 0 < x x < 1 b ≤ a
  shows x powr a ≤ x powr b
proof
  have x powr a = 1 / x powr −a
    by (simp add: powr-minus-divide)
  also have ... = (1 / x) powr −a
    using assms by (simp add: powr-divide)
  also have ... ≤ (1 / x) powr −b
    using assms by (simp add: powr-mono)
  also have ... = 1 / x powr −b
    using assms by (simp add: powr-divide)
  also have ... = x powr b
    by (simp add: powr-minus-divide)
finally show ?thesis.
qed

lemma card-union: finite A ⇒ finite B ⇒ card (A ∪ B) = card A + card B - card (A ∩ B)
by (metis card-Un-Int[symmetric] diff-add-inverse2)

lemma card-1-element:
  assumes card E = 1
  shows ∃ a. E = {a}
proof
  from assms obtain a where a ∈ E
    by force
  let ?E' = E - {a}
  have finite ?E'
    using assms card-ge-0-finite by force
  hence card (insert a ?E') = 1 + card ?E'
    using card-insert by fastforce
moreover have $E = \text{insert} \ a \ ?E'$
  using $(a \in E)$ by blast
ultimately have $\text{card} \ E = 1 + \text{card} \ ?E'$
  by simp
hence $\text{card} \ ?E' = 0$
  using assms by simp
hence $?E' = \{\}$
  using $\text{finite} \ ?E'$ by simp
thus $?\text{thesis}$
  using $(a \in E)$ by blast
qed

lemma card-2-elements:
  assumes $\text{card} \ E = 2$
  shows $\exists \ a \ b. \ E = \{a, b\} \land a \neq b$
proof
  from assms obtain $a$ where $a \in E$
    by force
  let $?E' = E - \{a\}$
  have $\text{finite} \ ?E'$
    using assms card-ge-0-finite by force
  hence $\text{card} (\text{insert} \ a \ ?E') = 1 + \text{card} \ ?E'$
    using card-insert by fastforce
  moreover have $E = \text{insert} \ a \ ?E'$
    using $(a \in E)$ by blast
  ultimately have $\text{card} \ E = 1 + \text{card} \ ?E'$
    by simp
  hence $\text{card} \ ?E' = 1$
    using assms by simp
  then obtain $b$ where $?E' = \{b\}$
    using card-1-element by blast
  hence $E = \{a, b\}$
    using $(a \in E)$ by blast
  moreover have $a \neq b$
    using $(?E' = \{b\})$ by blast
  ultimately show $?\text{thesis}$
    by blast
qed

lemma bij-lift:
  assumes bij-betw $f \ A \ B$
  shows bij-betw $(\lambda e. \ f' e) \ (\text{Pow} \ A) \ (\text{Pow} \ B)$
proof
  have $f \colon \text{inj-on} \ f \ A \ f' \ A = B$
    using assms unfolding bij-betw-def by simp-all
  have inj-on $(\lambda e. \ f' e) \ (\text{Pow} \ A)$
    unfolding inj-on-def by clarify (metis $f(1)$ inv-into-image-cancel)
  moreover have $(\lambda e. \ f' e) \ (\text{Pow} \ A) = (\text{Pow} \ B)$
ultimately show thesis unfolding bij-betw-def by simp qed

lemma card-inj-subs: inj-on f A \Rightarrow B \subseteq A \Rightarrow card (f ' B) = card B by (metis card-image subset-inj-on)

lemma image-comp-cong: (\(\forall \ a. \ a \in A \Rightarrow f \ a = f \ (g \ a)\)) \Rightarrow f ' A = f ' (g ' A) by auto

abbreviation less-fun :: (nat \Rightarrow real) \Rightarrow (nat \Rightarrow real) \Rightarrow bool (infix \langle 50 \rangle) where f \langle g \equiv (\lambda n. \ f \ n / g \ n) \rightarrow 0

context fixes f :: nat \Rightarrow real begin

lemma LIMSEQ-power-zero: f \rightarrow 0 \Rightarrow 0 < n \Rightarrow (\lambda x. f \ x ^ n :: real) \rightarrow 0 by (metis power-eq-0-iff tendsto-power)

lemma LIMSEQ-cong: assumes f \rightarrow x \forall \infty n. f n = g n shows g \rightarrow x by (rule real-tendsto-sandwich[where f = f and h = f, OF eventually-mono[OF assms(2)] eventually-mono[OF assms(2)]] (auto simp: assms(1))

print-statement Lim-transform-eventually

lemma LIMSEQ-le-zero: assumes g \rightarrow 0 \forall \infty n. 0 \leq f n \forall \infty n. f n \leq g n shows f \rightarrow 0 by (rule real-tendsto-sandwich[OF assms(2) assms(3) tendsto-const assms(1)])

lemma LIMSEQ-const-mult: assumes f \rightarrow a shows (\lambda x. c * f x) \rightarrow c * a by (rule tendsto-mult[OF tendsto-const[where k = c] assms])

lemma LIMSEQ-const-div: assumes f \rightarrow a c \neq 0 shows (\lambda x. f x / c) \rightarrow a / c using LIMSEQ-const-mult[where c = 1/c] assms by simp

end

lemma quot-bounds: fixes x :: 'a :: linordered-field
assumes \( x \leq x' \ y' \leq y \ 0 < y \ 0 \leq x \ 0 < y' \)
shows \( x / y \leq x' / y' \)

proof (rule order-trans)
have \( 0 \leq y \)
  using assms by simp
thus \( x / y \leq x' / y \)
  using assms by (simp add: divide-right-mono)

next
have \( 0 \leq x' \)
  using assms by simp
moreover have \( 0 < y' \)
  using assms by simp
ultimately show \( x' / y \leq x' / y' \)
  using assms by (simp add: divide-left-mono)
qed

lemma less-fun-bounds:
assumes \( f' \ll g' \ \forall n. \ f n \leq f' n \ \forall n. \ g' n \leq g n \ \forall n. \ 0 < g n \ \forall n. \ 0 < g' n \)
shows \( f \ll g \)
proof (rule real-tendsto-sandwich)
show \( \forall n. \ 0 \leq f n / g n \)
  using assms (4,5) by eventually-elim simp
next
show \( \forall n. \ f n / g n \leq f' n / g' n \)
  using assms (2-) by eventually-elim (simp only: quot-bounds)
qed (auto intro: assms (1))

lemma less-fun-const-quot:
assumes \( f \ll g \ c \neq 0 \)
shows \( (\lambda n. \ b * f n) \ll (\lambda n. \ c * g n) \)
proof
  have \( \lambda n. \ (b * (f n / g n)) / c \longrightarrow (b * 0) / c \)
  using assms by (rule LIMSEQ-const-div [OF LIMSEQ-const-mult])
  hence \( \lambda n. \ (b * (f n / g n)) / c \longrightarrow 0 \)
  by simp
  with eventually-sequentiallyI show ?thesis
  by (fastforce intro: Lim-transform-sequentially)
qed

lemma partition-set-of-intersecting-sets-by-card:
assumes finite \( A \)
shows \( \{ B. \ A \cap B \neq \{} \} = (\bigcup n \in \{1..\text{card} \ A\}. \ \{ B. \ \text{card} \ (A \cap B) = n \}) \)
proof (rule set-eqI, rule iffI)
fix \( B \)
assume \( B \in \{ B. \ A \cap B \neq \{} \}
hence \( 0 < \text{card} \ (A \cap B) \)
  using assms by auto
moreover have \( \text{card} \ (A \cap B) \leq \text{card} \ A \)
using assms by (simp add: card-mono)
ultimately have \( \text{card} (A \cap B) \in \{1..\text{card} A \} \)
by simp
thus \( B \in (\bigcup n \in \{1..\text{card} A \}. \{ B. \text{card} (A \cap B) = n \}) \)
by blast
qed force

lemma card-set-of-intersecting-sets-by-card:
 assumes \( A \subseteq I \) finite \( I \) \( k \leq n \leq \text{card} I \) \( k \leq \text{card} A \)
 shows \( \text{card} \{ B. B \subseteq I \land \text{card} B = n \land \text{card} (A \cap B) = k \} = (\text{card} A \text{ choose } k) \ast ((\text{card} I - \text{card} A) \text{ choose } (n - k)) \)
proof –
 note finite-A = finite-subset[OF assms(1,2)]
 have \( \text{card} \{ B. B \subseteq I \land \text{card} B = n \land \text{card} (A \cap B) = k \} = \text{card} \{ K. K \subseteq A \land \text{card} K = k \} \times \{ B'. B' \subseteq I - A \land \text{card} B' = n - k \} \) (is \( ?\text{lhs} = \text{card} ?\text{rhs} \))
 proof (rule bij-betw-same-card[ symmetric])
 let \( ?f = \lambda (K, B'). K \cup B' \)
 have inj-on \( ?f \) ?rhs
 by (blast intro: inj-onI)
 moreover have \( ?f ' ?rhs = ?\text{lhs} \)
 proof (rule set-eqI, rule iffI)
 fix \( B \)
 assume \( B \in ?f ' ?rhs \)
 then obtain \( K B \) where
 \( K \subseteq A \) \( \text{card} K = k \) \( B' \subseteq I - A \) \( \text{card} B' = n - k \)
 by blast
 show \( B \in ?\text{lhs} \)
 proof safe
 fix \( x \)
 assume \( x \in B \) thus \( x \in I \)
 using \( K \langle A \subseteq I \rangle \) by blast
 next
 have \( \text{card} B = \text{card} K + \text{card} B' - \text{card} (K \cap B') \)
 using \( K \) assms by (metis card-union finite-A finite-subset finite-Diff)
 moreover have \( K \cap B' = \{} \)
 using \( K \) assms by blast
 ultimately show \( \text{card} B = n \)
 using \( K \) assms by simp
 next
 have \( A \cap B = K \)
 using \( K \) assms(l1) by blast
 thus \( \text{card} (A \cap B) = k \)
 using \( K \) by simp
 qed

next
 fix \( B \)
 assume \( B \in ?\text{lhs} \)
 hence \( B \subseteq I \) \( \text{card} B = n \) \( \text{card} (A \cap B) = k \)
let \( \mathcal{K} = A \cap B \)
let \( \mathcal{B}' = B - A \)

have \( \mathcal{K} \subseteq A \cap \text{card } \mathcal{K} = k \cap \mathcal{B}' \subseteq I - A \)
using \( B \) by \text{auto}

moreover have \( \text{card } \mathcal{B}' = n - k \)
using \( B \) \text{finite-A assms(1)} by \text{metis Int-commute card-Diff-subset-Int finite-Un inf.left-idem le-iff-inf sup-absorb2)

ultimately have \( \mathcal{K}, \mathcal{B}' \in \text{?rhs} \)
by \text{blast}

moreover have \( B = \mathcal{f}(\mathcal{K}, \mathcal{B}') \)
by \text{auto}

ultimately show \( B \in \mathcal{f} \cdot \text{?rhs} \)
by \text{blast}

qed

ultimately show \( \text{bij-betw } \mathcal{f} \text{ ?rhs ?lhs} \)
unfolding \text{bij-betw-def} ..

qed

also have \( \ldots = \sum K | K \subseteq A \wedge \text{card } K = k \cdot \text{card } \{ \mathcal{B}', \mathcal{B}' \subseteq I - A \wedge \text{card } B' = n - k \} \)

proof \text{(rule card-SigmaI, safe)}

show \( \text{finite } \{ K. K \subseteq A \wedge \text{card } K = k \} \)
by \text{(blast intro: finite-subset[where \( B = \text{Pow } A \) finite-A])}

next

fix \( K \)

assume \( K \subseteq A \)

thus \( \text{finite } \{ \mathcal{B}', \mathcal{B}' \subseteq I - A \wedge \text{card } B' = n - k \} \)
using \( \text{assms by \text{auto}} \)

qed

also have \( \ldots = \text{card } \{ K. K \subseteq A \wedge \text{card } K = k \} \cdot \text{card } \{ \mathcal{B}', \mathcal{B}' \subseteq I - A \wedge \text{card } B' = n - k \} \)
by \text{simp}

also have \( \ldots = (\text{card } A \text{ choose } k) \cdot (\text{card } (I - A) \text{ choose } (n - k)) \)
by \text{simpl only: n-subsets[OF finite-A] n-subsets[OF finite-Diff[OF assms(2)]]}

also have \( \ldots = (\text{card } A \text{ choose } k) \cdot ((\text{card } I - \text{ card } A) \text{ choose } (n - k)) \)
by \text{simp only: card-Diff-subset[OF finite-A assms(1)]}

finally show \( ?\text{thesis} \)

qed

lemma \text{card-dep-pair-set:}

assumes \( \text{finite } A \wedge a. a \subseteq A \implies \text{finite } (f a) \)

shows \( \text{card } \{ (a, b). a \subseteq A \wedge \text{card } a = n \wedge b \subseteq f a \wedge \text{card } b = g a \} = \sum a \cdot a \subseteq A \wedge \text{card } a = n \cdot \text{card } (f a) \text{ choose } g a \) \text{is card ?S = ?C} \)

proof –

have \( S: ?S = \text{Sigma } \{ a. a \subseteq A \wedge \text{card } a = n \} \cdot (\lambda a. \{ b. b \subseteq f a \wedge \text{card } b = g a \} \text{is } \text{Sigma } ?A ?B) \)
by \text{auto}
have \( \text{card} \left( \Sigma A \cap B \right) = \sum a \in \{ a \mid a \subseteq A \land \text{card } a = n \}. \text{card } \left( A \setminus B \right) \)

proof (rule card-SigmaI, safe)

  show \( \text{finite } A \)
    by (rule finite-subset[OF - finite-Collect-subsets[OF assms]]) blast

next

  fix \( a \)
  assume \( a \subseteq A \)
  hence \( \text{finite } (f a) \)
    by (fact assms)
  thus \( \text{finite } (A \setminus B a) \)
    by (rule finite-subset[rotated, OF finite-Collect-subsets]) blast

qed

also have \( \ldots = ?C \)

proof (rule sum.cong)

  fix \( a \)
  assume \( a \in \{ a \mid a \subseteq A \land \text{card } a = n \} \)
  hence \( \text{finite } (f a) \)
    using assms by blast
  thus \( \text{card } (A \setminus B a) = \text{card } (f a) \) choose \( g a \)
    by (fact n-subsets)

qed simp

finally have \( \text{card } (\Sigma A \cap B) = ?C \)

thus \( \theta \text{thesis} \)
  by (subst \( S \))

qed

lemma \( \text{prod-cancel-nat} \):
  — Contributed by Manuel Eberl

  fixes \( f :: 'a \Rightarrow \text{nat} \)

  assumes \( B \subseteq A \) and \( \text{finite } A \) and \( \forall x \in B. \ f x \neq 0 \)

  shows \( \prod A / \prod B = \prod (A \setminus B) \) (is \( ?A / ?B = ?C \))

proof –

  from \( \text{prod.subset-diff}[OF assms(1,2)] \) have \( ?A = ?C \ast ?B \) by auto

  moreover have \( ?B \neq 0 \) using assms by (simp add: finite-subset)

  ultimately show \( \theta \text{thesis} \) by simp

qed

lemma \( \text{prod-id-cancel-nat} \):
  — Contributed by Manuel Eberl

  fixes \( A :: \text{nat set} \)

  assumes \( B \subseteq A \) and \( \text{finite } A \) and \( 0 \notin B \)

  shows \( \prod A / \prod B = \prod (A \setminus B) \)

  using assms(1-2) by (rule prod-cancel-nat) (metis assms)

lemma \( \text{(in prob-space)} \text{ integrable-squareD} \):
  — Contributed by Johannes HlzI

  fixes \( X :: - \Rightarrow \text{real} \)
assumes integrable M (\(\lambda x. (X x)^2\)) \(X \in\) borel-measurable M
shows integrable M X
proof –
  have integrable M (\(\lambda x. \max 1 ((X x)^2)\))
    using assms by auto
then show integrable M X
  proof (rule Bochner-Integration.integrable-bound[OF - - always-eventually[OF allI]])
    fix x show norm (X x) \(\leq\) norm (\(\max 1 ((X x)^2)\))
      using abs-le-square-iff[of 1 X x] power-increasing[of 1 2 abs (X x)]
      by (auto split: split-max)
  qed fact
qed

end
theory Prob-Lemmas
imports
  HOL-Probability.Probability
  Girth-Chromatic.Girth-Chromatic
  Ugraph-Misc
begin

3 Lemmas about probabilities

In this section, auxiliary lemmas for computing bounds on expectation and probabilities of random variables are set up.

3.1 Indicator variables and valid probability values

abbreviation rind :: 'a set \(\Rightarrow\) 'a \(\Rightarrow\) real where
  rind \(\equiv\) indicator

lemma product-indicator:
  rind A x \(\ast\) rind B x = rind (A \(\cap\) B) x
unfolding indicator-def
by auto

We call a real number ‘valid’ iff it is in the range 0 to 1, inclusively, and additionally ‘nonzero’ iff it is neither 0 nor 1.

abbreviation valid-prob (p :: real) \(\equiv\) 0 \(\leq\) p \(\land\) p \(\leq\) 1
abbreviation nonzero-prob (p :: real) \(\equiv\) 0 \(<\) p \(\land\) p \(<\) 1

A function \(\forall a \Rightarrow\) real is a ‘valid probability function’ iff each value in the image is valid, and similarly for ‘nonzero’.

abbreviation valid-prob-fun f \(\equiv\) (\(\forall n.\) valid-prob (f n))
abbreviation nonzero-prob-fun f \(\equiv\) (\(\forall n.\) nonzero-prob (f n))
lemma nonzero-fun-is-valid-fun: nonzero-prob-fun f ⇒ valid-prob-fun f
by (simp add: less-imp-le)

3.2 Expectation and variance
context prob-space
begin

Note that there is already a notion of independent sets (see indep-set), but we use the following – simpler – definition:
definition indep A B ←→ prob (A ∩ B) = prob A * prob B

The probability of an indicator variable is equal to its expectation:
lemma expectation-indicator:
A ∈ events ⇒ expectation (rind A) = prob A
by simp

For a non-negative random variable X, the Markov inequality gives the following upper bound:
Pr[X ≥ a] ≤ E[X] / a

lemma markov-inequality:
assumes ⋀ a. 0 ≤ X a and integrable M X 0 < t
shows prob {a ∈ space M. t ≤ X a} ≤ expectation X / t
proof —
— proof adapted from edge-space.Markov-inequality, but generalized to arbitrary prob-spaces
have (∫ x. ennreal (X x) ∂M) = (∫ x. X x ∂M)
  using assms by (intro nn-integral-eq-integral) auto
thus ?thesis
  using assms nn-integral-Markov-inequality[of X M space M 1 / t]
  by (auto cong: nn-integral-cong simp: emeasure-eq-measure ennreal-mult[symmetric])
qed

Var[X] = E[X^2] − E[X]^2

lemma variance-expectation:
fixes X :: 'a ⇒ real
assumes integrable M (λx. (X x) ^2) and X ∈ borel-measurable M
shows integrable M (λx. (X x − expectation X)^2) (is ?integrable)
  variance X = expectation (λx. (X x)^2) − (expectation X)^2 (is ?variance)
proof —
have int: integrable M X
  using integrable-squareD[OF assms] by simp

have (λx. (X x − expectation X)^2) = (λx. (X x)^2 + (expectation X)^2 − (2 * X x * expectation X))
  by (simp only: power2-diff)
hence
\[
\text{variance } X = \text{expectation } (\lambda x. (X x)^2) + (\text{expectation } X)^2 + \text{expectation } \left( \lambda x. - (2 * X x * \text{expectation } X) \right)
\]
\text{using} \text{ integral-add by (simp add: int assms prob-space)} +

thus \text{variance} \text{ integrable}
\text{ by (simp add: int power2-eq-square)} +
\text{qed}

A corollary from the Markov inequality is Chebyshev’s inequality, which gives an upper bound for the deviation of a random variable from its expectation:
\[
\Pr[|Y - E[Y]| \geq s] \leq \frac{\text{Var}[X]}{s^2}
\]

\text{lemma chebyshev-inequality:}
\text{ fixes } Y :: 'a \Rightarrow real
\text{ assumes } Y-int: \text{ integrable } M (\lambda y. (Y y)^2)
\text{ assumes } Y-borel: Y \in \text{ borel-measurable } M
\text{ fixes } s :: real
\text{ assumes } s-pos: 0 < s
\text{ shows } \text{prob } \{a \in \text{space } M. s \leq |Y a - \text{expectation } Y|\} \leq \text{variance } Y / s^2
\text{proof –}
\text{ let } \exists X = \lambda a. (Y a - \text{expectation } Y)^2
\text{ let } \exists t = s^2

\text{ have } 0 < \exists t
\text{ using } s-pos \text{ by simp}
\text{ hence } \text{prob } \{a \in \text{space } M. \exists t \leq \exists X a\} \leq \text{variance } Y / s^2
\text{ using } \text{markov-inequality variance-expectation}[OF \ Y-int \ Y-borel] \text{ by (simp add: field-simps) }
\text{ moreover have } \{a \in \text{space } M. \exists t \leq \exists X a\} = \{a \in \text{space } M. s \leq |Y a - \text{expectation } Y|\}
\text{ using } \text{abs-le-square-iff } s-pos \text{ by force }
\text{ ultimately show } \exists \text{thesis}
\text{ by simp}
\text{qed}

Hence, we can derive an upper bound for the probability that a random variable is 0.

\text{corollary chebyshev-prob-zero:}
\text{ fixes } Y :: 'a \Rightarrow real
\text{ assumes } Y-int: \text{ integrable } M (\lambda y. (Y y)^2)
\text{ assumes } Y-borel: Y \in \text{ borel-measurable } M
\text{ assumes } \mu-pos: \text{expectation } Y > 0
\text{ shows } \text{prob } \{a \in \text{space } M. Y a = 0\} \leq \text{expectation } (\lambda y. (Y y)^2) / (\text{expectation } Y)^2 - 1
\text{proof –}
let \( ?s = \text{expectation } Y \)

have \( \text{prob } \{ a \in \text{space } M. \ Y a = 0 \} \leq \text{prob } \{ a \in \text{space } M. \ ?s \leq \lvert Y a - ?s \rvert \} \)

using \( Y \)-borel by (auto intro!: finite-measure-mono borel-measurable-diff borel-measurable-abs borel-measurable-le)

also have \( \ldots \leq \text{variance } Y / ?s^2 \)

using \( \text{assms by } (\text{fact chebyshev-inequality}) \)

also have \( \ldots = (\text{expectation } (\lambda y. (Y y)^2) - ?s^2) / ?s^2 \)

using \( Y \)-int \( Y \)-borel by (simp add: variance-expectation)

also have \( \ldots = \text{expectation } (\lambda y. (Y y)^2) / ?s^2 - 1 \)

using \( \mu \)-pos by (simp add: field-simps)

finally show \( ?\text{thesis} \).

qed

end

3.3 Sets of indicator variables

This section introduces some inequalities about expectation and other values related to the sum of a set of random indicators.

locale prob-space-with-indicators = prob-space +

fixes \( I \) :: `'i set

assumes finite-I: finite \( I \)

fixes \( A \) :: `'i \Rightarrow `'a set

assumes \( A \) : \( A \cdot I \subseteq \text{events} \)

assumes \( \text{prob-non-zero: } \exists i \in I. \ 0 < \text{prob } (A i) \)

begin

We call the underlying sets \( A i \) for each \( i \in I \), and the corresponding indicator variables \( X i \). The sum is denoted by \( Y \), and its expectation by \( \mu \).

definition \( X i = \text{rind } (A i) \)
definition \( Y x = (\sum i \in I. \ X i x) \)
definition \( \mu = \text{expectation } Y \)

In the lecture notes, the following two relations are called \( \sim \) and \( \simeq \), respectively. Note that they are not the opposite of each other.

abbreviation \( \text{ineq-indep } :: \ 'i \Rightarrow \ 'i \Rightarrow \text{bool} \ where \text{ineq-indep } i \ j \equiv (i \neq j \land \text{indep } (A i) (A j)) \)

abbreviation \( \text{ineq-dep } :: \ 'i \Rightarrow \ 'i \Rightarrow \text{bool} \ where \text{ineq-dep } i \ j \equiv (i \neq j \land \neg \text{indep } (A i) (A j)) \)
definition \( \Delta_n = (\sum i \in I. \ \sum j \mid j \in I \land i \neq j. \ \text{prob } (A i \cap A j)) \)
definition \( \Delta_d = (\sum i \in I. \ \sum j \mid j \in I \land \text{ineq-dep } i \ j. \ \text{prob } (A i \cap A j)) \)
lemma $\Delta$-zero:
assumes $\forall i, j. i \in I \Rightarrow j \in I \Rightarrow i \neq j \Rightarrow \text{indep } (A_i)(A_j)$
shows $\Delta_d = 0$
proof –
\{ 
  fix $i$
  assume $i \in I$
  hence $\{j. j \in I \land \text{ineq-dep } i \ j\} = \{\}$
  using assms by auto
  hence $\left(\sum j \mid j \in I \land \text{ineq-dep } i \ j. \text{prob } (A_i \cap A_j)\right) = 0$
  using sum.empty by metis
\}
  hence $\Delta_d = (0 :: \text{real}) \ast \text{card } I$
  unfolding $\Delta_d$-def by simp
  thus $?\text{thesis}$
  by simp
qed

lemma $A$-events[measurable]: $i \in I \Rightarrow A_i \in \text{events}$
using $A$ by auto

lemma expectation-X-Y: $\mu = \left(\sum i \in I. \text{expectation } (X_i)\right)$
unfolding $\mu$-def $Y$-def[abs-def] $X$-def
by (simp add: less-top[symmetric])

lemma expectation-X-non-zero: $\exists i \in I. 0 < \text{expectation } (X_i)$
unfolding $X$-def using prob-non-zero expectation-indicator by simp

corollary $\mu$-non-zero[simp]: $0 < \mu$
unfolding expectation-X-Y
using expectation-X-non-zero
by (auto intro!: sum-lower finite-I
  simp add: expectation-indicator X-def)

lemma $\Delta_d$-nonneg: $0 \leq \Delta_d$
unfolding $\Delta_d$-def
by (simp add: sum-nonneg)

corollary $\mu$-sq-non-zero[simp]: $0 < \mu^2$
by (rule zero-less-power) simp

lemma Y-square-unfold: $(\lambda x. (Y x)^2) = (\lambda x. \sum i \in I. \sum j \in I. \text{rind } (A_i \cap A_j) x)$
unfolding fun-eq-iff Y-def X-def
by (auto simp: sum-square product-indicator)

lemma integrable-Y-sq[simp]: integrable $M (\lambda y. (Y y)^2)$
unfolding Y-square-unfold
lemma measurable-Y [measurable]: \( Y \in \) borel-measurable \( M \)

unfolding Y-definition X-definition by simp

lemma expectation-Y-\( \Delta \): expectation (\( \lambda x. (Y x) ^2 \)) = \( \mu + \Delta_a \)

proof -
  let \( \epsilon = \lambda i j. \) expectation (rind (\( A i \cap A j \)))

  have expectation (\( \lambda x. (Y x) ^2 \)) = (\( \sum i \in I. \sum j \in I. \epsilon(i j) \))
  unfolding Y-square-unfold by (simp add: less-top[symmetric])
  also have \( \ldots = (\sum i \in I. \sum j \in I. \text{if} \ i = j \text{ then } \epsilon(i j) \text{ else } \epsilon(i j) \)
  by simp
  also have \( \ldots = (\sum i \in I. (\sum j | j \in I \land i = j \ . \ \epsilon(i j)) + (\sum j | j \in I \land i \neq j \ . \ \epsilon(i j)) \)
  by simp only: sum-split[OF finite-I]
  also have \( \ldots = \mu + \Delta_a \)
  proof -
    have \( \text{lhs} = \mu \)
    proof -
      fix \( i \)
      assume \( i \in I \)
      have \( (\sum j | j \in I \land i = j \ . \ \epsilon(i j)) = (\sum j | j \in I \land i = j \ . \ \epsilon(i i)) \)
      by simp
      also have \( \ldots = (\sum j | i = j \ . \ \epsilon(i i)) \)
      using \( i \) by metis
      also have \( \ldots = \text{expectation (rind (A i))} \)
      by auto
      finally have \( (\sum j | j \in I \land i = j \ . \ \epsilon(i j)) = \ldots . \)
    hence \( \text{lhs} = (\sum i \in I. \text{expectation (rind (A i)))} \)
    by force
    also have \( \ldots = \mu \)
    unfolding expectation-X-Y X-definition ..
    finally show \( \text{lhs} = \mu . \)
  qed

moreover have \( \text{rhs} = \Delta_a \)

proof -
  fix \( i j \)
  assume \( i \in I \ j \in I \)
  with \( A \) have \( A i \cap A j \in \) events by blast
  hence \( \epsilon(i j) = \text{prob (A i} \cap \text{A j}) \)
  by (fact expectation-indicator)

qed
thus \(?thesis\)
unfolding \(\Delta_a\)-def by simp
qed
ultimately show \(?lhs + ?rhs = \mu + \Delta_a\)
by simp
qed
finally show \(?thesis\).
qed

lemma \(\Delta\)-expectation-X: \(\Delta_a \leq \mu^2 + \Delta_d\)
proof
  let \(?p = \lambda i j. \text{prob} \ (A i \cap A j)\)
  let \(?p' = \lambda i j. \text{prob} \ (A i) \ast \text{prob} \ (A j)\)
  let \(?ie = \lambda i j. \text{indep} \ (A i) \ (A j)\)

  have \(\Delta_a = \left( \sum_{i \in I} \sum_{j \mid j \in I \land i \neq j \ if \ ?ie \ i \ j \ then \ ?p \ i \ j \ else \ ?p \ i \ j} \right)\)
unfolding \(\Delta_a\)-def by simp
also have \(\ldots = (\sum_{i \in I} \sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p \ i \ j}) + (\sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p \ i \ j) + \Delta_d \ (\text{is - =} \ ?lhs + -)\)
unfolding \(\Delta_d\)-def by (fact sum.distrib)
also have \(\ldots \leq \mu^2 + \Delta_d\)
proof (rule add-right-mono)
  have \(\left( \sum_{i \in I} \sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p' \ i \ j} \right) = (\sum_{i \in I} \sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p' \ i \ j) + \Delta_d \ (\text{is - =} \ ?lhs + -)\)
unfolding indep-def by simp
also have \(\ldots \leq (\sum_{i \in I} \sum_{j \mid j \in I. \ ?p' \ i \ j})\)
proof (rule sum-mono)
  fix \(i\)
  assume \(i \in I\)
  show \(\left( \sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p' \ i \ j} \right) \leq (\sum_{j \mid j \in I \land \text{ineq-indep} \ i \ j. \ ?p' \ i \ j)\)
  by (rule sum-upper[OF finite-I]) (simp add: zero-le-mult-iff)
qed
also have \(\ldots = (\sum_{i \in I. \ \text{prob} \ (A i)) \ ^{\ast 2}\)
  by (fact sum-square[ symmetric])
also have \(\ldots = (\sum_{i \in I. \ \text{expectation} \ (X i)) \ ^{\ast 2}\)
  unfolding X-def using expectation-indicator A by simp
also have \(\ldots = \mu^2\)
  using expectation-X-Y[ symmetric] by simp
finally show \(?lhs \leq \mu^2\).
qed
finally show ?thesis.
qed

lemma prob-\(\mu\)-\(\Delta_a\): prob \(\{ a \in \text{space} \ M. \ Y a = 0 \} \leq 1 - \mu + \Delta_a / \mu^2 - 1\)
proof
  have prob \(\{ a \in \text{space} \ M. \ Y a = 0 \} \leq \text{expectation} \ (\lambda y. \ (Y y) \ ^{\ast 2}) / \mu^2 - 1\)
unfolding \( \mu \text{-def} \) by (rule chebyshev-prob-zero) (simp add: \( \mu \text{-def}[\text{symmetric}] \))+
also have \( \ldots = (\mu + \Delta_a) / \mu^2 - 1 \)
using expectation-Y-\( \Delta \) by simp
also have \( \ldots = 1 / \mu + \Delta_a / \mu^2 - 1 \)
unfolding power2-eq-square by (simp add: field-simps add-divide-distrib)
finally show \(?thesis\).

lemma prob-\( \mu \)-\( \Delta \)-d: \( \text{prob} \{ a \in \text{space } M. Y \ a = 0 \} \leq 1 / \mu + \Delta_d / \mu^2 \)
proof
  have \( \text{prob} \{ a \in \text{space } M. Y \ a = 0 \} \leq 1 / \mu + \Delta_d / \mu^2 - 1 \)
    by (fact prob-\( \mu \)-\( \Delta \))
  also have \( \ldots = (1 / \mu - 1) + \Delta_d / \mu^2 \)
    by simp
  also have \( \ldots \leq (1 / \mu - 1) + (\mu^2 + \Delta_d) / \mu^2 \)
    using divide-right-mono[OF \( \Delta \text{-expectation-X} \)] by simp
  also have \( \ldots = 1 / \mu + \Delta_d / \mu^2 \)
    using \( \mu \text{-sq-non-zero} \) by (simp add: field-simps)
finally show \(?thesis\).

qed

end

end

4 Lemmas about undirected graphs

theory Ugraph-Lemmas
imports
  Prob-Lemmas
  Girth-Chromatic.Girth-Chromatic
begin

The complete graph is a graph where all possible edges are present. It is
wellformed by definition.

definition complete :: nat set \( \Rightarrow \) ugraph where
complete V = \((V, \text{all-edges } V)\)

lemma complete-wellformed: uwellformed (complete V)
unfolding complete-def uwellformed-def all-edges-def
by simp

If the set of vertices is finite, the set of edges in the complete graph is finite.

lemma all-edges-finite: finite V \( \Rightarrow \) finite (all-edges V)
unfolding all-edges-def
by simp

corollary complete-finite-edges: finite V \( \Rightarrow \) finite (uedges (complete V))
The sets of possible edges of disjoint sets of vertices are disjoint.

**Lemma** all-edges-disjoint: \( S \cap T = \{\} \implies \text{all-edges } S \cap \text{all-edges } T = \{\} \\

**Proof** 

A graph is called ‘finite’ if its set of edges and its set of vertices are finite.

**Definition** finite-graph \( G \equiv \text{finite } (uverts ) G \land \text{finite } (uedges ) G \)

The complete graph is finite.

**Corollary** complete-finite: \( \text{finite } V \implies \text{finite-graph } (\text{complete } V) \\

**Proof** 

A graph is called ‘nonempty’ if it contains at least one vertex and at least one edge.

**Definition** nonempty-graph \( G \equiv \text{uverts } G \neq \{\} \land \text{uedges } G \neq \{\} \)

A random graph is both wellformed and finite.

**Lemma** (in edge-space) wellformed-and-finite: 

**Proof** 

The probability for a random graph to have \( e \) edges is \( p^e \).

**Lemma** (in edge-space) cylinder-empty-prob: 

**Proof**

4.1 Subgraphs

**Definition** subgraph :: ugraph \( \Rightarrow \) ugraph \( \Rightarrow \) bool where

**Lemma** subgraph-refl: subgraph \( G G \)
lemma subgraph-trans: subgraph \( G'' G' \) \( \Rightarrow \) subgraph \( G' G \) \( \Rightarrow \) subgraph \( G'' G \)

unfolding subgraph-def
by auto

lemma subgraph-antisym: subgraph \( G G' \) \( \Rightarrow \) subgraph \( G' G \) \( \Rightarrow \) \( G = G' \)

unfolding subgraph-def
by (auto simp add: Product-Type.prod-eqI)

lemma subgraph-complete:
assumes \([\text{wellformed } G]\)
shows \( \text{subgraph } G (\text{complete } (\text{uverts } G)) \)

proof -
{
  fix \( e \)
  assume \( e \in \text{uedges } G \)
  with \text{assms}
  have \( \text{card } e = 2 \) \text{ and } \( \forall u. u \in e \Rightarrow u \in \text{uverts } G \)
  unfolding \text{wellformed-def} \by auto
  moreover then obtain \( u v \) \text{ where } \( e = \{ u, v \} \ u \neq v \)
  by (metis \text{card-2-elements})
  ultimately have \( e = \text{mk-uedge } (u, v) u \in \text{uverts } G v \in \text{uverts } G \)
  by auto
  hence \( e \in \text{all-edges } (\text{uverts } G) \)
  unfolding \text{all-edges-def} \text{ using } \langle u \neq v \rangle \text{ by fastforce}
  }
thus \( ?\text{thesis} \)
unfolding \text{complete-def subgraph-def} \by auto
qed

corollary wellformed-all-edges: \( \text{wellformed } G \Rightarrow \text{uedges } G \subseteq \text{all-edges } (\text{uverts } G) \)
using subgraph-complete subgraph-def complete-def \by simp

lemma subgraph-finite: \[ \text{finite-graph } G; \text{subgraph } G' G \] \( \Rightarrow \) \text{finite-graph } G'
unfolding finite-graph-def subgraph-def
by (metis \text{rev-finite-subset})

corollary wellformed-finite:
assumes \( \text{finite } (\text{uverts } G) \) \text{ and } \text{wellformed } G
shows \text{finite-graph } G
proof (rule subgraph-finite\[\text{where } G = \text{complete } (\text{uverts } G)\])
  show \text{subgraph } G (\text{complete } (\text{uverts } G))
  using \text{assms} \by (simp add: subgraph-complete)
  next
  have \( \text{finite } (\text{uedges } (\text{complete } (\text{uverts } G))) \)
  using \text{complete-finite-edges}[OF \text{assms}(1)]
  thus \( \text{finite-graph } (\text{complete } (\text{uverts } G)) \)
unfolding finite-graph-def complete-def using assms(1) by auto

qed

definition subgraphs :: ugraph ⇒ ugraph set where
subgraphs G = {G': subgraph G' G}

definition nonempty-subgraphs :: ugraph ⇒ ugraph set where
nonempty-subgraphs G = {G': uwellformed G' ∧ subgraph G' G ∧ nonempty-graph G'}

lemma subgraphs-finite:
  assumes finite-graph G
  shows finite (subgraphs G)
proof
  have subgraphs G = {{V', E'}). V' ⊆ uverts G ∧ E' ⊆ uedges G}
    unfolding subgraphs-def subgraph-def by force
  moreover have finite (uverts G) finite (uedges G)
    using assms unfolding finite-graph-def by auto
  ultimately show ?thesis
    by simp
qed

corollary nonempty-subgraphs-finite: finite-graph G ⇒ finite (nonempty-subgraphs G)
using subgraphs-finite
unfolding nonempty-subgraphs-def subgraphs-def
by auto

4.2 Induced subgraphs

definition induced-subgraph :: uvert set ⇒ ugraph ⇒ ugraph where
induced-subgraph V G = (V, uedges G ∩ all-edges V)

lemma induced-is-subgraph:
  V ⊆ uverts G ⇒ subgraph (induced-subgraph V G) G
  V ⊆ uverts G ⇒ subgraph (induced-subgraph V G) (complete V)
unfolding subgraph-def induced-subgraph-def complete-def
by simp

lemma induced-wellformed: uwellformed G ⇒ V ⊆ uverts G ⇒ uwellformed (induced-subgraph V G)
unfolding uwellformed-def induced-subgraph-def all-edges-def
by force

lemma subgraph-union-induced:
  assumes uverts H_1 ⊆ S and uverts H_2 ⊆ T
  assumes uwellformed H_1 and uwellformed H_2
  shows subgraph H_1 (induced-subgraph S G) ∧ subgraph H_2 (induced-subgraph T G)
4.3 Graph isomorphism

We define graph isomorphism slightly different than in the literature. The usual definition is that two graphs are isomorphic iff there exists a bijection between the vertex sets which preserves the adjacency. However, this complicates many proofs.

Instead, we define the intuitive mapping operation on graphs. An isomorphism between two graphs arises if there is a suitable mapping function from the first to the second graph. Later, we show that this operation can be inverted.

fun $\text{map-ugraph} :: (\text{nat} \Rightarrow \text{nat}) \Rightarrow \text{ugraph} \Rightarrow \text{ugraph}$ where
\[
\text{map-ugraph}
\quad f
\quad (V, E)
\quad =
\quad (f \cdot V, (\lambda e. f \cdot e) \cdot E)
\]

definition $\text{isomorphism} :: \text{ugraph} \Rightarrow \text{ugraph} \Rightarrow (\text{nat} \Rightarrow \text{nat}) \Rightarrow \text{bool}$ where
isomorphism \( G_1, G_2 \) \( f \equiv \text{bij-betw} \ (\text{uverts} \ G_1) \ (\text{uverts} \ G_2) \land G_2 = \text{map-ugraph} \ f \ G_1 \)

**abbreviation** isomorphic :: ugraph \( \Rightarrow \) ugraph \( \Rightarrow \) bool \((\sim)\) where
\( G_1 \sim G_2 \equiv \text{uwelformed} \ G_1 \land \text{uwelformed} \ G_2 \land (\exists \ f. \ \text{isomorphism} \ G_1 \ G_2 \ f) \)

**lemma** map-ugraph-id: map-ugraph id = id
**unfolding** fun-eq-iff
**by** simp

**lemma** map-ugraph-trans: map-ugraph \((g \circ f)\) = \((\text{map-ugraph} \ g) \circ (\text{map-ugraph} \ f)\)
**by** (simp add: fun-eq-iff image-image)

**lemma** map-ugraph-wellformed:
- assumes \( \text{uwelformed} \ G \) and \( \text{inj-on} \ f \ (\text{uverts} \ G) \)
- shows \( \text{uwelformed} \ (\text{map-ugraph} \ f \ G) \)
**unfolding** uwelformed-def
**proof** safe
  fix \( e' \)
  assume \( e' \in \text{uedges} \ (\text{map-ugraph} \ f \ G) \)
  hence \( e' \in (\lambda e. \ f ' e) \ ' (\text{uedges} \ G) \)
  by (metis map-ugraph.simps snd-conv surjective-pairing)
  then obtain \( e \) where \( e: e' = f ' e \ e \in \text{uedges} \ G \)
  by blast
  hence \( \text{card} \ e = 2 \ e \subseteq \text{uverts} \ G \)
  using assms(1) **unfolding** uwelformed-def **by** blast+
  thus \( \text{card} \ e' = 2 \)
  using \( e(1) \) by (simp add: card-inj-subs[OF assms(2)])

  fix \( u' \)
  assume \( u' \in e' \)
  hence \( u' \in f ' e \)
  using \( e \) by force
  then obtain \( u \) where \( u: u' = f ' u \ u \in e \)
  by blast
  hence \( u \in \text{uverts} \ G \)
  using assms(1) \( e(2) \) **unfolding** uwelformed-def **by** blast
  hence \( u' \in f ' \text{uverts} \ G \)
  using \( u(1) \) by simp
  thus \( u' \in \text{uverts} \ (\text{map-ugraph} \ f \ G) \)
  by (metis map-ugraph.simps fst-conv surjective-pairing)
**qed**

**lemma** map-ugraph-finite: \( \text{finite-graph} \ G \Longrightarrow \text{finite-graph} \ (\text{map-ugraph} \ f \ G) \)
**unfolding** finite-graph-def
**by** (metis finite-imageI fst-conv map-ugraph.simps snd-conv surjective-pairing)

**lemma** map-ugraph-preserves-sub:
assumes subgraph $G_1$, $G_2$
shows subgraph (map-ugraph $f$ $G_1$) (map-ugraph $f$ $G_2$)
proof –
have $f \cdot \text{uverts} \ G_1 \subseteq f \cdot \text{uverts} \ (\lambda e. \ f \cdot e) \cdot \text{uedges} \ G_1 \subseteq (\lambda e. \ f \cdot e) \cdot \text{uedges} \ G_2$
  using assms(1) unfolding subgraph-def by auto
thus ?thesis
unfolding subgraph-def by (metis map-ugraph.simps fst-conv snd-conv surjective-pairing)
qed

lemma isomorphic-refl: \text{uwellformed} \ G \implies \ G \simeq \ G
unfolding isomorphism-def
by (metis bij-betw-id id-def map-ugraph-id)

lemma isomorphic-trans:
assumes $G_1 \simeq G_2$ and $G_2 \simeq G_3$
shows $G_1 \simeq G_3$
proof –
from assms obtain $f_1$, $f_2$ where
  bij: \text{bij-betw} \ f_1 \ (\text{uverts} \ G_1) \ (\text{uverts} \ G_2) \ \text{bij-betw} \ f_2 \ (\text{uverts} \ G_2) \ (\text{uverts} \ G_3) \text{ and}
  map: \ G_2 = \text{map-ugraph} \ f_1 \ G_1 \ G_3 = \text{map-ugraph} \ f_2 \ G_2
unfolding isomorphism-def by blast
let $\ ?f = f_2 \circ f_1$
have \text{bij-betw} \ ?f \ (\text{uverts} \ G_1) \ (\text{uverts} \ G_3)
  using bij by (simp add: bij-betw-comp-iff)
moreover have $G_3 = \text{map-ugraph} \ ?f \ G_1$
  using map by (simp add: map-ugraph-trans)
moreover have \text{uwellformed} \ G_1 \ \text{uwellformed} \ G_3
  using assms unfolding isomorphism-def by simp+
ultimately show $G_1 \simeq G_3$
unfolding isomorphism-def by blast
qed

lemma isomorphic-sym:
assumes $G_1 \simeq G_2$
shows $G_2 \simeq G_1$
proof safe
from assms obtain $f$ where isomorphism $G_1$ $G_2$ $f$
  by blast
hence bij: \text{bij-betw} \ f \ (\text{uverts} \ G_1) \ (\text{uverts} \ G_2) \text{ and}
  map: \ G_2 = \text{map-ugraph} \ f \ G_1
unfolding isomorphism-def by auto
let $\ ?f' = \text{inv-into} \ (\text{uverts} \ G_1) \ f$
have \text{bij'}: \text{bij-betw} \ ?f' \ (\text{uverts} \ G_2) \ (\text{uverts} \ G_1)
  by (rule bij-betw-inv-into) fact
moreover have $\text{uverts} \ G_1 = ?f' \cdot \text{uverts} \ G_2$
  using bij' unfolding bij-betw-def by force
moreover have $\text{uedges} \ G_1 = (\lambda e. \ ?f' \cdot e) \cdot \text{uedges} \ G_2$

proof –
have uedges $G_1 = \text{id} \cdot \text{uedges } G_1$
by simp
also have \( \ldots = (\lambda e. \not\exists f'. (f' e)) \cdot \text{uedges } G_1 \)
proof (rule image-cong)
fix $a$
assume $a \in \text{uedges } G_1$
hence $a \subseteq \text{uverts } G_1$
using assms unfolding isomorphism-def uwellformed-def by blast
thus $\text{id } a = \text{inv-into } (\text{uverts } G_1) f' f' a$
by (metis (full-types) id-def bij bij-betw-imp-inj-on inv-into-image-cancel)
qed simp
also have \( \ldots = (\lambda e. \not\exists f' e) \cdot ((\lambda e. f' e) \cdot \text{uedges } G_1) \)
by (rule image-image [symmetric])
also have \( \ldots = (\lambda e. \not\exists f' e) \cdot \text{uedges } G_2 \)
using bij map by (metis map-ugraph.simps prod.collapse snd-eqD)
finally show \( ?\text{thesis} \)
qed

ultimately have isomorphism $G_2 \cong G_1$
unfolding isomorphism-def by (metis map-ugraph assms split-pairs)
thus $\exists f. \text{isomorphism } G_2 \cong G_1 f$
by blast
qed (auto simp: assms)

lemma isomorphic-cards:
assumes $G_1 \cong G_2$
shows $\text{card } (\text{uverts } G_1) = \text{card } (\text{uverts } G_2)$ (is $?V$
$\text{card } (\text{uedges } G_1) = \text{card } (\text{uedges } G_2)$ (is $?E$
proof –
from assms obtain $f$ where
bij: bij-betw $f$ (uverts $G_1$) (uverts $G_2$) and
map: $G_2 = \text{map-ugraph } f G_1$
unfolding isomorphism-def by blast
from assms have wellformed: uwellformed $G_1$ uwellformed $G_2$
by simp+

show $?V$
by (rule bij-betw-same-card[OF bij])

let $?g = \lambda e. f' e$
have bij-betw $?g$ (Pow (uverts $G_1$)) (Pow (uverts $G_2$))
by (rule bij-lift[OF bij])
moreover have uedges $G_1 \subseteq \text{Pow } (\text{uverts } G_1)$
using wellformed(1) unfolding uwellformed-def by blast
ultimately have card $(?g \cdot \text{uedges } G_1) = \text{card } (\text{uedges } G_1)$
unfolding bij-betw-def by (metis card-inj-subs)
thus $?E$
4.4 Isomorphic subgraphs

The somewhat sloppy term ‘isomorphic subgraph’ denotes a subgraph which is isomorphic to a fixed other graph. For example, saying that a graph contains a triangle usually means that it contains any triangle, not the specific triangle with the nodes 1, 2 and 3. Hence, such a graph would have a triangle as an isomorphic subgraph.

**definition** subgraph-isomorphic :: ugraph ⇒ ugraph ⇒ bool (⊑ -) where
G' ⊑ G ≡ uwellformed G ∧ (∃ G''. G' ≃ G'' ∧ subgraph G'' G)

**lemma** subgraph-is-subgraph-isomorphic: [ ] uwellformed G; uwellformed G; subgraph G' G ] ⇒ G' ⊑ G
unfolding subgraph-isomorphic-def
by (metis isomorphic-refl)

**lemma** isomorphic-is-subgraph-isomorphic: G₁ ≃ G₂ ⇒ G₁ ⊑ G₂
unfolding subgraph-isomorphic-def
by (metis subgraph-refl)

**lemma** subgraph-isomorphic-refl: uwellformed G ⇒ G ⊑ G
unfolding subgraph-isomorphic-def
by (metis isomorphic-refl subgraph-refl)

**lemma** subgraph-isomorphic-pre-iso-closed:
assumes G₁ ≃ G₂ and G₂ ⊑ G₃
shows G₁ ⊑ G₃
unfolding subgraph-isomorphic-def
proof
show uwellformed G₃
  using assms unfolding subgraph-isomorphic-def by blast
next
from assms(2) obtain G₂' where G₂ ≃ G₂' subgraph G₂' G₃
  unfolding subgraph-isomorphic-def by blast
moreover with assms(1) have G₁ ≃ G₂'
  by (metis isomorphic-trans)
ultimately show ∃ G''. G₁ ≃ G'' ∧ subgraph G'' G₃
  by blast
qed

**lemma** subgraph-isomorphic-pre-subgraph-closed:
assumes uwellformed G₁ and subgraph G₁ G₂ and G₂ ⊑ G₃
shows G₁ ⊑ G₃
unfolding subgraph-isomorphic-def
proof
show uwellformed G₃

26
using `assms unfoldsubgraph-isomorphic-def by blast`

next

from `assms(3) obtain G_2' where G_2 \cong G_2' subgraph G_2' G_3`

unfolding `subgraph-isomorphic-def by blast`

then obtain f where bij: bij-betw (uverts G_2) (uverts G_2') G_2' = map-ugraph f G_2

unfolding `isomorphism-def by blast`

let \(?G_1' = map-ugraph f G_1\)

have bij-betw f (uverts G_1) (f ' uverts G_1)

using bij(1) `assms(2) unfolding subgraph-def by (auto intro: bij-betw-subset)`

moreover hence `uwellformed \(?G_1'`

using `map-ugraph-wellformed[OF `assms(1)`] unfolding bij-betw-def ..`

ultimately have G_1 \cong \(?G_1'`

using `assms(1) unfolding isomorphism-def by (metis map-ugraph.simps fst-cone surjective-pairing)`

moreover have `subgraph \(?G_1' G_3`

using `map-ugraph-preserves-sub[OF `assms(2)`] bij(2) (subgraph G_2' G_3) by simp`

ultimately show \(\exists G''\). G_1 \cong G'' \land subgraph G'' G_3`

by blast

qed

lemmas `subgraph-isomorphic-pre-closed` = `subgraph-isomorphic-pre-subgraph-closed`

`subgraph-isomorphic-pre-iso-closed`

```
lemma `subgraph-isomorphic-trans[trans]`:
   assumes `G_1 \subseteq G_2` and `G_2 \subseteq G_3`
   shows `G_1 \subseteq G_3`

proof –
   from `assms(1) obtain G where G_1 \cong G subgraph G G_2`
   unfolding `subgraph-isomorphic-def by blast`
   thus \(?thesis`
   using `assms(2)` by (metis `subgraph-isomorphic-pre-closed`)

qed
```

```
lemma `subgraph-isomorphic-post-iso-closed` [H \subseteq G; G \cong G'] \Rightarrow H \subseteq G'

using `isomorphic-is-subgraph-isomorphic subgraph-isomorphic-trans`
by blast
```

```
```

```
```

```
4.5 Density
```

The density of a graph is the quotient of the number of edges and the number of vertices of a graph.

definition `density :: ugraph \Rightarrow real where`
density \( G = \text{card(\text{\textit{edges}} G)} / \text{card(\text{\textit{verts}} G)} \)

The maximum density of a graph is the density of its densest nonempty subgraph.

**definition** max-density :: ugraph \( \Rightarrow \) real where

\[ \text{max-density } G = \text{Lattices-Big.Max (density ' nonempty-subgraphs } G) \]

We prove some obvious results about the maximum density, such as that there is a subgraph which has the maximum density and that the (maximum) density is preserved by isomorphisms. The proofs are a bit complicated by the fact that most facts about \textit{Max} require non-emptiness of the target set, but we need that anyway to get a value out of it.

**lemma** subgraph-has-max-density:

- **assumes** finite-graph \( G \) and nonempty-graph \( G \) and \( uwellformed \) \( G \)
- **shows** \( \exists G'. \text{density } G' = \text{max-density } G \land \text{subgraph } G' \land \text{nonempty-graph } G' \land \text{finite-graph } G' \land \text{uwellformed } G' \)

**proof** –

- have \( G \in \text{nonempty-subgraphs } G \)
- unfolding nonempty-subgraphs-def using subgraph-refl assms by simp
- hence \( \text{density } G \in \text{density ' nonempty-subgraphs } G \)
- by simp
- hence \( (\text{density ' nonempty-subgraphs } G) \neq \{\} \)
- by fast
- hence \( \text{max-density } G \in (\text{density ' nonempty-subgraphs } G) \)
- unfolding max-density-def by (auto simp add: nonempty-subgraphs-finite[OF assms(1)] Max.closed)
- thus \( \text{?thesis} \)
- unfolding nonempty-subgraphs-def using subgraph-finite[OF assms(1)] by force

**qed**

**lemma** max-density-is-max:

- **assumes** finite-graph \( G \) and finite-graph \( G' \) and nonempty-graph \( G' \) and \( uwellformed \) \( G' \) and \( \text{subgraph } G' \)
- **shows** \( \text{density } G' \leq \text{max-density } G \)

**unfolding** max-density-def

**proof** (rule Max-ge)

- show \( \text{finite (density ' nonempty-subgraphs } G) \)
  - using assms(1) by (simp add: nonempty-subgraphs-finite)
- next
  - show \( \text{density } G' \in \text{density ' nonempty-subgraphs } G \)
  - unfolding nonempty-subgraphs-def using assms by blast

**qed**

**lemma** max-density-gr-zero:

- **assumes** finite-graph \( G \) and nonempty-graph \( G \) and \( uwellformed \) \( G \)
- **shows** \( 0 < \text{max-density } G \)

**proof** –
have $0 < \text{card (uverts } G) \leq \text{card (uedges } G)$

using asms unfolding finite-graph-def nonempty-graph-def by auto

hence $0 < \text{density } G$

unfolding density-def by simp

also have $\text{density } G \leq \text{max-density } G$

using asms by (simp add: max-density-is-max subgraph-refl)

finally show \( \text{thesis} \).

qed

lemma isomorphic-density:
assumes $G_1 \simeq G_2$
shows $\text{density } G_1 = \text{density } G_2$
unfolding density-def
using isomorphic-cards[OF asms]
by simp

lemma isomorphic-max-density:
assumes $G_1 \simeq G_2$ and nonempty-graph $G_1$ and nonempty-graph $G_2$ and finite-graph $G_1$ and finite-graph $G_2$
shows $\text{max-density } G_1 = \text{max-density } G_2$
proof

— The proof strategy is not completely straightforward. We first show that if two graphs are isomorphic, the maximum density of one graph is less or equal than the maximum density of the other graph. The reason is that this proof is quite long and the desired result directly follows from the symmetry of the isomorphism relation.\(^1\)

\{ 
fix $A \ B$
assume $A$: nonempty-graph $A$ finite-graph $A$
assume iso: $A \simeq B$

then obtain $f$ where $f: B = \text{map-ugraph } f \ A$ bij-betw $f$ (uverts $A$) (uverts $B$)
unfolding isomorphism-def by blast
have wellformed: wellformed $A$
using iso unfolding isomorphism-def by simp
— We observe that the set of densities of the subgraphs does not change if we map the subgraphs first.

have $\text{density } ' \text{nonempty-subgraphs } A = \text{density } ' \ (\text{map-ugraph } f ' \text{nonempty-subgraphs } A)$

proof (rule image-comp-cong)
fix $G$
assume $G \in \text{nonempty-subgraphs } A$
hence uverts $G \subseteq \text{uverts } A$ wellformed $G$
unfolding nonempty-subgraphs-def subgraph-def by simp+
hence inj-on $f$ (uverts $G$)

\(^1\)Some famous mathematician once said that if you prove that $a \leq b$ and $b \leq a$, you know that these numbers are equal, but not why. Since many proofs in this work are mostly opaque to me, I can live with that.
using \( f(2) \) unfolding bij-betw-def by (metis subset-inj-on)

hence \( G \cong \text{map-ugraph } f \ G \)

unfolding isomorphism-def bij-betw-def
by (metis map-ugraph.simps fst-conv surjective-pairing map-ugraph-wellformed
\( \text{uwellformed } G \))

thus density \( G = \text{density } (\text{map-ugraph } f \ G) \)
by (fact isomorphic-density)

qed

Additionally, we show that the operations \( \text{nonempty-subgraphs} \) and \( \text{map-ugraph} \) can be swapped without changing the densities. This is an obvious result, because \( \text{map-ugraph} \) does not change the structure of a graph. Still, the proof is a bit hairy, which is why we only show inclusion in one direction and use symmetry of isomorphism later.

also have \( \ldots \subseteq \text{density } ^{\cdot} \text{nonempty-subgraphs } (\text{map-ugraph } f \ A) \)

proof (rule image-mono, rule subsetI)
fix \( G'' \)
assume \( G'' \in \text{map-ugraph } f ^{\cdot} \text{nonempty-subgraphs } A \)
then obtain \( G' \) where \( G' = \text{map-ugraph } f \ G' \ G' \in \text{nonempty-subgraphs } A \)

by blast

hence \( G': \text{subgraph } G' \ A \text{nonempty-graph } G' \ \text{uwellformed } G' \)

unfolding nonempty-subgraphs-def by auto

hence inj-on \( f \) (uverts \( G' \))
using \( f \) unfolding bij-betw-def subgraph-def by (metis subset-inj-on)

hence \( \text{uwellformed } G'' \)
using \( \text{map-ugraph-wellformed } G' \ G'-\text{subst} \) by simp

moreover have \( \text{nonempty-graph } G'' \)
using \( G' \ G'-\text{subst unfolding nonempty-graph-def} \) by (metis map-ugraph.simps fst-cone snd-cone surjective-pairing empty-is-image)

moreover have \( \text{subgraph } G'' \ (\text{map-ugraph } f \ A) \)
using \( \text{map-ugraph-preserves-sub } G' \ G'-\text{subst} \) by simp

ultimately show \( G'' \in \text{nonempty-subgraphs } (\text{map-ugraph } f \ A) \)

unfolding nonempty-subgraphs-def by simp

qed

finally have \( \text{density } ^{\cdot} \text{nonempty-subgraphs } A \subseteq \text{density } ^{\cdot} \text{nonempty-subgraphs } (\text{map-ugraph } f \ A) \)

hence \( \text{max-density } A \leq \text{max-density } (\text{map-ugraph } f \ A) \)

unfolding max-density-def

proof (rule Max-mono)
have \( A \in \text{nonempty-subgraphs } A \)
using \( A \text{ iso unfolding nonempty-subgraphs-def} \) by (simp add: subgraph-refl)
thus \( \text{density } ^{\cdot} \text{nonempty-subgraphs } A \neq \{\} \)
by blast

next
have finite \( (\text{nonempty-subgraphs } (\text{map-ugraph } f \ A)) \)
by (rule nonempty-subgraphs-finite[OF map-ugraph-finite[OF A(2)]])
thus finite \( (\text{density } ^{\cdot} \text{nonempty-subgraphs } (\text{map-ugraph } f \ A)) \)
by blast
qed

hence \( \max\text{-density} A \leq \max\text{-density} B \)

by \((\text{subst } f)\)

}\note le = this

show \(?\text{thesis}\)

using \([\text{OF assms(2) assms(4) assms(1)}] \text{le}[\text{OF assms(3) assms(5) isomorphic-sym[OF assms(1)]}]

by \((\text{fact antisym})\)

qed

4.6 Fixed selectors

In the proof of the main theorem in the lecture notes, the concept of a “fixed copy” of a graph is fundamental.

Let \( H \) be a fixed graph. A ‘fixed selector’ is basically a function mapping a set with the same size as the vertex set of \( H \) to a new graph which is isomorphic to \( H \) and its vertex set is the same as the input set.\(^2\)

\text{definition is-fixed-selector } H f = (\forall V. \text{finite } V \land \text{card } (\text{uverts } H) = \text{card } V \longrightarrow H \simeq f V \land \text{uverts } (f V) = V)

Obviously, there may be many possible fixed selectors for a given graph. First, we show that there is always at least one. This is sufficient, because we can always obtain that one and use its properties without knowing exactly which one we chose.

\text{lemma ex-fixed-selector:}

\text{assumes uwellformed } H \text{ and finite-graph } H

\text{obtains } f \text{ where is-fixed-selector } H f

\text{proof}

— I guess this is the only place in the whole work where we make use of a nifty little HOL feature called SOME, which is basically Hilbert’s choice operator. The reason is that any bijection between the the vertex set of \( H \) and the input set gives rise to a fixed selector function. In the lecture notes, a specific bijection was defined, but this is shorter and more elegant.

\text{let } ?bij = \lambda V. \text{SOME } g \cdot \text{bij-betw } g \cdot (\text{uverts } H) \cdot V

\text{let } ?f = \lambda V. \text{map-ugraph } (?bij V) \cdot H

}\{\text{fix } V :: \text{uvert set}

\text{assume finite } V \text{ card } (\text{uverts } H) = \text{card } V

\text{moreover have finite } (\text{uverts } H)

\text{using assms unfolding finite-graph-def by simp}

\text{ultimately have bij-betw } (?bij V) \cdot (\text{uverts } H) \cdot V

\text{by (metis finite-same-card-bij someI-ex)}

\text{moreover hence } *: \text{uverts } (?f V) = V \land \text{uwellformed } (?f V)

\text{using map-ugraph-wellformed[OF assms(1)]}

\(^2\)We call such a selector \textit{fixed} because its result is deterministic.
by (metis bij-betw-def map-ugraph.simps fst-conv surjective-pairing)
ultimately have **: \( H \simeq f V \)
  unfolding isomorphism-def using assms(1) by auto
  note * **
}
thus is-fixed-selector H \( f \)
  unfolding is-fixed-selector-def by blast
qed

lemma fixed-selector-induced-subgraph:
assumes is-fixed-selector H \( f \) and card (uverts H) = card V and finite V
assumes sub: subgraph \( (f V) \) (induced-subgraph V G) and V: \( V \subseteq uverts G \)
and G: uwellformed G
shows H \( \subseteq G \)
proof –
  have post: \( H \simeq f V \) uverts \( (f V) = V \)
  using assms unfolding is-fixed-selector-def by auto

  have H \( \subseteq f V \)
  by (rule isomorphic-is-subgraph-isomorphic)
      (simp add: post)
  also have \( f V \subseteq induced-subgraph V G \)
  by (rule subgraph-is-subgraph-isomorphic)
      (auto simp: induced-wellformed[OF G V] post sub)
  also have \( \ldots \subseteq G \)
  by (rule subgraph-is-subgraph-isomorphic[OF induced-wellformed])
      (auto simp: induced-is-subgraph[OF V])
  finally show H \( \subseteq G \)
.
qed

5 Classes and properties of graphs

theory Ugraph-Properties
imports
  Ugraph-Lemmas
  Girth-Chromatic.Girth-Chromatic
begin

A “graph property” is a set of graphs which is closed under isomorphism.

type-synonym ugraph-class = ugraph set

definition ugraph-property :: ugraph-class \Rightarrow bool where
  ugraph-property C \equiv \forall G \in C. \forall G'. G \simeq G' \rightarrow G' \in C

abbreviation prob-in-class :: (nat \Rightarrow real) \Rightarrow ugraph-class \Rightarrow nat \Rightarrow real where
  prob-in-class p c n \equiv probGn p n (\lambda es. edge-space.edge-ugraph n es \in c)
From now on, we consider random graphs not with fixed edge probabilities but rather with a probability function depending on the number of vertices. Such a function is called a “threshold” for a graph property iff

- for asymptotically larger probability functions, the probability that a random graph is an element of that class tends to 1 (“1-statement”), and
- for asymptotically smaller probability functions, the probability that a random graph is an element of that class tends to 0 (“0-statement”).

\[
\text{definition } \text{is-threshold} :: \text{ugraph-class } \Rightarrow (\text{nat } \Rightarrow \text{real}) \Rightarrow \text{bool where}
\]
\[
is-threshold \ c \ t \equiv \text{ugraph-property } c \land (\forall p. \text{nonzero-prob-fun } p \rightarrow (p << t \rightarrow \text{prob-in-class } p \ c \rightarrow 0) \land (t << p \rightarrow \text{prob-in-class } p \ c \rightarrow 1))
\]

\[
\text{lemma } \text{is-thresholdI}[\text{intro}]:
\]
\[
\text{assumes } \text{ugraph-property } c
\]
\[
\text{assumes } \bigwedge p. [\text{nonzero-prob-fun } p; p << t] \Rightarrow \text{prob-in-class } p \ c \rightarrow 0
\]
\[
\text{assumes } \bigwedge p. [\text{nonzero-prob-fun } p; t << p] \Rightarrow \text{prob-in-class } p \ c \rightarrow 1
\]
\[
\text{shows } \text{is-threshold } c \ t
\]
\[
\text{using } \text{assms unfolding } \text{is-threshold-def by blast}
\]

6 The subgraph threshold theorem

\[
\text{theory } \text{Subgraph-Threshold}
\]
\[
\text{imports}
\]
\[
\text{Ugraph-Properties}
\]
\[
\text{begin}
\]
\[
\text{lemma } (\text{in edge-space}) \text{ measurable-pred[measurable]: Measurable.pred } P \ Q
\]
\[
\text{by } (\text{simp add: P-def sets-point-measure space-point-measure subset-eq})
\]

This section contains the main theorem. For a fixed nonempty graph \( H \), we consider the graph property of ‘containing an isomorphic subgraph of \( H \)’. This is obviously a valid property, since it is closed under isomorphism.

The corresponding threshold function is

\[
t(n) = n^{\frac{1}{\rho'(H)}},
\]

where \( \rho' \) denotes max-density.

\[
\text{definition } \text{subgraph-threshold} :: \text{ugraph } \Rightarrow \text{nat } \Rightarrow \text{real where}
\]
\[
\text{subgraph-threshold } H \ n = n \ \text{powr } (\frac{1}{\text{max-density } H})
\]

\[
\text{theorem}
\]
assumes nonempty: nonempty-graph \( H \) and finite: finite-graph \( H \) and well-formed: wellformed \( H \)
shows is-threshold \{ \( G, H \subseteq G \) \} (subgraph-threshold \( H \))
proof
  show ugraph-property \{ \( G, H \subseteq G \) \}
    unfolding ugraph-property-def using subgraph-isomorphic-closed by blast
next
  — To prove the 0-statement, we introduce the subgraph with the maximum density as \( H_0 \). Note that \( \rho(H_0) = \rho'(H) \).

  fix \( p :: \text{nat} \Rightarrow \text{real} \)

  obtain \( H_0 \) where \( H_0 \) : density \( H_0 = \text{max-density} \) subgraph \( H_0 \)
  H nonempty-graph
  H_0 finite-graph H_0 wellformed H_0
    using subgraph-has-max-density assms by blast
  hence \( \text{card: } 0 < \text{card (uverts } H_0) \) \( 0 < \text{card (uedges } H_0) \)
    unfolding nonempty-graph-def finite-graph-def by auto

  let \( ?v = \text{card (uverts } H_0) \)
  let \( ?e = \text{card (uedges } H_0) \)

  assume p-nz: nonzero-prob-fun \( p \)
  hence \( p \) : valid-prob-fun \( p \)
    by (fact nonzero-fun-is-valid-fun)

  — Firstly, we follow from the assumption that \( p \) is asymptotically less than the threshold function that the product

  \[
  p(n)^{|E(H_0)|} \cdot n^{|V(H_0)|}
  \]

  tends to 0.

  assume \( p \ll \) subgraph-threshold \( H \)
  moreover
  \{
  fix \( n \)
  have \( p \cdot n \cdot n \text{ powr} (-\frac{1}{\text{max-density } H}) = p \cdot n \cdot n \text{ powr} (1 \cdot \text{max-density } H) \)
    by (simp add: powr-minus-divide)
  also have \( \ldots = p \cdot n \cdot n \text{ powr} \left( \frac{1}{\text{density } H_0} \right) \)
    using \( H_0 \) by simp
  also have \( \ldots = p \cdot n \cdot n \text{ powr} (\frac{?v}{?e}) \)
    using \( \text{card} \) unfolding density-def by simp
  finally have \( p \cdot n \cdot n \text{ powr} \left( -\frac{1}{\text{max-density } H} \right) \) = \ldots
  \}

  ultimately have \( (\lambda n. p \cdot n \cdot n \text{ powr} (\frac{?v}{?e})) \longrightarrow 0 \)
    unfolding subgraph-threshold-def by simp
  moreover have \( \wedge n. 1 \leq n \implies 0 < p \cdot n \cdot n \text{ powr} (\frac{?v}{?e}) \)
ultimately have \((\lambda n. (p n \ast n \text{ powr } (\lambda v / \lambda e)) \text{ powr } ?e) \longrightarrow 0\)

using \(\text{card}(2) p\) by (force intro: tendsto-zero-powrI)

hence limit: \((\lambda n. p n \ast n \text{ powr } ?e) \longrightarrow 0\)

by \(\text{rule LIMSEQ-cong}[\text{OF - eventually-sequentiallyI where } c = 1]\)

(auto simp: p card p-nz powr-powr powr-mult)

\{ fix n \\
assume n: \(?v \leq n\) \\
interpret ES: \(\text{edge-space } n (p n)\) \\
by unfolding-locales (auto simp: p) \\
let \(?\text{graph-of } = ES.\text{edge-ugraph}\) \\
— After fixing an \(n\), we define a family of random variables \(X\) indexed by a set of vertices \(v\) and a set of edges \(e\). Each \(X\) is an indicator for the event that \((v,e)\) is isomorphic to \(H_0\) and a subgraph of a random graph. The sum of all these variables is denoted by \(Y\) and counts the total number of copies of \(H_0\) in a random graph.

let \(?X = \lambda H_0' . \text{rind } \{ es \in \text{space } ES.P. \text{subgraph } H_0' (\text{?graph-of es}) \land H_0 \simeq H_0' \}\) \\
let \(?I = \{(v, e), v \subseteq \{1..n\} \land \text{card } v = ?v \land e \subseteq \text{all-edges } v \land \text{card } e = ?e\}\) \\
let \(?Y = \lambda es. \sum H_0' \in ?I. ?X H_0' es\) \\
— Now we prove an upper bound for the probability that a random graph contains a copy of \(H_0\). Observe that in that case, \(Y\) takes a value greater or equal than 1.

have prob-in-class p \{ G. H \subseteq G \} n = probGn p n (\lambda es. H \subseteq ?\text{graph-of es}) \\
by simp \\
also have \(\ldots \leq \text{probGn p n (\lambda es. 1 \leq ?Y es)}\) \\
proof (rule ES.finite-measure-mono, safe) \\
fix es \\
assume es: es \in \text{space } (MGn p n) \\
assume H \subseteq ?\text{graph-of es} \\
hence H_0 \subseteq ?\text{graph-of es} — since H_0 is a subgraph of H \\
using H_0 by (fast intro: subgraph-isomorphic-pre-subgraph-closed) \\
then obtain H_0' where H_0' : \text{subgraph } H_0' (\text{?graph-of es}) H_0 \simeq H_0' \\
unfolding subgraph-isomorphic-def \\
by blast \\
show I \leq ?Y es \\
proof (rule sum-lower-or-eq) \\
— The only relevant step here is to provide the specific instance of \((v,e)\) such that \(X_{(v,e)}\) takes a value greater or equal than 1. This is trivial, as we already
obtained that one above (i.e. $H'_0$). The remainder of the proof is just bookkeeping.

show $1 \leq \mathcal{X} H'_0$ es — by definition of $X$
  using $H'_0$ es by simp
next
  have uverts $H'_0 \subseteq \{1..n\}$ uedges $H'_0 \subseteq es$
  using $H_0'(1)$ unfolding subgraph-def ES.edge-ugraph-def ES.S-verts-def
ES.S-edges-def by simp+
  moreover have card (uverts $H'_0$) = ?v card (uedges $H'_0$) = ?e
    by (simp add: isomorphic-cards[OF ($H_0 \simeq H'_0$)])+
  moreover have uedges $H'_0$ \subseteq all-edges (uverts $H'_0$)
    using $H'_0$ by (simp add: wellformed-all-edges)
  ultimately show $H'_0 \in \mathcal{I}$
    by auto
next
  have \mathcal{I} \subseteq subgraphs (complete \{1..n\})
  unfolding complete-def subgraphs-def subgraph-def using all-edges-mono
  by auto blast
  moreover have finite (subgraphs (complete \{1..n\}))
    by (simp add: complete-finite subgraphs-finite)
  ultimately show finite \mathcal{I}
    by (fact finite-subset)
qed simp
qed simp

— Applying Markov’s inequality leaves us with estimating the expectation of $Y$, which is the sum of the individual $X$.
also have \ldots \leq ES.expectation \mathcal{X} / 1
  by (rule prob-space.markov-inequality) (auto simp: ES.prob-space-P sum-nonneg)
also have \ldots = ES.expectation \mathcal{X}
  by simp
also have \ldots = (\sum H'_0 \in \mathcal{I}. ES.expectation \mathcal{X} H'_0)
  by (rule Bochner-Integration.integral-sum(1)) simp

— Each expectation is bound by $p(n)^{|E(H_0)|}$. For the proof, we ignore the fact that the corresponding graph has to be isomorphic to $H_0$, which only increases the probability and thus the expectation. This only leaves us to compute the probability that all edges are present, which is given by $edge-space.cylinder-prob$.
also have \ldots \leq (\sum H'_0 \in \mathcal{I}. p n \wedge ?e)
  proof (rule sum-mono)
    fix $H'_0$
    assume $H'_0$ \in \mathcal{I}.
    have ES.expectation (\mathcal{X} H'_0) = ES.prob \{ es \in space ES.P. subgraph $H'_0$
      (?graph-of es) \wedge $H_0 \simeq H'_0$\}
      by (rule ES.expectation-indicator) (auto simp: ES.sets-eq ES.space-eq)
    also have \ldots \leq ES.prob \{ es \in space ES.P. uedges $H'_0$ \subseteq es\}
      unfolding subgraph-def by (rule ES.finite-measure-mono) (auto simp: ES.sets-eq ES.space-eq)
    also have \ldots = ES.prob (cylinder ES.S-edges (uedges $H'_0$) \{\})
      unfolding cylinder-def ES.space-eq by simp
also have ... = \text{card}(\text{uedges } H_0')

proof (rule ES.cylinder-empty-prob)
    have uverts H_0' \subseteq \{1..n\} uedges H_0' \subseteq all-edges (uverts H_0')
    using H_0' by auto
    hence uedges H_0' \subseteq all-edges \{1..n\}
    using all-edges-mono by blast
    thus uedges H_0' \subseteq ES.S-edges
    unfolding ES.S-edges-def ES.S-verts-def by simp
    qed
also have ... = \text{card}\ ?I \ast p n ^ ?e
    using H_0' by fastforce
finally show ES.expectation (?X H_0') \leq ...
    qed

— Since we have a sum of constant summands, we can rewrite it as a product.
also have ... = \text{card}\ ?I \ast p n ^ ?e
    by (rule sum-constant)

— We have to count the number of possible pairs \((v,e)\). From the definition of
the index set, note that we first choose \(|V(H_0)|\) elements out of a set of \(n\) vertices
and then \(|E(H_0)|\) elements out of all possible edges over these vertices.
also have ... = \((\binom{n}{?v} \ast ((?v choose 2) choose ?e)) \ast p n ^ ?e
proof (rule arg-cong[where \(x = \text{card}\ ?I\)]
    have \text{card}\ ?I = \(\sum v | v \subseteq \{1..n\} \land \text{card} v = ?v\). card (all-edges v) choose ?e
    by (rule card-dep-pair-set[where \(A = \{1..n\}\) and \(n = ?v\) and \(f = all-edges\)])
    (auto simp: finite-subset all-edges-finite)
also have ... = \((\sum v | v \subseteq \{1..n\} \land \text{card} v = ?v\). (?v choose 2) choose ?e)
proof (rule sum.cong)
    fix \(v\)
    assume \(v \in \{v. v \subseteq \{1..n\} \land \text{card} v = ?v\}\)
    hence \(v \subseteq \{1..n\}\) card \(v = ?v\)
    by auto
    thus \text{card}(all-edges \(v\)) choose ?e = (?v choose 2) choose ?e
    by (simp add: card-all-edges finite-subset)
    qed rule
also have ... = \text{card}(\{\{v. v \subseteq \{1..n\} \land \text{card} v = ?v\}\} \ast ((?v choose 2)
choose ?e)
    by simp
also have ... = \((n choose ?v) \ast ((?v choose 2) choose ?e) \ast p n ^ ?e
    by (simp add: n-subsets)
finally show \text{card}\ ?I = ...
    qed
also have ... = \((n choose ?v) \ast ((?v choose 2) choose ?e) \ast p n ^ ?e
    by simp

37
— Here, we use \( n^k \) as an upper bound for \( \binom{n}{k} \).
also have \( \ldots \leq (n \cdot ?v) \ast (((?v \text{ choose } 2) \text{ choose } ?e) \ast p \cdot n \cdot ?e) \) (is - \( \leq \) - * ?v)

proof (rule multi-right-mono)
- have \( n \text{ choose } ?v \leq n \cdot ?v \)
  - by (rule binomial-le-pow) (rule n)
- thus \( \text{real} \ (n \text{ choose } ?v) \leq \text{real} \ (n \cdot ?v) \)
  - by (metis of-nat-le-iff)
next
show \( 0 \leq ?r \) using \( p \) by simp
qed
also have \( \ldots \leq (((?v \text{ choose } 2) \text{ choose } ?e) \ast \ (p \cdot n \cdot ?e \cdot n \cdot ?v) \) (is - \( \leq \) ?factor * -)
  - by simp
also have \( \ldots = ?\text{factor} \ast (p \cdot n \cdot p \cdot ?e \ast n \cdot p \cdot ?v) \)
  using \( n \cdot \text{card}(I) \) (nonzero-prob-fun \( p \)) by (simp add: powr-realpow)
finally have \( \text{prob-in-class} \ p \ \{G. \ H \subseteq G\} \ n \leq ?\text{factor} \ast (p \cdot n \cdot p \cdot ?e \ast n \cdot p \cdot ?v) \).

— The final upper bound is a multiple of the expression which we have proven to tend to 0 in the beginning.
thus \( \text{prob-in-class} \ p \ \{G. \ H \subseteq G\} \rightarrow 0 \)
by (rule LIMSEQ-le-zero [OF tendsto-mult-right-zero [OF limit eventually-sequentiallyI [OF measure-nonneg eventually-sequentiallyI]])
next
fix \( p :: \text{nat} \Rightarrow \text{real} \)
assume \( p\text{-threshold: subgraph-threshold} \ H \ll p \)

— To prove the 1-statement, we obtain a fixed selector \( f \) as defined in section 4.6.
from \( \text{assms} \) obtain \( f \) where \( f : \text{is-fixed-selector} \ H f \)
  using \( \text{ex-fixed-selector} \) by blast
let \( ?v = \text{card} \ (\text{averts} \ H) \)
let \( ?e = \text{card} \ (\text{uedges} \ H) \)

— We observe that several terms involving \( |V(H)| \) are positive.
have \( v\cdot\text{-nz:} \ 0 < \text{real} \ ?v 0 < \text{real} \ ?e \)
  using nonempty finite unfolding nonempty-graph-def finite-graph-def by auto
hence \( 0 < \text{real} \ ?v \cdot ?e \) by simp
hence \( \text{powr\cdot inv\cdot gr\cdot z:} \ 0 < 1 / ?v \cdot ?e \) by simp

— For a given \( n \), let \( A \) be a family of events indexed by a set \( S \). Each \( A \) contains the graphs whose induced subgraphs over \( S \) contain the selected copy of \( H \) by \( f \) over \( S \).
let \( ?A = \lambda n. \lambda S. \left\{ es \in \text{space} \ (\text{edge-space} \ P n \ (p n)) . \text{subgraph} \ (f S) \ (\text{induced-subgraph} \ S \ (\text{edge-space} \ . \text{edge-ugraph} \ n \ es) \right\} \)
let $\mathcal{I} = \lambda n. \{ S. S \subseteq \{1..n\} \land \text{card } S = \mathcal{v}\}$

assume $p$-nz: nonzero-prob-fun $p$

**hence** $p$: valid-prob-fun $p$

by (fact nonzero-fun-is-valid-fun)

{fix $n$
  — At this point, we can assume almost anything about $n$: We only have to show that a function converges, hence the necessary properties are allowed to be violated for small values of $n$.

assume $n$-2v: $2 * \mathcal{v} \leq n$

**hence** $n$: $\mathcal{v} \leq n$

by simp

**have** is-es: edge-space ($p$ $n$)

by unfold-locales (auto simp: $p$)

then interpret edge-space $n$ $p$ $n$

let $\mathcal{A} = \mathcal{A} n$

let $\mathcal{I} = \mathcal{I} n$

— A nice potpourri with some technical facts about $S$.

{fix $S$

assume $S \in \mathcal{I}$

**hence** 0: $S \subseteq \{1..n\} \mathcal{v} = \text{card } S \text{ finite } S$

by (auto intro: finite-subset)

**hence** 1: $H \simeq f S \text{ weerts } (f S) = S$

using $f$ wellformed-finite unfolding finite-graph-def is-fixed-selector-def by auto

have 2: finite-graph ($f S$)

using 0(3) 1(1,2) by (metis wellformed-finite)

have 3: nonempty-graph ($f S$)

using 0(2) 1(1,2) by (metis card-eq-0-iff finite finite-graph-def isomorphic-cards(2) nonempty nonempty-graph-def prod.collapse snd_conv)

note 0 1 2 3

} note $I = \text{this}$

— In the following two blocks, we prove the probabilities of the events $A$ and the probability of the intersection of two events $A$. For both cases, we employ the auxiliary lemma edge-space.induced-subgraph-prob which is not very interesting. For the latter however, the tricky part is to argue that such an intersection is equivalent to the union of the desired copies of $H$ to be contained in the union of the induced subgraphs.

{fix $S$
assume $S$: $S \in ?I$
note $S' = I[\text{OF } S]$
have $\text{prob } (?A S) = p^n \cdot ?e$
using isomorphic-cards($2)(\text{OF } S'(4)) \text{ S' by (simp add: S-verts-def induced-subgraph-prob)}$
}

note $\text{prob-A} = \text{this}$

{ fix $S \; T$
  assume $S \in ?I$ note $S = I[\text{OF this}]$
  assume $T \in ?I$ note $T = I[\text{OF this}]$
  — Note that we do not restrict $S$ and $T$ to be disjoint, since we need the
general case later to determine when two events are independent. Additionally, it
would be unneeded at this point.

  have $\text{prob } (?A S \cap ?A T) = \text{prob } \{ \text{es} \in \text{space } P. \text{subgraph } (S \cup T, \text{uedges } (f S) \cup \text{uedges } (f T)) \text{ (induced-subgraph } (S \cup T) \text{ (edge-ugraph es)))} \} \text{ (is - = prob } ?M)\text{)}$
  proof (rule arg-cong [where $f = \text{prob}])$
    have $?A S \cap ?A T = \{ \text{es} \in \text{space } P. \text{subgraph } (f S) \text{ (induced-subgraph } S \text{ (edge-ugraph es)))} \wedge \text{subgraph } (f T) \text{ (induced-subgraph } T \text{ (edge-ugraph es)))} \}$
    by blast
    also have ... = $?M$
    using $S \; T$ by (auto simp: subgraph-union-induced)
    finally show $?A S \cap ?A T = \ldots$
    .
  qed
also have ... = $p^n \cdot \text{card } (\text{uedges } (S \cup T, \text{uedges } (f S) \cup \text{uedges } (f T)))$
proof (rule induced-subgraph-prob)
  show $\text{uwellformed } (S \cup T, \text{uedges } (f S) \cup \text{uedges } (f T))$
  using $S(4,5)$ \text{ T}(4,5) unfolding uwellformed-def by auto
next
  show $S \cup T \subseteq S$-verts
  using $S(1)$ \text{ T}(1) unfolding S-verts-def by simp
  qed simp
also have ... = $p^n \cdot \text{card } (\text{uedges } (f S) \cup \text{uedges } (f T))$
by simp

finally have $\text{prob } (?A S \cap ?A T) = p^n \cdot \text{card } (\text{uedges } (f S) \cup \text{uedges } (f T))$
.
}

note $\text{prob-A-intersect} = \text{this}$

— Another technical detail is that our family of events $A$ are a valid instantiation
for the “$\Delta$ lemmas” from section 3.3.

have is-psi: $\text{prob-space-with-indicators } P \; ?I \; ?A$
proof
  show finite $?I$
  by (rule finite-subset[where $B = \text{Pow } \{1 .. n\}]$) auto
next
  show ?A : ?I ⊆ sets P
  unfolding sets-eq space-eq by blast
next
  let ?V = {1..?v}
  have 0 < prob (?A ?V)
    by (simp add: prob-A n p-nz)
  moreover have ?V ∈ ?I
    using n by force
  ultimately show ∃i ∈ ?I. 0 < prob (?A i)
    by blast
qed

then interpret prob-space-with-indicators P ?I ?A

— We proceed by reducing the claim of the 1-statement that the probability tends to 1 to showing that the expectation that the sum of all indicators of the respective events \( A \) tends to 0. (The actual reduction is done at the end of the proof, we merely collect the facts here.)

have compl-prob: \( 1 - \text{prob}\{es ∈ \text{space } P. \neg H ⊑ \text{edge-ugraph } es\} = \text{prob-in-class } p\{G. H ⊆ G\} n \)
  by (subst prob-compl[symmetric]) (auto simp: space-eq sets-eq intro: arg-cong[where f = prob])

have prob \{es ∈ \text{space } P. \neg H ⊑ \text{edge-ugraph } es\} ≤ \text{prob}\{es ∈ \text{space } P. Y es = 0\} (is ?compl ≤ -)
proof (rule finite-measure-mono, safe)
  fix es
  assume es ∈ space P
  hence es: wellformed (edge-ugraph es)
    unfolding space-eq by (rule wellformed-and-finite(2))
  assume H: \( \neg H ⊑ \text{edge-ugraph } es \)
  \{
    fix S
    assume S ⊆ {1..n} card S = ?v
    moreover hence finite S S ⊆ verts (edge-ugraph es)
      unfolding verts-edge-ugraph S-verts-def by (auto intro: finite-subset)
    ultimately have \( \neg \text{subgraph } f S\) (induced-subgraph S (edge-ugraph es))
      using H es by (metis fixed-selector-induced-subgraph[OF f])
    hence X S es = 0
      unfolding X-def by simp
    \}
  thus Y es = 0
    unfolding Y-def by simp
qed simp

— By applying the \( \Delta \) lemma, we obtain our central inequality. The rest of the proof gives bounds for \( \mu, \Delta_\Delta \) and quotients which occur on the right hand side.
hence compl-upper: $\text{compl} \leq 1 / \mu + \Delta_d / \mu^{-2}$

by (rule order-trans) (fact prob-$\mu$-$\Delta_d$)

— Lower bound for the expectation. We use $\binom{n}{k}$ as lower bound for $\binom{n}{k}$.

have $1 / ?v \cdot ?v \star (\text{real } n \cdot ?v \star p \cdot n \cdot ?e) = (n / ?v \cdot ?v \star p \cdot n \cdot ?e)$

by (simp add: power-divide)
also have $\ldots \leq (\text{choose } ?v) \cdot p \cdot n \cdot ?e$

proof (rule mult-right-mono, rule binomial-ge-n-over-k-pow-k)
  show $?v \leq n$
    using $n$.
  show $0 \leq p \cdot n \cdot ?e$
    using $p$ by simp
qed

also have $\ldots = (\sum S \in ?I. \ p \cdot n \cdot ?e)$

by (simp add: n-subsets)
also have $\ldots = (\sum S \in ?I. \ \text{prob } (?A S))$

by (simp add: prob-A)
also have $\ldots = \mu$

unfolding expectation-X-Y X-def using expectation-indicator by force
finally have ex-lower: $1 / (?v \cdot ?v) \cdot (\text{real } n \cdot ?v \star p \cdot n \cdot ?e) \leq \mu$

— Upper bound for the inverse expectation. Follows trivially from above.

have ex-lower-pos: $0 < 1 / ?v \cdot ?v \star (\text{real } n \cdot ?v \star p \cdot n \cdot ?e)$

proof (rule mult-pos-pos [OF vpowv-inv-gr-z mult-pos-pos])
  have $0 < \text{real } n$
    using $n$ nonempty finite unfolding nonempty-graph-def finite-graph-def

by auto
  thus $0 < \text{real } n \cdot ?v$
    by simp

next
  show $0 < p \cdot n \cdot ?e$ (uedges $H$)

using $p$-nz by simp
qed

hence $1 / \mu \leq (1 / ?v \cdot ?v \star (\text{real } n \cdot ?v \star p \cdot n \cdot ?e))$

by (rule divide-left-mono [OF ex-lower-zero-le-one mult-pos-pos [OF $\mu$-non-zero]])

hence inv-ex-upper: $1 / \mu \leq ?v \cdot ?v \star (1 / (\text{real } n \cdot ?v \star p \cdot n \cdot ?e))$

by simp

— Recall the definition of $\Delta_d$:

$$\Delta_d = \sum_{S \in I, T \in I, S \neq T, A_S, A_T \text{ not independent}} \text{Pr}[A_S \cap A_T]$$

We are going to prove an upper bound for that sum, so we can safely augment the index set by replacing it with a necessary condition.

The idea is that if the two sets $S$ and $T$ are not independent, their intersection is not empty. We prove that by contraposition, i.e. if the intersection is empty, then they are independent. This in turn can be shown using some basic properties of $f$. 42
{ 
-fixed $S, T$
-\textbf{assume} $S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I}$
-\textbf{hence} $*$: \text{prob} \,(S) \ast \text{prob} \,(T) = p \, n \ast (2 \ast ?e)$
-\textbf{using} prob-$A$ \textbf{by} (simp add: power-even-eq power2-eq-square)

\textbf{note} \ $S = I[\forall S \sqsubset \mathcal{I}]
\textbf{note} \ $T = I[\forall T \sqsubset \mathcal{I}]
\textbf{assume} \ \text{disj}: S \cap T = \{\}

\textbf{have} \ \text{prob} \,(S \cap T) = p \, n \ast \text{card} \,(\text{uedges} \,(S) \cup \text{uedges} \,(T))$
-\textbf{using} \ $(S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I})$ \textbf{by} (fact prob-$A$-intersect)
-\textbf{also have} \ $\ldots = p \, n \ast (\text{card} \,(\text{uedges} \,(S))) + \text{card} \,(\text{uedges} \,(T)))$

\textbf{proof} (rule arg-cong [OF card-Un-disjoint])
-\textbf{have} \ \text{finite-graph} \,(f S) \text{ finite-graph} \,(f T)
-\textbf{using} \ $(S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I})$ \textbf{by} (auto simp: wellformed-finite)

\textbf{thus} \ \text{finite} \,(\text{uedges} \,(f S)) \text{ finite} \,(\text{uedges} \,(f T))
-\textbf{unfolding} \ \text{finite-graph-def} \textbf{by} \ \text{auto}

\textbf{next}
-\textbf{have} \ \text{uedges} \,(f S) \subseteq \text{all-edges} \,(S) \text{ uedges} \,(f T) \subseteq \text{all-edges} \,(T)
-\textbf{using} \ $(\mathfrak{4}, 5)$ \textbf{by} (metis wellformed-all-edges)
-\textbf{moreover have} \ \text{all-edges} \,(S) \cap \text{all-edges} \,(T) = \{\}
-\textbf{by} (fact all-edges-disjoint [OF disj])

\textbf{ultimately show} \ \text{uedges} \,(f S) \cap \text{uedges} \,(f T) = \{\}
-\textbf{by} \ \text{blast}

\textbf{qed}
-\textbf{also have} \ \ldots = p \, n \ast (2 \ast ?e)$
-\textbf{using} \ $(S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I})$ \textbf{by} (simp add: mult-2)

\textbf{finally have} $**$: \text{prob} \,(S \cap T) = \ldots$

from $**$ have \text{indep} \,(S) \,(T)$
-\textbf{unfolding} \ \text{indep-def \ textbf{by} \ \text{force}}
}

\textbf{note} \ \text{indep} = \text{this}

---

Now we prove an upper bound for $\Delta_d$.

$\Delta_d = (\sum_{S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I}} \text{prob} \,(S \cap T) \text{ prob} \,(S \cap T)$
-\textbf{unfolding} \ $\Delta_d$-def $..$

---

Augmenting the index set as described above.

\textbf{also have} \ $\ldots \leq (\sum_{S \sqsubset \mathcal{I}, T \sqsubset \mathcal{I}} \text{prob} \,(S \cap T)$
-\textbf{by} (rule sum-mono [OF sum-mono2]) \textbf{by} (auto simp: indep measure-nonneg)

---

So far, we are adding the intersection probabilities over pairs of sets which have a nonempty intersection. Since we know that these intersections have at least one element (as they are nonempty) and at most $|V(H)|$ elements (by definition of
I). In this step, we will partition this sum by cardinality of the intersections.

also have \( \ldots = (\sum S \in \Omega \cdot \sum T \in \{\bigcup k \in \{1..?v\}. \{T \in \Omega. \operatorname{card} (S \cap T) = k\}). \operatorname{prob} (?A S \cap ?A T) \)

proof (rule sum.cong, rule refl, rule sum.cong)
  fix \( S \)
  assume \( S \in \Omega \)
  note \( I(2,3)[OF \ this] \)
  hence \( \{T. \ S \cap T \neq \{\}\} = (\bigcup k \in \{1..?v\}. \{T. \operatorname{card} (S \cap T) = k\}) \)
    by (simp add: partition-set-of-intersecting-sets-by-card)
  thus \( \{T \in \Omega. \ S \cap T \neq \{\}\} = (\bigcup k \in \{1..?v\}. \{T \in \Omega. \operatorname{card} (S \cap T) = k\}) \)
    by blast
qed simp

also have \( \ldots = (\sum S \in \Omega \cdot \sum k = 1..?v. \sum T \mid T \in \Omega \land \operatorname{card} (S \cap T) = k. \operatorname{prob} (?A S \cap ?A T) \)
  by (rule sum.cong, rule refl, rule sum.UNION-disjoint) auto
also have \( \ldots = (\sum k = 1..?v. \sum S \in \Omega \cdot \sum T \mid T \in \Omega \land \operatorname{card} (S \cap T) = k. \operatorname{prob} (?A S \cap ?A T) \)
  by (rule sum.swap)

— In this step, we compute an upper bound for the intersection probability and argue that it only depends on the cardinality of the intersection.

also have \( \ldots \leq (\sum k = 1..?v. \sum S \in \Omega \cdot \sum T \mid T \in \Omega \land \operatorname{card} (S \cap T) = k. \ p \ n \ \operatorname{powr} (2 \ * \ ?e \ - \ \max\text{-}density H \ * \ k) \)

proof (rule sum-mono)+
  fix \( k \)
  assume \( k : k \in \{1..?v\} \)
  fix \( S \ T \)
  assume \( S \in \Omega \ T \in \{T. \ T \in \Omega \land \operatorname{card} (S \cap T) = k\} \)
  hence \( T \in \Omega \ \and \ ST-k: \operatorname{card} (S \cap T) = k \)
    by auto
  note \( S = I[OF \ S \in \Omega]\)
  note \( T = I[OF \ T \in \Omega]\)

let \( ?cST = \operatorname{card} (\operatorname{uedges} (f S) \cap \operatorname{uedges} (f T)) \)

— We already know the intersection probability.

have \( \operatorname{prob} (?A S \cap ?A T) = \ p \ n \ \operatorname{powr} (\operatorname{card} (\operatorname{uedges} (f S) \cup \operatorname{uedges} (f T)) \)
  using \( S \in \Omega \ \& \ T \in \Omega \) by (fact prob-A-intersect)

— Now, we consider the number of edges shared by the copies of \( H \) over \( S \) and \( T \).

also have \( \ldots = \ p \ n \ \ ^{\wedge} \ (\operatorname{card} (\operatorname{uedges} (f S)) + \operatorname{card} (\operatorname{uedges} (f T)) - \ ?cST) \)
  using \( S \ T \) unfolding finite-graph-def by (simp add: card-union)
also have \( \ldots = \ p \ n \ \ ^{\wedge} \ (\ ?e \ + \ ?e - \ ?cST) \)
  by (metis isomorphic-cards(2)[OF S(4)] isomorphic-cards(2)[OF T(4)])
also have \( \ldots = \ p \ n \ \ ^{\wedge} \ (2 \ * \ ?e - \ ?cST) \)
  by (simp add: mult-2)
also have \( \ldots = \ p \ n \ \operatorname{powr} (2 \ * \ ?e - \ ?cST) \)
— Since the intersection graph is also an isomorphic subgraph of $H$, we know that its density has to be less than or equal to the maximum density of $H$. The proof is quite technical.

also have \[ \ldots \leq p \ n \ \text{powr} \ (2 \ast \ ?e - \ \text{max-density} \ H \ast k) \]

proof (rule powr-mono3)

have \(?cST = \text{density} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T)) \ast k\)

unfolding density-def by simp

also have \[\ldots \leq \text{max-density} \ (f S) \ast k\]

proof (rule rule-right-mono, cases \text{edges} \ (f S) \cap \text{edges} \ (f T) = \{\})

next

\begin{itemize}
  \item case True
    
    hence \[\text{density} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T)) = 0\]

    unfolding density-def by simp

    also have \[0 \leq \text{density} \ (f S)\]

    unfolding density-def by simp

    also have \[\text{density} \ (f S) \leq \text{max-density} \ (f S)\]

    using \(S\) by (simp add: max-density-is-max subgraph-refl)

    finally show \[\text{density} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T)) \leq \max-density \ (f S)\]

  \item case False
    
    show \[\text{density} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T)) \leq \max-density \ (f S)\]

    proof (rule max-density-is-max)

    show \[\text{finite-graph} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T))\]

    using \(T(3,6)\) by (metis finite-Int finite-graph-def fst-eqD snd-conv)

    show \[\text{nonempty-graph} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T))\]

    unfolding nonempty-graph-def using \(k \ ST-k \ False\) by force

    show \[\text{uwellformed} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T))\]

    using \(S(4,5)\) \(T(4,5)\) unfolding uwellformed-def by (metis Int-iff fst-eqD snd-eqD)

    show \[\text{subgraph} \ (S \cap T, \text{edges} \ (f S) \cap \text{edges} \ (f T)) \ (f S)\]

    using \(S(5)\) by (metis fst-eqD inf-sup-ord(1) snd-conv subgraph-def)

    qed (simp add: \(S\))

    qed simp

    also have \[\ldots = \text{max-density} \ H \ast k\]

    using assms \(S\) by (simp add: isomorphic-max-density[where \(G_1 = H\) and \(G_2 = f (S)\)])

    finally have \(?cST \leq \text{max-density} \ H \ast k\)

    ..

    thus \[2 \ast \ ?e - \text{max-density} \ H \ast k \leq 2 \ast \ ?e - \text{real} \ ?cST\]

    by linarith

    qed (auto simp: p-nz)

    finally show \[\text{prob} \ (?A S \cap ?A T) \leq \ldots\]
qed

— Further rewriting the index sets.
also have \ldots = (\sum k = 1..?v. \sum (S, T) \in (SIGMA S : ?I. \{ T \in ?I. \text{card} (S \cap T) = k\}), \ p \ n \ powr (2 * \ ?e - \ max-density H * k)) by (rule sum.cong, rule refl, rule sum.Sigma) auto
also have \ldots = (\sum k = 1..?v. \text{card} (SIGMA S : ?I. \{ T \in ?I. \text{card} (S \cap T) = k\}) * p \ n \ powr (2 * \ ?e - \ max-density H * k)) by (rule sum.cong) auto

— Here, we compute the cardinality of the index sets and use the same upper bounds for the binomial coefficients as for the 0-statement.
also have \ldots \leq (\sum k = 1..?v. ?v \sim k * (real n - (2 * ?v - k)) * p \ n \ powr (2 * \ ?e - \ max-density H * k))
proof (rule sum-mono)
fix k
assume k: k \in \{1..?v\}
let \ ?p = p \ n \ powr (2 * \ ?e - \ max-density H * k)

have \text{card} (SIGMA S : ?I. \{ T \in ?I. \text{card} (S \cap T) = k\}) = (\sum S \in ?I. \text{card} \{ T \in ?I. \text{card} (S \cap T) = k\}) (is \ ?lhs = -)
by simp
also have \ldots = (\sum S \in ?I. (?v choose k) * ((n - ?v) choose (?v - k)))
using n k by (fastforce simp: card-set-of-intersecting-sets-by-card)
also have \ldots = (n choose ?v) * ((?v choose k) * ((n - ?v) choose (?v - k)))
by (auto simp: n-subsets)
also have \ldots \leq n - ?v * ((?v choose k) * ((n - ?v) choose (?v - k)))
using n by (simp add: binomial-le-pow)
also have \ldots \leq n - ?v * ?v \sim k * ((n - ?v) choose (?v - k))
using k by (simp add: binomial-le-pow)
also have \ldots \leq n - ?v * ?v \sim k * (n - ?v) \sim (?v - k)
using n-2v by (simp add: binomial-le-pow)
also have \ldots \leq n - ?v * ?v \sim k * n \sim (?v - k)
by (simp add: power-mono)
also have \ldots = ?v \sim k * (n \sim (?v + (?v - k)))
by (simp add: power-add)
also have \ldots = ?v \sim k * n \sim (2 * ?v - k) (is - = \ ?rhs)
using k by (simp add: mult-2)
finally have \ ?lhs \leq \ ?rhs ,
hence real \ ?lhs \leq real \ ?rhs
using of-nat-le-iff by blast
moreover have 0 \leq \ ?p
by simp
ultimately have \ ?lhs * \ ?p \leq \ ?rhs * \ ?p
by (rule mult-right-mono)
also have \ldots = ?v \sim k * (real n \sim (2 * ?v - k) * \ ?p)
by simp

46
finally show \( \text{lhs} \times \text{p} \leq \ldots \).

qed

finally have delta-upper: \( \Delta_d \leq (\sum_{k = 1}^{\infty} v \cdot k \ast (\text{real } n \ast (2 \ast v - k) \ast p \ast n \ast \text{powr} (2 \ast e - \text{max-density } H \ast k))) \).

— At this point, we have established all necessary bounds.

note is-es is-ps is-compl-prob compl-upper ex-lower ex-lower-pos inv-ex-upper delta-upper

} note facts = this

— Recall our central inequality. We now prove that both summands tend to 0. This is mainly an exercise in bookkeeping and real arithmetics as no intelligent ideas are involved.

have \((\lambda n. \text{real } n \ast v \ast p \ast n \ast e) \longrightarrow 0\)
proof (rule LIMSEQ-le-zero)
have \((\lambda n. \text{real } n \ast v \ast p \ast n \ast e) \longrightarrow 0\)
proof (rule LIMSEQ-le-zero [OF eventually-sequentiallyI eventually-sequentiallyI])

fix n
show \(0 \leq \lambda n \ast \text{real } n \ast v \ast p \ast n \ast e\)
using p by simp

assume \(1 \leq n\)
have \(1 \ast (\text{real } n \ast v \ast p \ast n \ast e) = 1 \ast (\text{real } n \ast \text{powr} \ast v \ast p \ast n \ast \text{powr} \ast e)\)
using n p-nz by (simp add: powr-realpow[symmetric])
also have \(\ldots = \text{real } n \ast \text{powr} \ast \text{real } v \ast p \ast n \ast \text{powr} \ast \text{real } e\)
by (simp add: powr-minus-powr)
also have \(\ldots = (\text{real } n \ast \text{powr} \ast (\text{?v} / \text{?e})) \ast (\text{p } \ast \text{n } \ast \text{powr} \ast -1) \ast \text{powr} \ast \text{e}\)
using v-e-nz by (simp add: powr-powr)
also have \(\ldots = (\text{real } n \ast \text{powr} \ast (\text{?v} / \text{?e})) \ast (\text{p } \ast \text{n } \ast \text{powr} \ast -1) \ast \text{powr} \ast \text{e}\)
by (simp add: powr-mult)
also have \(\ldots = (\text{real } n \ast \text{powr} \ast (1 / (\text{?e} / \text{?v})) \ast (\text{p } \ast \text{n } \ast \text{powr} \ast -1) \ast \text{powr} \ast \text{e}\)
by simp
also have \(\ldots \leq (\text{real } n \ast \text{powr} \ast (1 / \text{max-density } H) \ast (\text{p } \ast \text{n } \ast \text{powr} \ast -1) \ast \text{powr} \ast \text{e}\)
apply (rule powr-mono2 [OF - mult-right-mono [OF powr-mono [OF le-imp-neg-le [OF divide-left-mono]]]]
using n v-e-nz p p-nz
by (auto simp: max-density-is-max [unfolded density-def], OF: finite finite nonempty wellformed subgraph-refl)
also have \(\ldots = (\text{real } n \ast \text{powr} \ast (1 / \text{max-density } H) \ast (1 / \text{p } \ast \text{n } \ast \text{powr} \ast 1)) \ast \text{powr} \ast \text{e}\)
by (simp add: powr-minus-powr[symmetric])
also have \(\ldots = (\text{real } n \ast \text{powr} \ast (1 / \text{max-density } H) / \text{p } \ast \text{n } \ast \text{powr} \ast \text{e}\)

47
using $p$ $p$-nz by simp
also have ..., = (subgraph-threshold $H$ $n$ / $p$ $n$) powr $?e$
unfolding subgraph-threshold-def ..
finally show $1 / (\text{real } n \^\ (#v \* p \^ n \^ \#e)) \leq (\text{subgraph-threshold } H \! / \! p\! n) \powr \#e$.

next
  show $(\lambda n. (\text{subgraph-threshold } H \! / \! p\! n) \powr \text{real } (\text{card } (\text{uedges } H)))$$
  \longrightarrow 0$
  using $p$-threshold $p$-nz $v$-$e$-nz
  by (auto simp: subgraph-threshold-def divide-nonneg-pos intro: tendsto-zero-powr)
qed
hence $(\lambda n. \#v \^ \#v \* (1 / (\text{real } n \^ \#v \* p \^ n \^ \#e)))$$
\longrightarrow 0$
by (rule LIMSEQ-const-mult)
thus $(\lambda n. \#v \^ \#v \* (1 / (\text{real } n \^ \#v \* p \^ n \^ \#e)))$$
\longrightarrow 0$
by simp

next
  show $\forall \approx n. 0 \leq 1 / \text{prob-space-with-indicators.$\mu$ } (MGn p \! n) (\#I \! n) (\#A \! n)$$
  \text{by (rule eventually-sequentiallyI \[OF less-imp-le \[OF divide-pos-pos \[OF - prob-space-with-indicators.$\mu$-non-zero[OF facts(2)]])]}$ simp+
next
  show $\forall \approx n. 1 / \text{prob-space-with-indicators.$\mu$ } (MGn p \! n) (\#I \! n) (\#A \! n) \leq \#v \^ \#v \* (1 / (\text{real } n \^ \#v \* p \^ n \^ \#e))$$
  \text{using facts(7) by (rule eventually-sequentiallyI)}$
qed
moreover have $(\lambda n. \text{prob-space-with-indicators.$\mu$ } (MGn p \! n) (\#I \! n) (\#A \! n))$$
\ll (\lambda n. (\text{prob-space-with-indicators.$\mu$ } (MGn p \! n) (\#I \! n) (\#A \! n))\^2)$
proof (rule less-fun-bounds)
  let $?num = \lambda n \ k. \#v \^ \#k \* (\text{real } n \^ \# (2 \* \#v \ - \ #k) \* p \^ n \powr (2 \* \#e - \max-density H \* \#k))$
  let $?den = \lambda n. (((1 / \#v \^ \#v) \* (\text{real } n \^ \#v \* p \^ n \^ \#e))\^2)$

  — We have to show that a sum is asymptotically smaller than a constant
term. We do that by showing that each summand is asymptotically smaller than the term.

  \{ 
  fix $k$
  assume $k$: $k \in \{1..\#v\}$
  let $?den' = \lambda n. (1 / \#v \^ \#v)\^2 \* (\text{real } n \^ \# (2 \* \#v \ + \ #k) \* p \^ n \powr (2 \* \#e - \max-density H \* \#k))$
  have $\text{den'}$: $?den' = $?den$
  by (subst power-mult-distrib) (simp add: power-mult-distrib power-even-eq)

  have $(\lambda n. \#num n \# k) \ll $ $?den'$$
  \text{proof (rule less-fun-const-quot)}$
  \quad \text{have $(\lambda n. (\text{subgraph-threshold } H \! / \! p\! n) \powr \text{max-density } H \* \#k))$$
  \longrightarrow 0$
  \quad \text{using } p$-threshold $\text{mult-pos-pos}[OF \text{max-density-gr-zero}[OF \text{finite nonempty wellformed}]$ $p$-$nz$ $k$
tends to zero power I

thus \((\lambda n. (\text{real } n ^* (2 * ?e - k) * p n \text{ powr} (2 * ?e - \text{max-density } H * k)) / (\text{real } n ^* (2 * ?e) * p n \text{ powr} (2 * ?e))) \rightarrow 0\)

proof (rule LIMSEQ-cong[OF - eventually-sequentially])

fix \(n :: \text{nat}\)

assume \(n \leq n\)

have \((\text{real } n ^* (2 * ?e - k) * p n \text{ powr} (2 * ?e - \text{max-density } H * k)) / (\text{real } n ^* (2 * ?e)) = (n \text{ powr} (2 * ?e - k) * p n \text{ powr} (2 * ?e - \text{max-density } H * k)) / (n \text{ powr} (2 * ?e) * p n \text{ powr} (2 * ?e))\) (is \(?\text{lhs} = -\))

using \(n \text{ p-nz by (simp add: powr-realpow[ symmetric])}\)

also have \(\ldots = (n \text{ powr} (2 * ?e - k) / n \text{ powr} (2 * ?e)) * (p n \text{ powr} (2 * ?e - \text{max-density } H * k) / (p n \text{ powr} (2 * ?e)))\)

by simp

also have \(\ldots = n \text{ powr} - \text{real } k * p n \text{ powr} ((2 * ?e - \text{max-density } H * k) - (2 * ?e))\)

apply (rule arg-cong[where \(y = - \text{real } k\)])

using \(k \text{ by fastforce}\)

also have \(\ldots = n \text{ powr} - \text{real } k * p n \text{ powr} - (\text{max-density } H * k)\)

by simp

also have \(\ldots = (n \text{ powr} -(1 / \text{max-density } H)) \text{ powr} (\text{max-density } H * k) * p n \text{ powr} - (\text{max-density } H * k)\)

using \(\text{max-density-gr-zero[OF finite nonempty wellformed]} \text{ by (simp add: powr-powr)}\)

also have \(\ldots = (n \text{ powr} -(1 / \text{max-density } H)) \text{ powr} (\text{max-density } H * k) * (p n \text{ powr} 1) \text{ powr} (\text{max-density } H * k)\)

by (simp add: powr-mult)

also have \(\ldots = (n \text{ powr} -(1 / \text{max-density } H) * (1 / p n \text{ powr} 1)) \text{ powr} (\text{max-density } H * k)\)

by (simp add: powr-minus-divide[ symmetric])

also have \(\ldots = (n \text{ powr} -(1 / \text{max-density } H) / p n) \text{ powr} (\text{max-density } H * k)\)

by (simp add: p p-nz)

also have \(\ldots = (\text{subgraph-threshold } H n / p n) \text{ powr} (\text{max-density } H * k)\) (is \(= ?\text{rhs}\))

unfolding \(\text{subgraph-threshold-def} \ldots\)

finally have \(?\text{lhs} = ?\text{rhs}\)

thus \(?\text{rhs} = ?\text{lhs}\)

by simp

qed

next
show \((1 / ?v \cdot ?v)^2 \neq 0\)
using \(vpow-inv-gr-z\) by \(auto\)
qed

hence \((\lambda n. ?num n k) \ll \ ?den\)
by (rule subst\([OF \ OF \ denom]\))

hence \((\lambda n. \sum k = 1..?v. \ ?num n k / \ ?den n) \longrightarrow \ (\sum k = 1..?v. \ 0)\)
by (rule tendsto-sum)

hence \((\lambda n. \sum k = 1..?v. \ ?num n k / \ ?den n) \longrightarrow \ 0\)
by \(simp\)

moreover have \((\lambda n. \sum k = 1..?v. \ ?num n k / \ ?den n) = (\lambda n. (\sum k = 1..?v. \ ?num n k) / \ ?den n)\)
by (simp add: \(sum-left-div-distrib\))

ultimately show \((\lambda n. \sum k = 1..?v. \ ?num n k) \ll \ ?den\)
by \(metis\)

show \(\forall \infty n. \ prob-space-with-indicators.\Delta_d (MGn p n) (?I n) (\ ?A n) \leq (\sum k = 1..?v. \ ?num n k)\)
using facts\((8)\) by (rule eventually-sequentiallyI)

show \(\forall \infty n. \ ?den n \leq (\prob-space-with-indicators.\mu (MGn p n) (\ ?I n) (\ ?A n)) \cdot 2\)
using facts\((5)\) facts\((6)\) by (rule eventually-sequentiallyI\([OF power-mono\[OF -less-imp-le]]\))

show \(\forall \infty n. \ 0 \leq \prob-space-with-indicators.\Delta_d (MGn p n) (\ ?I n) (\ ?A n)\)
using facts\((2)\) by (rule eventually-sequentiallyI\([OF prob-space-with-indicators.\Delta_d-nonneg]\))

show \(\forall \infty n. \ 0 < (\prob-space-with-indicators.\mu (MGn p n) (\ ?I n) (\ ?A n)) \cdot 2\)
using facts\((2)\) by (rule eventually-sequentiallyI\([OF prob-space-with-indicators.\mu-sq-non-zero]\))

show \(\forall \infty n. \ 0 < \ ?den n\)
using facts\((6)\) by (rule eventually-sequentiallyI\([OF zero-less-power]\))

qed

ultimately have \((\lambda n. \ 1 / \prob-space-with-indicators.\mu (MGn p n) (\ ?I n) (\ ?A n) + \prob-space-with-indicators.\Delta_d (MGn p n) (\ ?I n) (\ ?A n) / (\prob-space-with-indicators.\mu (MGn p n) (\ ?I n) (\ ?A n)) \cdot 2\))
\(\longrightarrow \ 0\)
by (subst add-0-left\([where \ a = 0, \ symmetric]\)) (rule tendsto-add)

— By now, we can actually perform the reduction mentioned above.

hence \((\lambda n. \probGn p n (\ ?es. \neg H \subseteq edge-space.edge-ugraph n \ ?es)) \longrightarrow \ 0\)

proof (rule LIMSEQ-le-zero)

show \(\forall \infty n. \ 0 \leq \probGn p n (\ ?es. \neg H \subseteq edge-space.edge-ugraph n \ ?es)\)
by (rule eventually-sequentiallyI) (rule measure-nonneg)

next

show \(\forall \infty n.\)
\[ \text{probGn } p \ n \ (\lambda \text{es}. \neg H \subseteq \text{edge-space.edge-ugraph } n \ \text{es}) \leq \\
\frac{1}{\mu (\text{MGn } p \ n) (\forall n) (\forall A \ n \ n)} + \\
\text{prob-space-with-indicators.\Delta_d (MGn } p \ n) (\forall n) (\forall A \ n) / (\mu (\text{MGn } p \ n) (\forall n) (\forall A \ n))^2 \]

\text{by (rule eventually-sequentiallyI[OF facts(4)])}

\textbf{qed}

\textbf{hence} (\lambda n. 1 - \text{probGn } p \ n \ (\lambda \text{es}. \neg H \subseteq \text{edge-space.edge-ugraph } n \ \text{es})) \longrightarrow 1

\textbf{using tendsto-diff[OF tendsto-const] by fastforce}

\textbf{thus} \text{prob-in-class p } \{G, H \subseteq G\} \longrightarrow 1

\text{by (rule LIMSEQ-cong[OF - eventually-sequentiallyI[OF facts(3)]]})

\textbf{qed}

\textbf{end}

\textbf{References}


