

Properties of Random Graphs – Subgraph Containment

Lars Hupel

December 14, 2021

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 $a :: 'i \Rightarrow 'a :: \{\text{monoid-mult, semiring-0}\}$

shows $(\sum i \in I. a\ i)^2 = (\sum i \in I. \sum j \in I. a\ i * a\ j)$

by (*simp only: sum-product power2-eq-square*)

lemma *sum-split*:

finite I \implies

$(\sum i \in I. \text{if } p\ i \text{ then } f\ i \text{ else } g\ i) = (\sum i \mid i \in I \wedge p\ i. f\ i) + (\sum i \mid i \in I \wedge \neg p\ i. g\ i)$

by (*simp add: sum.If-cases Int-def*)

lemma *sum-split2*:

assumes *finite I*

shows $(\sum i \mid i \in I \wedge P\ i. \text{if } Q\ i \text{ then } f\ i \text{ else } g\ i) = (\sum i \mid i \in I \wedge P\ i \wedge Q\ i. f\ i) + (\sum i \mid i \in I \wedge P\ i \wedge \neg Q\ i. g\ i)$

proof (*subst sum.If-cases*)
show $\text{finite } \{i \in I. P\ i\}$
using *assms* **by** *simp*

have $\{i \in I. P\ i\} \cap \text{Collect } Q = \{i \in I. P\ i \wedge Q\ i\} \{i \in I. P\ i\} \cap - \text{Collect } Q$
 $= \{i \in I. P\ i \wedge \neg Q\ i\}$
by *auto*
thus $\text{sum } f\ (\{i \in I. P\ i\} \cap \text{Collect } Q) + \text{sum } g\ (\{i \in I. P\ i\} \cap - \text{Collect } Q) =$
 $\text{sum } f\ \{i \in I. P\ i \wedge Q\ i\} + \text{sum } g\ \{i \in I. P\ i \wedge \neg Q\ i\}$
by *presburger*
qed

lemma *sum-upper*:
fixes $f :: 'i \Rightarrow 'a :: \text{ordered-comm-monoid-add}$
assumes $\text{finite } I \wedge i. i \in I \implies 0 \leq f\ i$
shows $(\sum i \mid i \in I \wedge P\ i. f\ i) \leq \text{sum } f\ I$
proof –
have $\text{sum } f\ I = (\sum i \in I. \text{if } P\ i \text{ then } f\ i \text{ else } f\ i)$
by *simp*
hence $\text{sum } f\ I = (\sum i \mid i \in I \wedge P\ i. f\ i) + (\sum i \mid i \in I \wedge \neg P\ i. f\ i)$
by (*simp only: sum-split[OF <finite I>]*)
moreover **have** $0 \leq (\sum i \mid i \in I \wedge \neg P\ i. f\ i)$
by (*rule sum-nonneg*) (*simp add: assms*)
ultimately show *?thesis*
by (*metis (full-types) add.comm-neutral add-left-mono*)
qed

lemma *sum-lower*:
fixes $f :: 'i \Rightarrow 'a :: \text{ordered-comm-monoid-add}$
assumes $\text{finite } I \wedge i \in I \wedge i. i \in I \implies 0 \leq f\ i\ x < f\ i$
shows $x < \text{sum } f\ I$
proof –
have $x < f\ i$ **by** *fact*
also **have** $\dots \leq \text{sum } f\ I$
using *sum-mono2[OF <finite I>, of {i} f] assms* **by** *auto*
finally show *?thesis* .
qed

lemma *sum-lower-or-eq*:
fixes $f :: 'i \Rightarrow 'a :: \text{ordered-comm-monoid-add}$
assumes $\text{finite } I \wedge i \in I \wedge i. i \in I \implies 0 \leq f\ i\ x \leq f\ i$
shows $x \leq \text{sum } f\ I$
proof –
have $x \leq f\ i$ **by** *fact*
also **have** $\dots \leq \text{sum } f\ I$
using *sum-mono2[OF <finite I>, of {i} f] assms* **by** *auto*
finally show *?thesis* .
qed

lemma *sum-left-div-distrib*:
fixes $f :: 'i \Rightarrow \text{real}$
shows $(\sum i \in I. f\ i / x) = \text{sum } f\ I / x$
proof –
have $(\sum i \in I. f\ i / x) = (\sum i \in I. f\ i * (1 / x))$
by *simp*
also have $\dots = \text{sum } f\ I * (1 / x)$
by (*rule sum-distrib-right[symmetric]*)
also have $\dots = \text{sum } f\ I / x$
by *simp*
finally show *?thesis*
qed

lemma *powr-mono3*:
fixes $x::\text{real}$
assumes $0 < x\ x < 1\ b \leq a$
shows $x\ \text{powr } a \leq x\ \text{powr } b$
proof –
have $x\ \text{powr } a = 1 / x\ \text{powr } -a$
by (*simp add: powr-minus-divide*)
also have $\dots = (1 / x)\ \text{powr } -a$
using *assms* **by** (*simp add: powr-divide*)
also have $\dots \leq (1 / x)\ \text{powr } -b$
using *assms* **by** (*simp add: powr-mono*)
also have $\dots = 1 / x\ \text{powr } -b$
using *assms* **by** (*simp add: powr-divide*)
also have $\dots = x\ \text{powr } b$
by (*simp add: powr-minus-divide*)
finally show *?thesis*
qed

lemma *card-union*: $\text{finite } A \implies \text{finite } B \implies \text{card } (A \cup B) = \text{card } A + \text{card } B - \text{card } (A \cap B)$
by (*metis card-Un-Int[symmetric] diff-add-inverse2*)

lemma *card-1-element*:
assumes $\text{card } E = 1$
shows $\exists a. E = \{a\}$
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-remove* **by** *fastforce*

moreover have $E = \text{insert } a \ ?E'$
using $\langle a \in E \rangle$ **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 $\langle \text{finite } ?E' \rangle$ **by** *simp*
thus *?thesis*
using $\langle a \in E \rangle$ **by** *blast*
qed

lemma *card-2-elements*:
assumes $\text{card } E = 2$
shows $\exists a \ b. E = \{a, b\} \wedge a \neq b$
proof –
from *assms* **obtain** a **where** $a \in E$
by *force*
let $?E' = E - \{a\}$

have *finite ?E'*
using *assms card-ge-0-finite* **by** *force*
hence $\text{card } (\text{insert } a \ ?E') = 1 + \text{card } ?E'$
using *card.insert-remove* **by** *fastforce*
moreover have $E = \text{insert } a \ ?E'$
using $\langle a \in E \rangle$ **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 $\langle a \in E \rangle$ **by** *blast*
moreover have $a \neq b$
using $\langle ?E' = \{b\} \rangle$ **by** *blast*
ultimately show *?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 $f: \text{inj-on } f \ A \ f \ ' \ A = B$
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* $(\text{metis } f(1) \ \text{inv-into-image-cancel})$
moreover have $(\lambda e. f \ ' e) \ ' (\text{Pow } A) = (\text{Pow } B)$

by (metis f(2) image-Pow-surj)
ultimately show ?thesis
unfolding bij-betw-def by simp
qed

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

lemma image-comp-cong: ($\bigwedge a. a \in A \implies f a = f (g a)$) \implies f ' A = f ' (g ' A)
by auto

abbreviation less-fun :: (nat \Rightarrow real) \Rightarrow (nat \Rightarrow real) \Rightarrow bool (infix \ll 50) where
f \ll g \equiv ($\lambda n. f n / g n \longrightarrow 0$)

context

fixes f :: nat \Rightarrow real
begin

lemma LIMSEQ-power-zero: f \longrightarrow 0 \implies 0 < n \implies ($\lambda x. f x \wedge n :: real$)
 \longrightarrow 0
by (metis power-eq-0-iff tendsto-power)

lemma LIMSEQ-cong:
assumes f \longrightarrow x $\forall^\infty n. f n = g n$
shows g \longrightarrow 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 \longrightarrow 0 $\forall^\infty n. 0 \leq f n \forall^\infty n. f n \leq g n$
shows f \longrightarrow 0
by (rule real-tendsto-sandwich[OF assms(2) assms(3) tendsto-const assms(1)])

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

lemma LIMSEQ-const-div:
assumes f \longrightarrow a c \neq 0
shows ($\lambda x. f x / c$) \longrightarrow 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** $0 < y * y'$
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 $f' \ll g' \ \forall^\infty n. f \ n \leq f' \ n \ \forall^\infty n. g' \ n \leq g \ n \ \forall^\infty n. 0 \leq f \ n \ \forall^\infty n. 0 < g \ n \ \forall^\infty n. 0 < g' \ n$
shows $f \ll g$
proof (*rule real-tendsto-sandwich*)
show $\forall^\infty n. 0 \leq f \ n / g \ n$
using *assms*(4,5) **by** *eventually-elim simp*
next
show $\forall^\infty n. f \ n / g \ n \leq f' \ n / g' \ n$
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-eventually*)
qed

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

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 $k \leq n$ $n \leq \text{card } I$ $k \leq \text{card } A$
shows $\text{card } \{B. B \subseteq I \wedge \text{card } B = n \wedge \text{card } (A \cap B) = k\} = (\text{card } A \text{ choose } k)$
 $* ((\text{card } I - \text{card } A) \text{ choose } (n - k))$

proof –

note $\text{finite-}A = \text{finite-subset}[OF \text{ assms}(1,2)]$

have $\text{card } \{B. B \subseteq I \wedge \text{card } B = n \wedge \text{card } (A \cap B) = k\} = \text{card } (\{K. K \subseteq A \wedge \text{card } K = k\} \times \{B'. B' \subseteq I - A \wedge \text{card } B' = n - k\})$ (**is** $\text{card } ?lhs = \text{card } ?rhs$)

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 $?f ' ?rhs = ?lhs$

proof (*rule set-eqI, rule iffI*)

fix B

assume $B \in ?f ' ?rhs$

then obtain $K B'$ **where** $K: K \subseteq A$ $\text{card } K = k$ $B': B' \subseteq I - A$ $\text{card } B' = n - k$ $K \cup B' = B$

by *blast*

show $B \in ?lhs$

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*(1) **by** *blast*

thus $\text{card } (A \cap B) = k$

using K **by** *simp*

qed

next

fix B

assume $B \in ?lhs$

hence $B: B \subseteq I$ $\text{card } B = n$ $\text{card } (A \cap B) = k$

by *auto*


```

let ?K = A ∩ B
let ?B' = B - A
have ?K ⊆ A card ?K = k ?B' ⊆ I - A
  using B by auto
moreover have card ?B' = n - k
  using B finite-A assms(1) by (metis Int-commute card-Diff-subset-Int
finite-Un inf.left-idem le-iff-inf sup-absorb2)
ultimately have (?K, ?B') ∈ ?rhs
  by blast
moreover have B = ?f (?K, ?B')
  by auto
ultimately show B ∈ ?f ' ?rhs
  by blast
qed
ultimately show bij-betw ?f ?rhs ?lhs
unfolding bij-betw-def ..
qed
also have ... = (∑ K | K ⊆ A ∧ card K = k. card {B'. B' ⊆ I - A ∧ card B'
= n - k})
proof (rule card-SigmaI, safe)
  show finite {K. K ⊆ A ∧ card K = k}
  by (blast intro: finite-subset[where B = Pow A] finite-A)
next
fix K
assume K ⊆ A
thus finite {B'. B' ⊆ I - A ∧ card B' = n - card K}
  using assms by auto
qed
also have ... = card {K. K ⊆ A ∧ card K = k} * card {B'. B' ⊆ I - A ∧ card
B' = n - k}
  by simp
also have ... = (card A choose k) * (card (I - A) choose (n - k))
  by (simp only: n-subsets[OF finite-A] n-subsets[OF finite-Diff[OF assms(2)]])
also have ... = (card A choose k) * ((card I - card A) choose (n - k))
  by (simp only: card-Diff-subset[OF finite-A assms(1)])
finally show ?thesis

```

qed

lemma *card-dep-pair-set*:

```

assumes finite A ∧ a. a ⊆ A ⇒ finite (f a)
shows card {(a, b). a ⊆ A ∧ card a = n ∧ b ⊆ f a ∧ card b = g a} = (∑ a | a
⊆ A ∧ card a = n. card (f a) choose g a) (is card ?S = ?C)
proof -
  have S: ?S = Sigma {a. a ⊆ A ∧ card a = n} (λa. {b. b ⊆ f a ∧ card b = g a})
(is - = Sigma ?A ?B)
  by auto

```

```

have card (Sigma ?A ?B) = (∑ a ∈ {a. a ⊆ A ∧ card a = n}. card (?B a))

```

proof (*rule card-SigmaI, safe*)
show *finite ?A*
by (*rule finite-subset[OF - finite-Collect-subsets[OF assms(1)]]*) *blast*
next
fix *a*
assume $a \subseteq A$
hence *finite (f a)*
by (*fact assms(2)*)
thus *finite (?B a)*
by (*rule finite-subset[rotated, OF finite-Collect-subsets]*) *blast*
qed
also have $\dots = ?C$
proof (*rule sum.cong*)
fix *a*
assume $a \in \{a. a \subseteq A \wedge \text{card } a = n\}$
hence *finite (f a)*
using *assms(2)* **by** *blast*
thus $\text{card } (?B a) = \text{card } (f a)$ *choose g a*
by (*fact n-subsets*)
qed *simp*
finally have $\text{card } (\text{Sigma } ?A ?B) = ?C$

thus *?thesis*
by (*subst S*)
qed

lemma *prod-cancel-nat*:
— Contributed by Manuel Eberl
fixes $f :: 'a \Rightarrow \text{nat}$
assumes $B \subseteq A$ **and** *finite A* **and** $\forall x \in B. f x \neq 0$
shows $\text{prod } f A / \text{prod } f B = \text{prod } f (A - B)$ (**is** $?A / ?B = ?C$)
proof—
from *prod.subset-diff[OF assms(1,2)]* **have** $?A = ?C * ?B$ **by** *auto*
moreover have $?B \neq 0$ **using** *assms* **by** (*simp add: finite-subset*)
ultimately show *?thesis* **by** *simp*
qed

lemma *prod-id-cancel-nat*:
— Contributed by Manuel Eberl
fixes $A :: \text{nat set}$
assumes $B \subseteq A$ **and** *finite A* **and** $0 \notin B$
shows $\prod A / \prod B = \prod (A - B)$
using *assms(1-2)* **by** (*rule prod-cancel-nat*) (*metis assms(3)*)

lemma (**in** *prob-space*) *integrable-squareD*:
— Contributed by Johannes Hölzl
fixes $X :: - \Rightarrow \text{real}$
assumes *integrable M* $(\lambda x. (X x) ^ 2)$ $X \in \text{borel-measurable } M$

```

  shows integrable M X
proof -
  have integrable M (λx. max 1 ((X x) ^ 2))
    using assms by auto
  then show integrable M X
    proof (rule Bochner-Integration.integrable-bound[OF - - always-eventually[OF all]])
      fix x show norm (X x) ≤ 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 $rind :: 'a \text{ set} \Rightarrow 'a \Rightarrow \text{real}$ where
 $rind \equiv \text{indicator}$

lemma *product-indicator*:
 $rind A x * rind B x = rind (A \cap B) x$
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 $valid\text{-}prob (p :: \text{real}) \equiv 0 \leq p \wedge p \leq 1$
abbreviation $nonzero\text{-}prob (p :: \text{real}) \equiv 0 < p \wedge p < 1$

A function $'a \Rightarrow \text{real}$ is a ‘valid probability function’ iff each value in the image is valid, and similarly for ‘nonzero’.

abbreviation $valid\text{-}prob\text{-}fun f \equiv (\forall n. \text{valid}\text{-}prob (f n))$
abbreviation $nonzero\text{-}prob\text{-}fun f \equiv (\forall n. \text{nonzero}\text{-}prob (f n))$

lemma *nonzero-fun-is-valid-fun*: $nonzero\text{-}prob\text{-}fun f \implies valid\text{-}prob\text{-}fun f$
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* $A B \iff \text{prob}(A \cap B) = \text{prob } A * \text{prob } B$

The probability of an indicator variable is equal to its expectation:

lemma *expectation-indicator*:

$A \in \text{events} \implies \text{expectation}(\text{rind } A) = \text{prob } A$

by *simp*

For a non-negative random variable X , the Markov inequality gives the following upper bound:

$$\Pr[X \geq a] \leq \frac{E[X]}{a}$$

lemma *markov-inequality*:

assumes $\bigwedge a. 0 \leq X a$ **and** *integrable* $M X 0 < t$

shows $\text{prob}\{a \in \text{space } M. t \leq X a\} \leq \text{expectation } X / t$

proof –

– proof adapted from *edge-space.Markov-inequality*, but generalized to arbitrary *prob-spaces*

have $(\int^+ x. \text{ennreal}(X x) \partial M) = (\int x. X x \partial M)$

using *assms* **by** (*intro nn-integral-eq-integral*) *auto*

thus *?thesis*

using *assms nn-integral-Markov-inequality*[*of X space M M 1 / t*]

by (*auto cong: nn-integral-cong simp: emeasure-eq-measure ennreal-mult[symmetric]*)

qed

$$\text{Var}[X] = E[X^2] - E[X]^2$$

lemma *variance-expectation*:

fixes $X :: 'a \Rightarrow \text{real}$

assumes *integrable* $M (\lambda x. (X x)^2)$ **and** $X \in \text{borel-measurable } M$

shows

integrable $M (\lambda x. (X x - \text{expectation } X)^2)$ (**is** *?integrable*)

variance $X = \text{expectation}(\lambda x. (X x)^2) - (\text{expectation } X)^2$ (**is** *?variance*)

proof –

have *int: integrable* $M X$

using *integrable-squareD[OF assms]* **by** *simp*

have $(\lambda x. (X x - \text{expectation } X)^2) = (\lambda x. (X x)^2 + (\text{expectation } X)^2 - (2 * X x * \text{expectation } X))$

by (*simp only: power2-diff*)

hence

variance $X = \text{expectation}(\lambda x. (X x)^2) + (\text{expectation } X)^2 + \text{expectation}(\lambda x. -(2 * X x * \text{expectation } X))$

```

    ?integrable
  using integral-add by (simp add: int assms prob-space)+

  thus ?variance ?integrable
    by (simp add: int power2-eq-square)+
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:

$$\Pr[|Y - \mathbb{E}[Y]| \geq s] \leq \frac{\text{Var}[X]}{a^2}$$

lemma *chebyshev-inequality*:

```

  fixes Y :: 'a ⇒ real
  assumes Y-int: integrable M (λy. (Y y) ^ 2)
  assumes Y-borel: Y ∈ borel-measurable M
  fixes s :: real
  assumes s-pos: 0 < s
  shows prob {a ∈ space M. s ≤ |Y a - expectation Y|} ≤ variance Y / s ^ 2
proof -
  let ?X = λa. (Y a - expectation Y) ^ 2
  let ?t = s ^ 2

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

```

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

corollary *chebyshev-prob-zero*:

```

  fixes Y :: 'a ⇒ real
  assumes Y-int: integrable M (λy. (Y y) ^ 2)
  assumes Y-borel: Y ∈ borel-measurable M
  assumes μ-pos: expectation Y > 0
  shows prob {a ∈ space M. Y a = 0} ≤ expectation (λy. (Y y) ^ 2) / (expectation
Y) ^ 2 - 1
proof -
  let ?s = expectation Y

  have prob {a ∈ space M. Y a = 0} ≤ prob {a ∈ space M. ?s ≤ |Y a - ?s|}

```

```

using Y-borel by (auto intro!: finite-measure-mono borel-measurable-diff borel-measurable-abs
borel-measurable-le)
also have ...  $\leq$  variance Y /  $?s^2$ 
using assms by (fact chebyshev-inequality)
also have ... = (expectation ( $\lambda y. (Y y)^2$ ) -  $?s^2$ ) /  $?s^2$ 
using Y-int Y-borel by (simp add: variance-expectation)
also have ... = expectation ( $\lambda y. (Y y)^2$ ) /  $?s^2$  - 1
using  $\mu$ -pos by (simp add: field-simps)
finally show ?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:  $A \text{ ' } I \subseteq$  events

```

```

assumes 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 μ .

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 \approx , respectively. Note that they are not the opposite of each other.

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

abbreviation *ineq-dep* :: 'i \Rightarrow 'i \Rightarrow bool **where**
ineq-dep $i j \equiv (i \neq j \wedge \neg \text{indep } (A i) (A j))$

definition $\Delta_a = (\sum i \in I. \sum j \mid j \in I \wedge i \neq j. \text{prob } (A i \cap A j))$

definition $\Delta_d = (\sum i \in I. \sum j \mid j \in I \wedge \text{ineq-dep } i j. \text{prob } (A i \cap A j))$

lemma Δ -*zero*:

assumes $\bigwedge i j. i \in I \Rightarrow j \in I \Rightarrow i \neq j \Rightarrow \text{indep } (A i) (A j)$

shows $\Delta_d = 0$
proof –
 {
 fix i
 assume $i \in I$
 hence $\{j. j \in I \wedge \text{ineq-dep } i\ j\} = \{\}$
 using *assms by auto*
 hence $(\sum j \mid j \in I \wedge \text{ineq-dep } i\ j. \text{prob } (A\ i \cap A\ j)) = 0$
 using *sum.empty by metis*
 }
hence $\Delta_d = (0 :: \text{real}) * \text{card } I$
unfolding $\Delta_d\text{-def}$ **by** *simp*
thus *?thesis*
by *simp*
qed

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

lemma *expectation-X-Y*: $\mu = (\sum i \in I. \text{expectation } (X\ i))$
unfolding $\mu\text{-def}$ *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 $\mu\text{-non-zero[simp]}$: $0 < \mu$
unfolding *expectation-X-Y*
using *expectation-X-non-zero*
by (*auto intro!: sum-lower finite-I*
simp add: expectation-indicator X-def)

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

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

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

lemma *integrable-Y-sq[simp]*: *integrable* M $(\lambda y. (Y\ y)^{\wedge 2})$
unfolding *Y-square-unfold*
by (*simp add: sets.Int less-top[symmetric]*)

lemma *measurable-Y[measurable]*: $Y \in \text{borel-measurable } M$

unfolding $Y\text{-def}[abs\text{-def}] X\text{-def}$ **by** *simp*

lemma *expectation-Y-Δ*: $expectation (\lambda x. (Y x) \hat{=} 2) = \mu + \Delta_a$

proof –

let $?ei = \lambda i j. expectation (rind (A i \cap A j))$

have $expectation (\lambda x. (Y x) \hat{=} 2) = (\sum i \in I. \sum j \in I. ?ei i j)$

unfolding *Y-square-unfold* **by** (*simp add: less-top[symmetric]*)

also have $\dots = (\sum i \in I. \sum j \in I. \text{if } i = j \text{ then } ?ei i j \text{ else } ?ei i j)$

by *simp*

also have $\dots = (\sum i \in I. (\sum j \mid j \in I \wedge i = j. ?ei i j) + (\sum j \mid j \in I \wedge i \neq j. ?ei i j))$

by (*simp only: sum-split[OF finite-I]*)

also have $\dots = (\sum i \in I. \sum j \mid j \in I \wedge i = j. ?ei i j) + (\sum i \in I. \sum j \mid j \in I \wedge i \neq j. ?ei i j)$ (**is** $= ?lhs + ?rhs$)

by (*fact sum.distrib*)

also have $\dots = \mu + \Delta_a$

proof –

have $?lhs = \mu$

proof –

{

fix i

assume $i: i \in I$

have $(\sum j \mid j \in I \wedge i = j. ?ei i j) = (\sum j \mid j \in I \wedge i = j. ?ei i i)$

by *simp*

also have $\dots = (\sum j \mid i = j. ?ei i i)$

using i **by** *metis*

also have $\dots = expectation (rind (A i))$

by *auto*

finally have $(\sum j \mid j \in I \wedge i = j. ?ei i j) = \dots$.

}

hence $?lhs = (\sum i \in I. expectation (rind (A i)))$

by *force*

also have $\dots = \mu$

unfolding *expectation-X-Y X-def* ..

finally show $?lhs = \mu$.

qed

moreover have $?rhs = \Delta_a$

proof –

{

fix $i j$

assume $i \in I j \in I$

with A **have** $A i \cap A j \in \text{events}$ **by** *blast*

hence $?ei i j = \text{prob } (A i \cap A j)$

by (*fact expectation-indicator*)

}

thus $?thesis$

unfolding $\Delta_a\text{-def}$ **by** *simp*

qed

ultimately show $?lhs + ?rhs = \mu + \Delta_a$
 by *simp*
 qed
 finally show *?thesis* .
 qed

lemma Δ -expectation-X: $\Delta_a \leq \mu^{\wedge 2} + \Delta_d$

proof -

let $?p = \lambda i j. \text{prob } (A\ i \cap A\ j)$
 let $?p' = \lambda i j. \text{prob } (A\ i) * \text{prob } (A\ j)$
 let $?ie = \lambda i j. \text{indep } (A\ i) (A\ j)$

have $\Delta_a = (\sum i \in I. \sum j \mid j \in I \wedge i \neq j. \text{if } ?ie\ i\ j\ \text{then } ?p\ i\ j\ \text{else } ?p\ i\ j)$
 unfolding Δ_a -def by *simp*
 also have $\dots = (\sum i \in I. (\sum j \mid j \in I \wedge \text{ineq-indep } i\ j. ?p\ i\ j) + (\sum j \mid j \in I \wedge \text{ineq-dep } i\ j. ?p\ i\ j))$
 by (*simp only: sum-split2[OF finite-I]*)
 also have $\dots = (\sum i \in I. \sum j \mid j \in I \wedge \text{ineq-indep } i\ j. ?p\ i\ j) + \Delta_d$ (*is - = ?lhs + -*)

unfolding Δ_d -def by (*fact sum.distrib*)

also have $\dots \leq \mu^{\wedge 2} + \Delta_d$

proof (*rule add-right-mono*)

have $(\sum i \in I. \sum j \mid j \in I \wedge \text{ineq-indep } i\ j. ?p\ i\ j) = (\sum i \in I. \sum j \mid j \in I \wedge \text{ineq-indep } i\ j. ?p'\ i\ j)$

unfolding *indep-def* by *simp*

also have $\dots \leq (\sum i \in I. \sum j \in I. ?p'\ i\ j)$

proof (*rule sum-mono*)

fix i

assume $i \in I$

show $(\sum j \mid j \in I \wedge \text{ineq-indep } i\ j. ?p'\ i\ j) \leq (\sum j \in I. ?p'\ i\ j)$

by (*rule sum-upper[OF finite-I]*) (*simp add: zero-le-mult-iff*)

qed

also have $\dots = (\sum i \in I. \text{prob } (A\ i))^{\wedge 2}$

by (*fact sum-square[symmetric]*)

also have $\dots = (\sum i \in I. \text{expectation } (X\ i))^{\wedge 2}$

unfolding *X-def* using *expectation-indicator A* by *simp*

also have $\dots = \mu^{\wedge 2}$

using *expectation-X-Y[symmetric]* by *simp*

finally show $?lhs \leq \mu^{\wedge 2}$.

qed

finally show *?thesis* .

qed

lemma *prob- μ - Δ_a* : $\text{prob } \{a \in \text{space } M. Y\ a = 0\} \leq 1 / \mu + \Delta_a / \mu^{\wedge 2} - 1$

proof -

have $\text{prob } \{a \in \text{space } M. Y\ a = 0\} \leq \text{expectation } (\lambda y. (Y\ y)^{\wedge 2}) / \mu^{\wedge 2} - 1$

unfolding μ -def by (*rule chebyshev-prob-zero*) (*simp add: μ -def[symmetric]*) +

also have $\dots = (\mu + \Delta_a) / \mu^{\wedge 2} - 1$

using *expectation-Y- Δ* by *simp*

also have $\dots = 1 / \mu + \Delta_a / \mu^2 - 1$
unfolding *power2-eq-square* **by** (*simp add: field-simps add-divide-distrib*)
finally show *?thesis* .
qed

lemma *prob- μ - Δ_d* : *prob {a \in space M. Y a = 0} \leq 1/ μ + Δ_d/μ^2*

proof –
have *prob {a \in space M. Y a = 0} \leq 1/ μ + $\Delta_a/\mu^2 - 1$*
by (*fact prob- μ - Δ_a*)
also have $\dots = (1/\mu - 1) + \Delta_a/\mu^2$
by *simp*
also have $\dots \leq (1/\mu - 1) + (\mu^2 + \Delta_d)/\mu^2$
using *divide-right-mono[OF Δ -expectation-X]* **by** *simp*
also have $\dots = 1/\mu + \Delta_d/\mu^2$
using *μ -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, 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 \implies finite (all-edges V)*

unfolding *all-edges-def*

by *simp*

corollary *complete-finite-edges*: *finite V \implies finite (uedges (complete V))*

unfolding *complete-def using all-edges-finite*

by *simp*

The sets of possible edges of disjoint sets of vertices are disjoint.

lemma *all-edges-disjoint*: $S \cap T = \{\} \implies \text{all-edges } S \cap \text{all-edges } T = \{\}$
unfolding *all-edges-def*
by *force*

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

definition *finite-graph* $G \equiv \text{finite } (\text{uverts } G) \wedge \text{finite } (\text{uedges } G)$

The complete graph is finite.

corollary *complete-finite*: $\text{finite } V \implies \text{finite-graph } (\text{complete } V)$
using *complete-finite-edges* **unfolding** *finite-graph-def* *complete-def*
by *simp*

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

definition *nonempty-graph* $G \equiv \text{uverts } G \neq \{\} \wedge \text{uedges } G \neq \{\}$

A random graph is both wellformed and finite.

lemma (in *edge-space*) *wellformed-and-finite*:
assumes $E \in \text{Pow } S\text{-edges}$
shows $\text{finite-graph } (\text{edge-ugraph } E) \text{ uwellformed } (\text{edge-ugraph } E)$
unfolding *finite-graph-def*
proof
show $\text{finite } (\text{uverts } (\text{edge-ugraph } E))$
unfolding *edge-ugraph-def* *S-verts-def* **by** *simp*
next
show $\text{finite } (\text{uedges } (\text{edge-ugraph } E))$
using *assms* **unfolding** *edge-ugraph-def* *S-edges-def* **by** (*auto intro: all-edges-finite*)
next
show $\text{uwellformed } (\text{edge-ugraph } E)$
using *complete-wellformed* **unfolding** *edge-ugraph-def* *S-edges-def* *complete-def*
uwellformed-def **by** *force*
qed

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

lemma (in *edge-space*) *cylinder-empty-prob*:
 $A \subseteq S\text{-edges} \implies \text{prob } (\text{cylinder } S\text{-edges } A \ \{\}) = p \wedge (\text{card } A)$
using *cylinder-prob* **by** *auto*

4.1 Subgraphs

definition *subgraph* :: $\text{ugraph} \Rightarrow \text{ugraph} \Rightarrow \text{bool}$ **where**
 $\text{subgraph } G' G \equiv \text{uverts } G' \subseteq \text{uverts } G \wedge \text{uedges } G' \subseteq \text{uedges } G$

lemma *subgraph-refl*: $\text{subgraph } G G$
unfolding *subgraph-def*
by *simp*

lemma *subgraph-trans*: $\text{subgraph } G'' G' \implies \text{subgraph } G' G \implies \text{subgraph } G'' G$

unfolding *subgraph-def*
by *auto*

lemma *subgraph-antisym*: $\text{subgraph } G \ G' \implies \text{subgraph } G' \ G \implies G = G'$
unfolding *subgraph-def*
by (*auto simp add: Product-Type.prod-eqI*)

lemma *subgraph-complete*:
assumes *uwellformed* G
shows *subgraph* G (*complete* (*uverts* G))
proof –
{
 fix e
 assume $e \in \text{uedges } G$
 with *assms* **have** $\text{card } e = 2$ **and** $u: \bigwedge u. u \in e \implies u \in \text{uverts } G$
 unfolding *uwellformed-def* **by** *auto*
 moreover then obtain $u \ v$ **where** $e = \{u, v\}$ $u \neq v$
 by (*metis 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 *all-edges-def* **using** $\langle u \neq v \rangle$ **by** *fastforce*
}
thus *?thesis*
unfolding *complete-def subgraph-def* **by** *auto*
qed

corollary *uwellformed-all-edges*: $\text{uwellformed } G \implies \text{uedges } G \subseteq \text{all-edges } (\text{uverts } G)$
using *subgraph-complete subgraph-def complete-def* **by** *simp*

lemma *subgraph-finite*: $\llbracket \text{finite-graph } G; \text{subgraph } G' \ G \rrbracket \implies \text{finite-graph } G'$
unfolding *finite-graph-def subgraph-def*
by (*metis rev-finite-subset*)

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

definition *subgraphs* :: $\text{ugraph} \Rightarrow \text{ugraph set}$ **where**

subgraphs $G = \{G'. \text{ subgraph } G' G\}$

definition *nonempty-subgraphs* :: *ugraph* \Rightarrow *ugraph set* **where**
nonempty-subgraphs $G = \{G'. \text{ uwellformed } G' \wedge \text{ subgraph } G' G \wedge \text{ nonempty-graph } G'\}$

lemma *subgraphs-finite*:

assumes *finite-graph* G

shows *finite* (*subgraphs* G)

proof –

have *subgraphs* $G = \{(V', E'). V' \subseteq \text{uverts } G \wedge E' \subseteq \text{uedges } G\}$

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* $G \Longrightarrow$ *finite* (*nonempty-subgraphs* G)

using *subgraphs-finite*

unfolding *nonempty-subgraphs-def* *subgraphs-def*

by *auto*

4.2 Induced subgraphs

definition *induced-subgraph* :: *uvert set* \Rightarrow *ugraph* \Rightarrow *ugraph* **where**
induced-subgraph $V G = (V, \text{uedges } G \cap \text{all-edges } V)$

lemma *induced-is-subgraph*:

$V \subseteq \text{uverts } G \Longrightarrow \text{subgraph } (\text{induced-subgraph } V G) G$

$V \subseteq \text{uverts } G \Longrightarrow \text{subgraph } (\text{induced-subgraph } V G) (\text{complete } V)$

unfolding *subgraph-def* *induced-subgraph-def* *complete-def*

by *simp+*

lemma *induced-wellformed*: *uwellformed* $G \Longrightarrow V \subseteq \text{uverts } G \Longrightarrow \text{uwellformed } (\text{induced-subgraph } V G)$

unfolding *uwellformed-def* *induced-subgraph-def* *all-edges-def*

by *force*

lemma *subgraph-union-induced*:

assumes $\text{uverts } H_1 \subseteq S$ **and** $\text{uverts } H_2 \subseteq T$

assumes *uwellformed* H_1 **and** *uwellformed* H_2

shows $\text{subgraph } H_1 (\text{induced-subgraph } S G) \wedge \text{subgraph } H_2 (\text{induced-subgraph } T G) \longleftrightarrow$

$\text{subgraph } (\text{uverts } H_1 \cup \text{uverts } H_2, \text{uedges } H_1 \cup \text{uedges } H_2) (\text{induced-subgraph } (S \cup T) G)$

unfolding *induced-subgraph-def* *subgraph-def*

apply *auto*

```

using all-edges-mono apply blast
using all-edges-mono apply blast
using assms(1,2) wellformed-all-edges[OF assms(3)] wellformed-all-edges[OF assms(4)]
all-edges-mono[OF assms(1)] all-edges-mono[OF assms(2)]
apply auto
done

```

lemma (in *edge-space*) *induced-subgraph-prob*:

```

assumes uverts  $H \subseteq V$  and uwellformed  $H$  and  $V \subseteq S\text{-verts}$ 
shows prob {es ∈ space  $P$ . subgraph  $H$  (induced-subgraph  $V$  (edge-ugraph es))}
=  $p \hat{\text{card}}$  (uedges  $H$ ) (is prob ? $A$  = -)
proof -
have prob ? $A$  = prob (cylinder  $S\text{-edges}$  (uedges  $H$ ) {})
unfolding cylinder-def space-eq subgraph-def induced-subgraph-def edge-ugraph-def
S-edges-def
by (rule arg-cong[OF Collect-cong]) (metis (no-types) assms(1,2) Pow-iff all-edges-mono
fst-conv inf-absorb1 inf-bot-left le-inf-iff snd-conv wellformed-all-edges)
also have ... =  $p \hat{\text{card}}$  (uedges  $H$ )
proof (rule cylinder-empty-prob)
have uedges  $H \subseteq$  all-edges (uverts  $H$ )
by (rule wellformed-all-edges[OF assms(2)])
also have all-edges (uverts  $H$ )  $\subseteq$  all-edges  $S\text{-verts}$ 
using assms by (auto simp: all-edges-mono[OF subset-trans])
finally show uedges  $H \subseteq$   $S\text{-edges}$ 
unfolding S-edges-def .
qed
finally show ?thesis
qed

```

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 map-ugraph :: (nat  $\Rightarrow$  nat)  $\Rightarrow$  ugraph  $\Rightarrow$  ugraph where
map-ugraph  $f$  ( $V$ ,  $E$ ) = ( $f$  ‘  $V$ , ( $\lambda e. f$  ‘  $e$ ) ‘  $E$ )

```

```

definition isomorphism :: ugraph  $\Rightarrow$  ugraph  $\Rightarrow$  (nat  $\Rightarrow$  nat)  $\Rightarrow$  bool where
isomorphism  $G_1$   $G_2$   $f \equiv$  bij-betw  $f$  (uverts  $G_1$ ) (uverts  $G_2$ )  $\wedge$   $G_2 =$  map-ugraph  $f$ 
 $G_1$ 

```

```

abbreviation isomorphic :: ugraph  $\Rightarrow$  ugraph  $\Rightarrow$  bool ( $- \simeq -$ ) where

```

$G_1 \simeq G_2 \equiv \text{uwellformed } G_1 \wedge \text{uwellformed } G_2 \wedge (\exists f. \text{isomorphism } G_1 \ G_2 \ f)$

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

lemma *map-ugraph-trans*: *map-ugraph (g ∘ f) = (map-ugraph g) ∘ (map-ugraph f)*
by (*simp add: fun-eq-iff image-image*)

lemma *map-ugraph-wellformed*:
 assumes *uwellformed G and inj-on f (uverts G)*
 shows *uwellformed (map-ugraph f G)*
unfolding *uwellformed-def*
proof *safe*
 fix *e'*
 assume *e' ∈ uedges (map-ugraph f G)*
 hence *e' ∈ (λe. f ' e) ' (uedges G)*
 by (*metis map-ugraph.simps snd-conv surjective-pairing*)
 then obtain e where *e: e' = f ' e e ∈ uedges G*
 by *blast*
 hence *card e = 2 e ⊆ uverts G*
 using *assms(1) unfolding uwellformed-def by blast+*
 thus *card e' = 2*
 using *e(1) by (simp add: card-inj-subst[OF assms(2)])*

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

lemma *map-ugraph-finite*: *finite-graph G ⇒ finite-graph (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 G₁ G₂*
 shows *subgraph (map-ugraph f G₁) (map-ugraph f G₂)*
proof –
 have *f ' uverts G₁ ⊆ f ' uverts G₂ (λe. f ' e) ' uedges G₁ ⊆ (λe. f ' e) ' 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: 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: bij-betw f_1 (uverts G_1) (uverts G_2) *bij-betw f_2 (uverts G_2) (uverts G_3)* **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 *bij-betw $?f$ (uverts G_1) (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** *uwellformed G_1 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: bij-betw f (uverts G_1) (uverts G_2)* **and** *map: $G_2 = \text{map-ugraph } f G_1$*
unfolding *isomorphism-def* **by** *auto*

let $?f' = \text{inv-into (uverts } G_1) f$
have *bij': bij-betw $?f'$ (uverts G_2) (uverts G_1)*
by (*rule bij-betw-inv-into*) *fact*
moreover **have** $\text{uverts } G_1 = ?f' \text{ ` uverts } G_2$
using *bij'* **unfolding** *bij-betw-def* **by** *force*
moreover **have** $\text{uedges } G_1 = (\lambda e. ?f' \text{ ` } e) \text{ ` uedges } G_2$
proof –
have $\text{uedges } G_1 = \text{id ` uedges } G_1$
by *simp*

also have $\dots = (\lambda e. ?f' \text{ ' } (f \text{ ' } e)) \text{ ' } \textit{uedges } G_1$
proof (*rule image-cong*)
fix a
assume $a \in \textit{uedges } G_1$
hence $a \subseteq \textit{uverts } G_1$
using *assms unfolding isomorphism-def uwellformed-def* **by** *blast*
thus $\textit{id } a = \textit{inv-into } (\textit{uverts } G_1) f \text{ ' } f \text{ ' } a$
by (*metis (full-types) id-def bij bij-betw-imp-inj-on inv-into-image-cancel*)
qed *simp*
also have $\dots = (\lambda e. ?f' \text{ ' } e) \text{ ' } ((\lambda e. f \text{ ' } e) \text{ ' } \textit{uedges } G_1)$
by (*rule image-image[symmetric]*)
also have $\dots = (\lambda e. ?f' \text{ ' } e) \text{ ' } \textit{uedges } G_2$
using *bij map* **by** (*metis map-ugraph.simps prod.collapse snd-eqD*)
finally show *?thesis*

·
qed
ultimately have *isomorphism* $G_2 G_1 ?f'$
unfolding *isomorphism-def* **by** (*metis map-ugraph.simps split-pairs*)
thus $\exists f. \textit{isomorphism } G_2 G_1 f$
by *blast*
qed (*auto simp: assms*)

lemma *isomorphic-cards*:
assumes $G_1 \simeq G_2$
shows
 $\textit{card } (\textit{uverts } G_1) = \textit{card } (\textit{uverts } G_2)$ (**is** $?V$)
 $\textit{card } (\textit{uedges } G_1) = \textit{card } (\textit{uedges } G_2)$ (**is** $?E$)

proof –
from *assms* **obtain** f **where**
 $\textit{bij}: \textit{bij-betw } f (\textit{uverts } G_1) (\textit{uverts } G_2)$ **and**
 $\textit{map}: G_2 = \textit{map-ugraph } f G_1$
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 $?g = \lambda e. f \text{ ' } e$
have $\textit{bij-betw } ?g (\textit{Pow } (\textit{uverts } G_1)) (\textit{Pow } (\textit{uverts } G_2))$
by (*rule bij-lift[OF bij]*)
moreover have $\textit{uedges } G_1 \subseteq \textit{Pow } (\textit{uverts } G_1)$
using *wellformed(1) unfolding uwellformed-def* **by** *blast*
ultimately have $\textit{card } (?g \text{ ' } \textit{uedges } G_1) = \textit{card } (\textit{uedges } G_1)$
unfolding *bij-betw-def* **by** (*metis card-inj-subst*)
thus $?E$
by (*metis map map-ugraph.simps snd-conv surjective-pairing*)
qed

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* \Rightarrow *ugraph* \Rightarrow *bool* (- \sqsubseteq -) **where**
 $G' \sqsubseteq G \equiv \text{uwellformed } G \wedge (\exists G''. G' \simeq G'' \wedge \text{subgraph } G'' G)$

lemma *subgraph-is-subgraph-isomorphic*: $\llbracket \text{uwellformed } G'; \text{uwellformed } G; \text{subgraph } G' G \rrbracket \Longrightarrow G' \sqsubseteq G$

unfolding *subgraph-isomorphic-def*

by (*metis isomorphic-refl*)

lemma *isomorphic-is-subgraph-isomorphic*: $G_1 \simeq G_2 \Longrightarrow G_1 \sqsubseteq G_2$

unfolding *subgraph-isomorphic-def*

by (*metis subgraph-refl*)

lemma *subgraph-isomorphic-refl*: $\text{uwellformed } G \Longrightarrow G \sqsubseteq G$

unfolding *subgraph-isomorphic-def*

by (*metis isomorphic-refl subgraph-refl*)

lemma *subgraph-isomorphic-pre-iso-closed*:

assumes $G_1 \simeq G_2$ **and** $G_2 \sqsubseteq G_3$

shows $G_1 \sqsubseteq G_3$

unfolding *subgraph-isomorphic-def*

proof

show *uwellformed* G_3

using *assms* **unfolding** *subgraph-isomorphic-def* **by** *blast*

next

from *assms*(2) **obtain** G_2' **where** $G_2 \simeq G_2'$ *subgraph* $G_2' G_3$

unfolding *subgraph-isomorphic-def* **by** *blast*

moreover with *assms*(1) **have** $G_1 \simeq G_2'$

by (*metis isomorphic-trans*)

ultimately show $\exists G''. G_1 \simeq G'' \wedge \text{subgraph } G'' G_3$

by *blast*

qed

lemma *subgraph-isomorphic-pre-subgraph-closed*:

assumes *uwellformed* G_1 **and** *subgraph* $G_1 G_2$ **and** $G_2 \sqsubseteq G_3$

shows $G_1 \sqsubseteq G_3$

unfolding *subgraph-isomorphic-def*

proof

show *uwellformed* G_3

using *assms* **unfolding** *subgraph-isomorphic-def* **by** *blast*

next

from *assms*(3) **obtain** G_2' **where** $G_2 \simeq G_2'$ *subgraph* $G_2' G_3$

unfolding *subgraph-isomorphic-def* **by** *blast*
then obtain f **where** *bij*: *bij-betw* f (*uverts* G_2) (*uverts* G_2') $G_2' = \text{map-ugraph } f G_2$
unfolding *isomorphism-def* **by** *blast*
let $?G_1' = \text{map-ugraph } f G_1$

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-uwellformed*[*OF assms(1)*] **unfolding** *bij-betw-def* ..
ultimately **have** $G_1 \simeq ?G_1'$
using *assms(1)* **unfolding** *isomorphism-def* **by** (*metis map-ugraph.simps*
fst-conv surjective-pairing)
moreover **have** *subgraph* $?G_1' G_3$
using *subgraph-trans*[*OF map-ugraph-preserves-sub*[*OF assms(2)*]] *bij(2)* $\langle \text{subgraph } G_2' G_3 \rangle$ **by** *simp*
ultimately **show** $\exists G'' . G_1 \simeq G'' \wedge \text{subgraph } G'' G_3$
by *blast*
qed

lemmas *subgraph-isomorphic-pre-closed = subgraph-isomorphic-pre-subgraph-closed*
subgraph-isomorphic-pre-iso-closed

lemma *subgraph-isomorphic-trans*[*trans*]:
assumes $G_1 \sqsubseteq G_2$ **and** $G_2 \sqsubseteq G_3$
shows $G_1 \sqsubseteq G_3$
proof –
from *assms(1)* **obtain** G **where** $G_1 \simeq G$ *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*: $\llbracket H \sqsubseteq G; G \simeq G' \rrbracket \implies H \sqsubseteq G'$
using *isomorphic-is-subgraph-isomorphic* *subgraph-isomorphic-trans*
by *blast*

lemmas *subgraph-isomorphic-post-closed = subgraph-isomorphic-post-iso-closed*

lemmas *subgraph-isomorphic-closed = subgraph-isomorphic-pre-closed subgraph-isomorphic-post-closed*

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 $G = \text{card} (\text{uedges } G) / \text{card} (\text{uverts } G)$

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

subgraph.

definition *max-density* :: *ugraph* \Rightarrow *real* **where**
max-density $G = \text{Lattices-Big.Max} (\text{density} \text{ ' nonempty-subgraphs } G)$

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 *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 $\exists G'. \text{density } G' = \text{max-density } G \wedge \text{subgraph } G' G \wedge \text{nonempty-graph } G' \wedge \text{finite-graph } G' \wedge \text{uwellformed } G'$

proof –

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

shows $\text{density } G' \leq \text{max-density } G$

unfolding *max-density-def*

proof (*rule Max-ge*)

show *finite* $(\text{density} \text{ ' nonempty-subgraphs } G)$

using *assms(1)* **by** (*simp* *add: nonempty-subgraphs-finite*)

next

show $\text{density } G' \in \text{density} \text{ ' nonempty-subgraphs } G$

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 $0 < \text{card} (\text{uverts } G) \wedge 0 < \text{card} (\text{uedges } G)$

using *assms* **unfolding** *finite-graph-def* *nonempty-graph-def* **by** *auto*

hence $0 < \text{density } G$

unfolding *density-def* **by** *simp*
also have *density* $G \leq \text{max-density } G$
using *assms* **by** (*simp add: max-density-is-max subgraph-refl*)
finally show *?thesis*

qed

lemma *isomorphic-density*:
assumes $G_1 \simeq G_2$
shows *density* $G_1 = \text{density } G_2$
unfolding *density-def*
using *isomorphic-cards[OF assms]*
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 *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** $f: B = \text{map-ugraph } f A \text{ bij-betw } f (\text{uverts } A) (\text{uverts } B)$
unfolding *isomorphism-def* **by** *blast*
have *wellformed*: *uwellformed* 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 *density* ‘*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* $G \subseteq \text{uverts } A$ *uwellformed* G
unfolding *nonempty-subgraphs-def* *subgraph-def* **by** *simp+*
hence *inj-on* $f (\text{uverts } G)$
using $f(2)$ **unfolding** *bij-betw-def* **by** (*metis subset-inj-on*)
hence $G \simeq \text{map-ugraph } f G$
unfolding *isomorphism-def* *bij-betw-def*

¹Some 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.

by (*metis map-ugraph.simps fst-conv surjective-pairing map-ugraph-wellformed*
⟨uwellformed G⟩)
thus $\text{density } G = \text{density } (\text{map-ugraph } f \ G)$
by (*fact isomorphic-density*)
qed
— Additionally, we show that the operations *nonempty-subgraphs* and *map-ugraph* can be swapped without changing the densities. This is an obvious result, because *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 $\dots \subseteq \text{density } \langle \text{nonempty-subgraphs } (\text{map-ugraph } f \ A)$
proof (*rule image-mono, rule subsetI*)
fix G''
assume $G'' \in \text{map-ugraph } f \ \langle \text{nonempty-subgraphs } A$
then obtain G' **where** $G\text{-subst: } G'' = \text{map-ugraph } f \ G' \ G' \in \text{nonempty-subgraphs}$
 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* $f \ (\text{uverts } G')$
using f **unfolding** *bij-betw-def subgraph-def* **by** (*metis subset-inj-on*)
hence *uwellformed* G''
using *map-ugraph-wellformed* $G' \ G\text{-subst}$ **by** *simp*
moreover have *nonempty-graph* G''
using $G' \ G\text{-subst}$ **unfolding** *nonempty-graph-def* **by** (*metis map-ugraph.simps*
fst-conv snd-conv surjective-pairing empty-is-image)
moreover have *subgraph* $G'' \ (\text{map-ugraph } f \ A)$
using *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 $\text{density } \langle \text{nonempty-subgraphs } A \subseteq \text{density } \langle \text{nonempty-subgraphs}$
 $(\text{map-ugraph } f \ A)$

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 **iso** **unfolding** *nonempty-subgraphs-def* **by** (*simp add: subgraph-refl*)
thus $\text{density } \langle \text{nonempty-subgraphs } A \neq \{\}$
by *blast*
next
have *finite* (*nonempty-subgraphs* (*map-ugraph* $f \ A$))
by (*rule nonempty-subgraphs-finite[OF map-ugraph-finite[OF A(2)]]*)
thus *finite* ($\text{density } \langle \text{nonempty-subgraphs } (\text{map-ugraph } f \ A)$)
by *blast*
qed
hence $\text{max-density } A \leq \text{max-density } B$
by (*subst* f)

```

}
note le = this

show ?thesis
  using le[OF assms(2) assms(4) assms(1)] le[OF assms(3) assms(5) isomor-
phic-sym[OF assms(1)]]
  by (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

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

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.

lemma *ex-fixed-selector*:

assumes *uwellformed H and finite-graph H*
obtains *f where is-fixed-selector H f*

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.

```

let ?bij = λV. SOME g. bij-betw g (uverts H) V
let ?f = λV. map-ugraph (?bij V) H
{
  fix V :: uvert set
  assume finite V card (uverts H) = card V
  moreover have finite (uverts H)
    using assms unfolding finite-graph-def by simp
  ultimately have bij-betw (?bij V) (uverts H) V
    by (metis finite-same-card-bij someI-ex)
  moreover hence *: uverts (?f V) = V ∧ uwellformed (?f V)
    using map-ugraph-wellformed[OF assms(1)]
    by (metis bij-betw-def map-ugraph.simps fst-conv surjective-pairing)
  ultimately have ** : H ≃ ?f V
    unfolding isomorphism-def using assms(1) by auto

```

²We call such a selector *fixed* because its result is deterministic.

```

    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  $\text{card } (uverts \ H) = \text{card } V$  and finite  $V$ 
  assumes sub: subgraph  $(f \ V)$  (induced-subgraph  $V \ G$ ) and  $V: V \subseteq uverts \ G$  and
   $G: \text{wellformed } G$ 
  shows  $H \sqsubseteq G$ 
proof -
  have post:  $H \simeq f \ V$   $uverts \ (f \ V) = V$ 
    using assms unfolding is-fixed-selector-def by auto

  have  $H \sqsubseteq f \ V$ 
    by (rule isomorphic-is-subgraph-isomorphic)
      (simp add: post)
  also have  $f \ V \sqsubseteq \text{induced-subgraph } V \ G$ 
    by (rule subgraph-is-subgraph-isomorphic)
      (auto simp: induced-wellformed[OF G V] post sub)
  also have  $\dots \sqsubseteq G$ 
    by (rule subgraph-is-subgraph-isomorphic[OF induced-wellformed])
      (auto simp:  $G \ V$  induced-is-subgraph(I)[OF V])
  finally show  $H \sqsubseteq G$ 
qed
end

```

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  $C \equiv \forall G \in C. \forall G'. G \simeq G' \longrightarrow G' \in C$ 

```

```

abbreviation prob-in-class :: (nat  $\Rightarrow$  real)  $\Rightarrow$  ugraph-class  $\Rightarrow$  nat  $\Rightarrow$  real where
prob-in-class  $p \ c \ n \equiv \text{prob}Gn \ p \ n$  (les. 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”).

definition *is-threshold* :: *ugraph-class* \Rightarrow (*nat* \Rightarrow *real*) \Rightarrow *bool* **where**
is-threshold *c* *t* \equiv *ugraph-property* *c* \wedge ($\forall p$. *nonzero-prob-fun* *p* \longrightarrow
(*p* \ll *t* \longrightarrow *prob-in-class* *p* *c* \longrightarrow 0) \wedge
(*t* \ll *p* \longrightarrow *prob-in-class* *p* *c* \longrightarrow 1))

lemma *is-thresholdI*[*intro*]:
assumes *ugraph-property* *c*
assumes $\bigwedge p$. [*nonzero-prob-fun* *p*; *p* \ll *t*] \Longrightarrow *prob-in-class* *p* *c* \longrightarrow 0
assumes $\bigwedge p$. [*nonzero-prob-fun* *p*; *t* \ll *p*] \Longrightarrow *prob-in-class* *p* *c* \longrightarrow 1
shows *is-threshold* *c* *t*
using *assms* **unfolding** *is-threshold-def* **by** *blast*
end

6 The subgraph threshold theorem

theory *Subgraph-Threshold*
imports
Ugraph-Properties
begin

lemma (**in** *edge-space*) *measurable-pred*[*measurable*]: *Measurable.pred* *P* *Q*
by (*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 ρ' denotes *max-density*.

definition *subgraph-threshold* :: *ugraph* \Rightarrow *nat* \Rightarrow *real* **where**
subgraph-threshold *H* *n* = *n* *powr* $(-(1 / \text{max-density } H))$

theorem
assumes *nonempty*: *nonempty-graph* *H* **and** *finite*: *finite-graph* *H* **and** *well-formed*: *uwellformed* *H*

shows *is-threshold* $\{G. H \sqsubseteq G\}$ (*subgraph-threshold* H)
proof
show *ugraph-property* $\{G. H \sqsubseteq G\}$
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* $H_0 = \text{max-density } H \text{ subgraph } H_0$ H *nonempty-graph*
 H_0 *finite-graph* H_0 *uwellformed* H_0
using *subgraph-has-max-density* *assms* **by** *blast*
hence *card*: $0 < \text{card } (\text{uverts } H_0)$ $0 < \text{card } (\text{uedges } H_0)$
unfolding *nonempty-graph-def* *finite-graph-def* **by** *auto*

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

assume $p\text{-nz}$: *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 $p \ll \text{subgraph-threshold } H$
moreover
{
fix n
have $p \ n / n \ \text{powr } (-(1 / \text{max-density } H)) = p \ n * n \ \text{powr } (1 / \text{max-density } H)$
by (*simp add: powr-minus-divide*)
also have $\dots = p \ n * n \ \text{powr } (1 / \text{density } H_0)$
using H_0 **by** *simp*
also have $\dots = p \ n * n \ \text{powr } (?v / ?e)$
using *card* **unfolding** *density-def* **by** *simp*
finally have $p \ n / n \ \text{powr } (-(1 / \text{max-density } H)) = \dots$
.
}
ultimately have $(\lambda n. p \ n * n \ \text{powr } (?v / ?e)) \longrightarrow 0$
unfolding *subgraph-threshold-def* **by** *simp*
moreover have $\bigwedge n. 1 \leq n \implies 0 < p \ n * n \ \text{powr } (?v / ?e)$
by (*auto simp: p-nz*)
ultimately have $(\lambda n. (p \ n * n \ \text{powr } (?v / ?e)) \ \text{powr } ?e) \longrightarrow 0$

using $\text{card}(2) p$ **by** (*force intro: tendsto-zero-powI*)
hence *limit*: $(\lambda n. p n \text{ powr } ?e * n \text{ powr } ?v) \longrightarrow 0$
by (*rule LIMSEQ-cong[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 : *edge-space* $n (p n)$
by *unfold-locales (auto simp: p)*

let $?graph\text{-of} = ES.\text{edge-ugraph}$

— 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 $?X = \lambda H_0'. \text{rind } \{es \in \text{space } ES.P. \text{ subgraph } H_0' (?graph\text{-of } es) \wedge H_0 \simeq H_0'\}$
let $?I = \{(v, e). v \subseteq \{1..n\} \wedge \text{card } v = ?v \wedge e \subseteq \text{all-edges } v \wedge \text{card } e = ?e\}$
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 . Observe that in that case, Y takes a value greater or equal than 1.

have *prob-in-class* $p \{G. H \sqsubseteq G\} n = \text{prob}Gn p n (\lambda es. H \sqsubseteq ?graph\text{-of } es)$
by *simp*
also have $\dots \leq \text{prob}Gn p n (\lambda es. 1 \leq ?Y es)$
proof (*rule ES.finite-measure-mono, safe*)
fix es
assume es : $es \in \text{space } (MGn p n)$

assume $H \sqsubseteq ?graph\text{-of } es$
hence $H_0 \sqsubseteq ?graph\text{-of } es$ — 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' : *subgraph* $H_0' (?graph\text{-of } es) H_0 \simeq H_0'$
unfolding *subgraph-isomorphic-def*
by *blast*

show $1 \leq ?Y es$
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 $1 \leq ?X H_0' es$ — 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  $\langle H_0 \simeq H_0' \rangle$ ])+
      moreover have  $uedges\ H_0' \subseteq all\_edges\ (uverts\ H_0')$ 
        using  $H_0'$  by (simp add: wellformed-all-edges)
      ultimately show  $H_0' \in ?I$ 
        by auto
    next
      have  $?I \subseteq subgraphs\ (complete\ \{1..n\})$ 
      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  $finite\ ?I$ 
        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 ...  $\leq ES.expectation\ ?Y / 1$ 
  by (rule prob-space.markov-inequality) (auto simp: ES.prob-space-P sum-nonneg)
also have ... =  $ES.expectation\ ?Y$ 
  by simp
also have ... =  $(\sum H_0' \in ?I. ES.expectation\ (?X\ H_0'))$ 
  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 ...  $\leq (\sum H_0' \in ?I. p\ n \wedge ?e)$ 
  proof (rule sum-mono)
    fix  $H_0'$ 
    assume  $H_0': H_0' \in ?I$ 
    have  $ES.expectation\ (?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 ...  $\leq ES.prob\ \{es \in space\ ES.P.\ uedges\ H_0' \subseteq es\}$ 
      unfolding subgraph-def by (rule ES.finite-measure-mono) (auto simp:
    ES.sets-eq ES.space-eq)
    also have ... =  $ES.prob\ (cylinder\ ES.S-edges\ (uedges\ H_0')\ \{\})$ 
      unfolding cylinder-def ES.space-eq by simp
    also have ... =  $p\ n \wedge card\ (uedges\ H_0')$ 
      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  $\dots = p\ n \wedge ?e$ 
using  $H_0'$  by fastforce
finally show ES.expectation  $(?X\ H_0') \leq \dots$ 
.
qed

```

— Since we have a sum of constant summands, we can rewrite it as a product.

```

also have  $\dots = card\ ?I * p\ n \wedge ?e$ 
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  $\dots = ((n\ choose\ ?v) * ((?v\ choose\ 2)\ choose\ ?e)) * p\ n \wedge ?e$ 
proof (rule arg-cong[where  $x = card\ ?I$ ])
have  $card\ ?I = (\sum v \mid v \subseteq \{1..n\} \wedge card\ v = ?v.\ card\ (all-edges\ v)\ choose\ ?e)$ 
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  $\dots = (\sum v \mid v \subseteq \{1..n\} \wedge card\ v = ?v.\ (?v\ choose\ 2)\ choose\ ?e)$ 
proof (rule sum.cong)
fix  $v$ 
assume  $v \in \{v.\ v \subseteq \{1..n\} \wedge card\ v = ?v\}$ 
hence  $v \subseteq \{1..n\}$   $card\ v = ?v$ 
by auto
thus  $card\ (all-edges\ v)\ choose\ ?e = (?v\ choose\ 2)\ choose\ ?e$ 
by (simp add: card-all-edges finite-subset)
qed rule
also have  $\dots = card\ (\{v.\ v \subseteq \{1..n\} \wedge card\ v = ?v\}) * ((?v\ choose\ 2)\ choose\ ?e)$ 
by simp
also have  $\dots = (n\ choose\ ?v) * ((?v\ choose\ 2)\ choose\ ?e)$ 
by (simp add: n-subsets)
finally show  $card\ ?I = \dots$ 
.
qed
also have  $\dots = (n\ choose\ ?v) * (((?v\ choose\ 2)\ choose\ ?e) * p\ n \wedge ?e)$ 
by simp

```

— Here, we use n^k as an upper bound for $\binom{n}{k}$.

```

also have  $\dots \leq (n \wedge ?v) * (((?v\ choose\ 2)\ choose\ ?e) * p\ n \wedge ?e)$  (is - ≤ - *)

```

```

?r)
  proof (rule mult-right-mono)
    have  $n \text{ choose } ?v \leq n \wedge ?v$ 
      by (rule binomial-le-pow) (rule n)
    thus  $\text{real } (n \text{ choose } ?v) \leq \text{real } (n \wedge ?v)$ 
      by (metis of-nat-le-iff)
    next
      show  $0 \leq ?r$  using  $p$  by simp
    qed
  also have  $\dots \leq ((?v \text{ choose } 2) \text{ choose } ?e) * (p \wedge ?e * n \wedge ?v)$  (is  $- \leq ?factor$ 
* -)
    by simp
  also have  $\dots = ?factor * (p \wedge ?e * n \wedge ?v)$ 
    using  $n \text{ card}(1) \langle \text{nonzero-prob-fun } p \rangle$  by (simp add: powr-realpow)

  finally have  $\text{prob-in-class } p \{G. H \sqsubseteq G\} n \leq ?factor * (p \wedge ?e * n \wedge ?v)$ 
}

```

— The final upper bound is a multiple of the expression which we have proven to tend to 0 in the beginning.

```

thus  $\text{prob-in-class } p \{G. H \sqsubseteq G\} \longrightarrow 0$ 
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  $?v = \text{card } (\text{uverts } H)$ 
let  $?e = \text{card } (\text{uedges } H)$ 

```

— We observe that several terms involving $|V(H)|$ are positive.

```

have  $v\text{-e-nz: } 0 < \text{real } ?v \ 0 < \text{real } ?e$ 
using  $\text{nonempty finite unfolding nonempty-graph-def finite-graph-def}$  by auto
hence  $0 < \text{real } ?v \wedge ?v$  by simp
hence  $\text{powv-inv-gr-z: } 0 < 1 / ?v \wedge ?v$  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  $?A = \lambda n. \lambda S. \{es \in \text{space } (\text{edge-space.P } n (p \ w \ n)). \text{subgraph } (f \ S) (\text{induced-subgraph } S (\text{edge-space.edge-ugraph } n \ es))\}$ 
let  $?I = \lambda n. \{S. S \subseteq \{1..n\} \wedge \text{card } S = ?v\}$ 

```

```

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:  $2 * ?v \leq n$ 
  hence n:  $?v \leq n$ 
    by simp

  have is-es: edge-space (p n)
    by unfold-locales (auto simp: p)

  then interpret edge-space n p n
  .

  let ?A = ?A n
  let ?I = ?I n

  — A nice potpourri with some technical facts about S.
  {
    fix S
    assume S ∈ ?I
    hence 0:  $S \subseteq \{1..n\}$   $?v = \text{card } S$  finite S
      by (auto intro: finite-subset)
    hence 1:  $H \simeq f S$  wverts (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 isomor-
      phic-cards(2) nonempty nonempty-graph-def prod.collapse snd-conv)
    note 0 1 2 3
  }
  note I = 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 ∈ ?I
  note S' = I[OF S]

```

```

have prob (?A S) = p n ^ ?e
  using isomorphic-cards(2)[OF S'(4)] S' by (simp add: S-verts-def in-
    duced-subgraph-prob)
}
note prob-A = this

{
fix S T
assume S ∈ ?I note S = I[OF this]
assume T ∈ ?I note T = I[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 prob (?A S ∩ ?A T) = prob {es ∈ space P. subgraph (S ∪ T, uedges
    (f S) ∪ uedges (f T)) (induced-subgraph (S ∪ T) (edge-ugraph es))} (is - = prob
    ?M)
proof (rule arg-cong[where f = prob])
  have ?A S ∩ ?A T = {es ∈ space P. subgraph (f S) (induced-subgraph S
    (edge-ugraph es)) ∧ subgraph (f T) (induced-subgraph T (edge-ugraph es))}
  by blast
  also have ... = ?M
  using S T by (auto simp: subgraph-union-induced)
  finally show ?A S ∩ ?A T = ...
.
qed
also have ... = p n ^ card (uedges (S ∪ T, uedges (f S) ∪ uedges (f T)))
proof (rule induced-subgraph-prob)
  show wellformed (S ∪ T, uedges (f S) ∪ uedges (f T))
  using S(4,5) T(4,5) unfolding wellformed-def by auto
next
  show S ∪ T ⊆ S-verts
  using S(1) T(1) unfolding S-verts-def by simp
qed simp
also have ... = p n ^ card (uedges (f S) ∪ uedges (f T))
by simp

finally have prob (?A S ∩ ?A T) = p n ^ card (uedges (f S) ∪ uedges (f T))
.
}
note prob-A-intersect = this

```

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

```

have is-psi: prob-space-with-indicators P ?I ?A
proof
  show finite ?I
  by (rule finite-subset[where B = Pow {1..n}]) auto
next

```



```

show ?A ‘ ?I  $\subseteq$  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  $\in$  ?I
    using n by force
  ultimately show  $\exists i \in ?I. 0 < \text{prob} (?A i)$ 
    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 - prob {es  $\in$  space P.  $\neg$  H  $\sqsubseteq$  edge-ugraph es} = prob-in-class
p {G. H  $\sqsubseteq$  G} n
  by (subst prob-compl[symmetric]) (auto simp: space-eq sets-eq intro: arg-cong[where
f = prob])

```

```

have prob {es  $\in$  space P.  $\neg$  H  $\sqsubseteq$  edge-ugraph es}  $\leq$  prob {es  $\in$  space P. Y es
= 0} (is ?compl  $\leq$  -)

```

```

proof (rule finite-measure-mono, safe)
  fix es
  assume es  $\in$  space P
  hence es: uwellformed (edge-ugraph es)
    unfolding space-eq by (rule wellformed-and-finite(2))
  assume H:  $\neg$  H  $\sqsubseteq$  edge-ugraph es
  {
    fix S
    assume S  $\subseteq$  {1..n} card S = ?v
    moreover hence finite S S  $\subseteq$  uverts (edge-ugraph es)
      unfolding uverts-edge-ugraph S-verts-def by (auto intro: finite-subset)
    ultimately have  $\neg$  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 Δ lemma, we obtain our central inequality. The rest of the proof gives bounds for μ , Δ_d and quotients which occur on the right hand side.

```

hence compl-upper: ?compl  $\leq$  1 /  $\mu$  +  $\Delta_d$  /  $\mu^2$ 

```

by (rule order-trans) (fact prob- μ - Δ_d)

— Lower bound for the expectation. We use $\left(\frac{n}{k}\right)^k$ as lower bound for $\binom{n}{k}$.
have $1 / ?v \wedge ?v * (\text{real } n \wedge ?v * p n \wedge ?e) = (n / ?v) \wedge ?v * p n \wedge ?e$
 by (simp add: power-divide)
also have $\dots \leq (n \text{ choose } ?v) * p n \wedge ?e$
proof (rule mult-right-mono, rule binomial-ge-n-over-k-pow-k)
 show $?v \leq n$
 using n .
 show $0 \leq p n \wedge ?e$
 using p by simp
 qed
also have $\dots = (\sum S \in ?I. p n \wedge ?e)$
 by (simp add: n-subsets)
also have $\dots = (\sum S \in ?I. \text{prob } (?A S))$
 by (simp add: prob-A)
also have $\dots = \mu$
 unfolding expectation-X-Y X-def using expectation-indicator by force
finally have ex-lower: $1 / (?v \wedge ?v) * (\text{real } n \wedge ?v * p n \wedge ?e) \leq \mu$
 .

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

have ex-lower-pos: $0 < 1 / ?v \wedge ?v * (\text{real } n \wedge ?v * p n \wedge ?e)$
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 $0 < \text{real } n \wedge ?v$
 by simp
 next
 show $0 < p n \wedge \text{card } (\text{uedges } H)$
 using p -nz by simp
 qed
hence $1 / \mu \leq 1 / (1 / ?v \wedge ?v * (\text{real } n \wedge ?v * p n \wedge ?e))$
by (rule divide-left-mono[OF ex-lower zero-le-one mult-pos-pos[OF μ -non-zero]])
hence inv-ex-upper: $1 / \mu \leq ?v \wedge ?v * (1 / (\text{real } n \wedge ?v * p n \wedge ?e))$
 by simp

— Recall the definition of Δ_d :

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

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 .

{

```

fix  $S T$ 
assume  $S \in ?I \ T \in ?I$ 
hence  $*$ :  $\text{prob } (?A \ S) * \text{prob } (?A \ T) = p \ n \ ^{\wedge} (2 * ?e)$ 
  using prob-A by (simp add: power-even-eq power2-eq-square)

note  $S = I[OF \ \langle S \in ?I \rangle]$ 
note  $T = I[OF \ \langle T \in ?I \rangle]$ 
assume disj:  $S \cap T = \{\}$ 

have  $\text{prob } (?A \ S \cap ?A \ T) = p \ n \ ^{\wedge} \text{card } (\text{uedges } (f \ S) \cup \text{uedges } (f \ T))$ 
  using  $\langle S \in ?I \rangle \ \langle T \in ?I \rangle$  by (fact prob-A-intersect)
also have  $\dots = p \ n \ ^{\wedge} (\text{card } (\text{uedges } (f \ S)) + \text{card } (\text{uedges } (f \ T)))$ 
  proof (rule arg-cong[OF card-Un-disjoint])
    have finite-graph  $(f \ S)$  finite-graph  $(f \ T)$ 
      using  $S \ T$  by (auto simp: wellformed-finite)
    thus finite  $(\text{uedges } (f \ S))$  finite  $(\text{uedges } (f \ T))$ 
      unfolding finite-graph-def by auto
  next
    have  $\text{uedges } (f \ S) \subseteq \text{all-edges } S$   $\text{uedges } (f \ T) \subseteq \text{all-edges } T$ 
      using  $S(4,5)$   $T(4,5)$  by (metis wellformed-all-edges)+
    moreover have  $\text{all-edges } S \cap \text{all-edges } T = \{\}$ 
      by (fact all-edges-disjoint[OF disj])
    ultimately show  $\text{uedges } (f \ S) \cap \text{uedges } (f \ T) = \{\}$ 
      by blast
  qed
also have  $\dots = p \ n \ ^{\wedge} (2 * ?e)$ 
using isomorphic-cards(2)[OF isomorphic-sym[OF S(4)]] isomorphic-cards(2)[OF
isomorphic-sym[OF T(4)]] by (simp add: mult-2)
finally have  $**$ :  $\text{prob } (?A \ S \cap ?A \ T) = \dots$ 
  .

from  $* **$  have indep  $(?A \ S) \ (?A \ T)$ 
  unfolding indep-def by force
}
note indep = this

```

— Now we prove an upper bound for Δ_d .

```

have  $\Delta_d = (\sum S \in ?I. \sum T \mid T \in ?I \wedge \text{ineq-dep } S \ T. \text{prob } (?A \ S \cap ?A \ T))$ 
  unfolding  $\Delta_d\text{-def}$  ..

```

— Augmenting the index set as described above.

```

also have  $\dots \leq (\sum S \in ?I. \sum T \mid T \in ?I \wedge S \cap T \neq \{\}. \text{prob } (?A \ S \cap ?A \ T))$ 
  by (rule sum-mono[OF sum-mono2]) (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 $\dots = (\sum S \in ?I. \sum T \in (\bigcup k \in \{1..?v\}. \{T \in ?I. \text{card}(S \cap T) = k\}))$. *prob (?A S \cap ?A T)*
proof (*rule sum.cong, rule refl, rule sum.cong*)
fix S
assume $S \in ?I$
note $I(2,3)[\text{OF this}]$
hence $\{T. S \cap T \neq \{\}\} = (\bigcup k \in \{1..?v\}. \{T. \text{card}(S \cap T) = k\})$
by (*simp add: partition-set-of-intersecting-sets-by-card*)
thus $\{T \in ?I. S \cap T \neq \{\}\} = (\bigcup k \in \{1..?v\}. \{T \in ?I. \text{card}(S \cap T) = k\})$
by *blast*
qed *simp*
also have $\dots = (\sum S \in ?I. \sum k = 1..?v. \sum T \mid T \in ?I \wedge \text{card}(S \cap T) = k$.
prob (?A S \cap ?A T))
by (*rule sum.cong, rule refl, rule sum.UNION-disjoint*) *auto*
also have $\dots = (\sum k = 1..?v. \sum S \in ?I. \sum T \mid T \in ?I \wedge \text{card}(S \cap T) = k$.
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 $\dots \leq (\sum k = 1..?v. \sum S \in ?I. \sum T \mid T \in ?I \wedge \text{card}(S \cap T) = k$.
*p n powr (2 * ?e - max-density H * k)*)
proof (*rule sum-mono*)
fix k
assume $k: k \in \{1..?v\}$
fix $S T$
assume $S \in ?I T \in \{T. T \in ?I \wedge \text{card}(S \cap T) = k\}$
hence $T \in ?I$ **and** $ST-k: \text{card}(S \cap T) = k$
by *auto*
note $S = I[\text{OF } \langle S \in ?I \rangle]$
note $T = I[\text{OF } \langle T \in ?I \rangle]$

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

— We already know the intersection probability.

have *prob (?A S \cap ?A T) = p n ^ card(uedges(f S) \cup uedges(f T))*
using $\langle S \in ?I \rangle \langle T \in ?I \rangle$ **by** (*fact prob-A-intersect*)

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

also have $\dots = p n ^ (\text{card}(\text{uedges}(f S)) + \text{card}(\text{uedges}(f T)) - ?cST)$
using $S T$ **unfolding** *finite-graph-def* **by** (*simp add: card-union*)
also have $\dots = p n ^ (?e + ?e - ?cST)$
by (*metis isomorphic-cards(2)[OF S(4)] isomorphic-cards(2)[OF T(4)]*)
also have $\dots = p n ^ (2 * ?e - ?cST)$
by (*simp add: mult-2*)
also have $\dots = p n \text{ powr } (2 * ?e - ?cST)$
using *p-nz* **by** (*simp add: powr-realpow*)
also have $\dots = p n \text{ powr } (\text{real } (2 * ?e) - \text{real } ?cST)$

using *isomorphic-cards*[*OF S(4)*] *S(6)* **by** (*metis of-nat-diff card-mono finite-graph-def inf-le1 mult-le-mono mult-numeral-1 numeral-One one-le-numeral*)

— 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 ... ≤ p n powr (2 * ?e - max-density H * k)
proof (rule powr-mono3)
  have ?cST = density (S ∩ T, uedges (f S) ∩ uedges (f T)) * k
  unfolding density-def using k ST-k by simp
  also have ... ≤ max-density (f S) * k
  proof (rule mult-right-mono, cases uedges (f S) ∩ uedges (f T) = {})
    case True
      hence density (S ∩ T, uedges (f S) ∩ uedges (f T)) = 0
      unfolding density-def by simp
      also have 0 ≤ density (f S)
      unfolding density-def by simp
      also have density (f S) ≤ max-density (f S)
      using S by (simp add: max-density-is-max subgraph-refl)
      finally show density (S ∩ T, uedges (f S) ∩ uedges (f T)) ≤
max-density (f S)
    .
  next
    case False
      show density (S ∩ T, uedges (f S) ∩ uedges (f T)) ≤ max-density (f
S)

      proof (rule max-density-is-max)
        show finite-graph (S ∩ T, uedges (f S) ∩ uedges (f T))
        using T(3,6) by (metis finite-Int finite-graph-def fst-eqD snd-conv)
        show nonempty-graph (S ∩ T, uedges (f S) ∩ uedges (f T))
        unfolding nonempty-graph-def using k ST-k False by force
        show wellformed (S ∩ T, uedges (f S) ∩ uedges (f T))
        using S(4,5) T(4,5) unfolding wellformed-def by (metis
Int-iff fst-eqD snd-eqD)
        show subgraph (S ∩ T, uedges (f S) ∩ uedges (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 ... = max-density H * k
      using assms S by (simp add: isomorphic-max-density[where G1 = H
and G2 = f S])
      finally have ?cST ≤ max-density H * k
    .
  thus 2 * ?e - max-density H * k ≤ 2 * ?e - real ?cST
  by linarith
  qed (auto simp: p-nz)
  finally show prob (?A S ∩ ?A T) ≤ ...
    .
qed

```

— Further rewriting the index sets.

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

also have ... = $(\sum k = 1..?v. card (SIGMA S : ?I. \{T \in ?I. card (S \cap T) = k\}) * p n powr (2 * ?e - max-density H * k))$
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 ... $\leq (\sum k = 1..?v. ?v \wedge k * (real n \wedge (2 * ?v - k) * p n powr (2 * ?e - max-density H * k)))$
proof (rule sum-mono)
fix k
assume k: k $\in \{1..?v\}$
let ?p = p n powr (2 * ?e - max-density H * k)

have card (SIGMA S : ?I. \{T \in ?I. card (S \cap T) = k\}) = $(\sum S \in ?I. card \{T \in ?I. card (S \cap T) = k\})$ (**is** ?lhs = -)
by simp

also have ... = $(\sum S \in ?I. (?v choose k) * ((n - ?v) choose (?v - k)))$
using n k **by** (fastforce simp: card-set-of-intersecting-sets-by-card)

also have ... = $(n choose ?v) * ((?v choose k) * ((n - ?v) choose (?v - k)))$
by (auto simp: n-subsets)

also have ... $\leq n \wedge ?v * ((?v choose k) * ((n - ?v) choose (?v - k)))$
using n **by** (simp add: binomial-le-pow)

also have ... $\leq n \wedge ?v * ?v \wedge k * ((n - ?v) choose (?v - k))$
using k **by** (simp add: binomial-le-pow)

also have ... $\leq n \wedge ?v * ?v \wedge k * (n - ?v) \wedge (?v - k)$
using n-2v **by** (simp add: binomial-le-pow)

also have ... $\leq n \wedge ?v * ?v \wedge k * n \wedge (?v - k)$
by (simp add: power-mono)

also have ... = $?v \wedge k * (n \wedge (?v + (?v - k)))$
by (simp add: power-add)

also have ... = $?v \wedge k * n \wedge (2 * ?v - k)$ (**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 ... = $?v \wedge k * (real n \wedge (2 * ?v - k) * ?p)$
by simp

finally show ?lhs * ?p \leq ...

.

qed
finally have *delta-upper*: $\Delta_d \leq (\sum k = 1..?v. ?v \wedge k * (\text{real } n \wedge (2 * ?v - k))$
 $* p \ n \ \text{powr } (2 * ?e - \text{max-density } H * k))$

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

note *is-es is-psi 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 $(\lambda n. 1 / \text{prob-space-with-indicators}.\mu (MGn \ p \ n) (?I \ n) (?A \ n)) \longrightarrow 0$
proof (*rule LIMSEQ-le-zero*)
have $(\lambda n. 1 / (\text{real } n \wedge ?v * p \ n \wedge ?e)) \longrightarrow 0$
proof (*rule LIMSEQ-le-zero[OF - eventually-sequentiallyI eventually-sequentiallyI]*)
fix *n*
show $0 \leq 1 / (\text{real } n \wedge ?v * p \ n \wedge ?e)$
using *p* **by** *simp*

assume *n*: $1 \leq n$
have $1 / (\text{real } n \wedge ?v * p \ n \wedge ?e) = 1 / (\text{real } n \ \text{powr } ?v * p \ n \ \text{powr } ?e)$
using *n p-nz* **by** (*simp add: powr-realpow[symmetric]*)
also have $\dots = \text{real } n \ \text{powr } -\text{real } ?v * p \ n \ \text{powr } -\text{real } ?e$
by (*simp add: powr-minus-divide*)
also have $\dots = (\text{real } n \ \text{powr } -(?v / ?e)) \ \text{powr } ?e * (p \ n \ \text{powr } -1) \ \text{powr } ?e$
using *v-e-nz* **by** (*simp add: powr-powr*)
also have $\dots = (\text{real } n \ \text{powr } -(?v / ?e) * p \ n \ \text{powr } -1) \ \text{powr } ?e$
by (*simp add: powr-mult*)
also have $\dots = (\text{real } n \ \text{powr } -(1 / (?e / ?v)) * p \ n \ \text{powr } -1) \ \text{powr } ?e$
by *simp*
also have $\dots \leq (\text{real } n \ \text{powr } -(1 / \text{max-density } H) * p \ n \ \text{powr } -1) \ \text{powr } ?e$

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]
max-density-gr-zero[OF finite nonempty wellformed])
also have $\dots = (\text{real } n \ \text{powr } -(1 / \text{max-density } H) * (1 / p \ n \ \text{powr } 1))$
powr } ?e
by (*simp add: powr-minus-divide[symmetric]*)
also have $\dots = (\text{real } n \ \text{powr } -(1 / \text{max-density } H) / p \ n) \ \text{powr } ?e$
using *p p-nz* **by** *simp*
also have $\dots = (\text{subgraph-threshold } H \ n / p \ n) \ \text{powr } ?e$

unfolding *subgraph-threshold-def* ..
finally show $1 / (\text{real } n \wedge ?v * p n \wedge ?e) \leq (\text{subgraph-threshold } H n / p n) \text{ powr } ?e$.
next
show $(\lambda n. (\text{subgraph-threshold } H n / p n) \text{ powr } \text{real } (\text{card } (\text{uedges } H))) \longrightarrow 0$
using *p-threshold p-nz v-e-nz*
by (*auto simp: subgraph-threshold-def divide-nonneg-pos intro!: tendsto-zero-powrI*)
qed
hence $(\lambda n. ?v \wedge ?v * (1 / (\text{real } n \wedge ?v * p n \wedge ?e))) \longrightarrow \text{real } (?v \wedge ?v) * 0$
by (*rule LIMSEQ-const-mult*)
thus $(\lambda n. ?v \wedge ?v * (1 / (\text{real } n \wedge ?v * p n \wedge ?e))) \longrightarrow 0$
by simp
next
show $\forall^\infty n. 0 \leq 1 / \text{prob-space-with-indicators}.\mu (MGn p n) (?I n) (?A n)$
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 $\forall^\infty n. 1 / \text{prob-space-with-indicators}.\mu (MGn p n) (?I n) (?A n) \leq ?v \wedge ?v * (1 / (\text{real } n \wedge ?v * p n \wedge ?e))$
using facts(7) by (*rule eventually-sequentiallyI*)
qed
moreover have $(\lambda n. \text{prob-space-with-indicators}.\Delta_d (MGn p n) (?I n) (?A n)) \ll (\lambda n. (\text{prob-space-with-indicators}.\mu (MGn p n) (?I n) (?A n))^2)$
proof (*rule less-fun-bounds*)
let $?num = \lambda n k. ?v \wedge k * (\text{real } n \wedge (2 * ?v - k) * p n \text{ powr } (2 * ?e - \text{max-density } H * k))$
let $?den = \lambda n. ((1 / ?v \wedge ?v) * (\text{real } n \wedge ?v * p n \wedge ?e))^2$

— 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 $?den' = \lambda n. (1 / ?v \wedge ?v)^2 * (\text{real } n \wedge (2 * ?v) * p n \wedge (2 * ?e))$
have $den': ?den' = ?den$
by (*subst power-mult-distrib (simp add: power-mult-distrib power-even-eq)*)

have $(\lambda n. ?num n k) \ll ?den'$
proof (*rule less-fun-const-quot*)
have $(\lambda n. (\text{subgraph-threshold } H n / p n) \text{ powr } (\text{max-density } H * k)) \longrightarrow 0$
using *p-threshold mult-pos-pos[OF max-density-gr-zero[OF finite nonempty wellformed]] p-nz k*
by (*auto simp: subgraph-threshold-def divide-nonneg-pos intro!: tendsto-zero-powrI*)


```

thus ( $\lambda n. (\text{real } n^{\wedge}(2 * ?v - k) * p \text{ n powr } (2 * ?e - \text{max-density } H * k)) / (\text{real } n^{\wedge}(2 * ?v) * p \text{ n }^{\wedge}(2 * ?e))$ )  $\longrightarrow 0$ 
proof (rule LIMSEQ-cong[OF - eventually-sequentiallyI])
  fix n :: nat
  assume n:  $1 \leq n$ 
  have ( $\text{real } n^{\wedge}(2 * ?v - k) * p \text{ n powr } (2 * ?e - \text{max-density } H * k) / (\text{real } n^{\wedge}(2 * ?v) * p \text{ n }^{\wedge}(2 * ?e)) =$ 
    ( $n \text{ powr } (2 * ?v - k) * p \text{ n powr } (2 * ?e - \text{max-density } H * k)$ )
    / ( $n \text{ powr } (2 * ?v) * p \text{ n powr } (2 * ?e)$ ) (is ?lhs = -)
    using n p-nz by (simp add: powr-realpow[symmetric])
    also have ... = ( $n \text{ powr } (2 * ?v - k) / n \text{ powr } (2 * ?v)$ ) * ( $p \text{ n powr } (2 * ?e - \text{max-density } H * k) / (p \text{ n powr } (2 * ?e))$ )
    by simp
    also have ... = ( $n \text{ powr } (\text{real } (2 * ?v - k) - 2 * ?v) * p \text{ n powr } ((2 * ?e - \text{max-density } H * k) - (2 * ?e))$ )
    by (simp add: powr-diff [symmetric])
    also have ... =  $n \text{ powr } -\text{real } k * p \text{ n powr } ((2 * ?e - \text{max-density } H * k) - (2 * ?e))$ 
    apply (rule arg-cong[where y = - real k])
    using k by fastforce
    also have ... =  $n \text{ powr } -\text{real } k * p \text{ n powr } - (\text{max-density } H * k)$ 
    by simp
    also have ... = ( $n \text{ powr } -(1 / \text{max-density } H)$ )  $\text{powr } (\text{max-density } H * k) * p \text{ n powr } - (\text{max-density } H * k)$ 
    using max-density-gr-zero[OF finite nonempty wellformed] by (simp add: powr-powr)
    also have ... = ( $n \text{ powr } -(1 / \text{max-density } H)$ )  $\text{powr } (\text{max-density } H * k) * (p \text{ n powr } -1) \text{ powr } (\text{max-density } H * k)$ 
    by (simp add: powr-powr)
    also have ... = ( $n \text{ powr } -(1 / \text{max-density } H) * p \text{ n powr } -1$ )  $\text{powr } (\text{max-density } H * k)$ 
    by (simp add: powr-mult)
    also have ... = ( $n \text{ powr } -(1 / \text{max-density } H) * (1 / p \text{ n powr } 1)$ )  $\text{powr } (\text{max-density } H * k)$ 
    by (simp add: powr-minus-divide[symmetric])
    also have ... = ( $n \text{ powr } -(1 / \text{max-density } H) / p \text{ n powr } (\text{max-density } H * k)$ )
    by (simp add: p p-nz)
    also have ... = ( $\text{subgraph-threshold } H \text{ n} / p \text{ n powr } (\text{max-density } H * k)$ ) (is - = ?rhs)
    unfolding subgraph-threshold-def ..
    finally have ?lhs = ?rhs
    .
  thus ?rhs = ?lhs
  by simp
qed
next
show  $(1 / ?v^{\wedge} ?v)^{\wedge} 2 \neq 0$ 
using vpowv-inv-gr-z by auto

```

qed

hence $(\lambda n. ?num\ n\ k) \ll ?den$
by $(rule\ subst[OF\ den])$
}
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 $\forall^\infty n. ?den\ n \leq (prob-space-with-indicators.\mu\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n))^{\wedge 2}$
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 $\forall^\infty n. 0 < (prob-space-with-indicators.\mu\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n))^{\wedge 2}$
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 $(\lambda n. 1 / prob-space-with-indicators.\mu\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n) + prob-space-with-indicators.\Delta_d\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n) / (prob-space-with-indicators.\mu\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n))^{\wedge 2}) \longrightarrow 0$
by $(subst\ add-0-left[where\ a = 0, symmetric])\ (rule\ tendsto-add)$

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

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

proof $(rule\ LIMSEQ-le-zero)$

show $\forall^\infty n. 0 \leq probGn\ p\ n\ (\lambda es. \neg H \sqsubseteq edge-space.edge-ugraph\ n\ es)$

by $(rule\ eventually-sequentiallyI)\ (rule\ measure-nonneg)$

next

show $\forall^\infty n.$

$probGn\ p\ n\ (\lambda es. \neg H \sqsubseteq edge-space.edge-ugraph\ n\ es) \leq$

$1 / prob-space-with-indicators.\mu\ (MGn\ p\ n)\ (?I\ n)\ (?A\ n) +$

```

      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))2
      by (rule eventually-sequentiallyI[OF facts(4)])
    qed
  hence ( $\lambda n. 1 - \text{probGn p n} (\lambda es. \neg H \sqsubseteq \text{edge-space.edge-ugraph n es})$ )  $\longrightarrow$ 
1
  using tendsto-diff[OF tendsto-const] by fastforce
  thus prob-in-class p {G. H  $\sqsubseteq$  G}  $\longrightarrow$  1
  by (rule LIMSEQ-cong[OF - eventually-sequentiallyI[OF facts(3)]])
qed
end

```

References

- [1] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [2] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61, 1960.
- [3] S. Janson, T. Luczak, and A. Rucinski. *Random graphs*. John Wiley & Sons, 2011.
- [4] L. Noschinski. A probabilistic proof of the girth-chromatic number theorem. *Archive of Formal Proofs*, Feb. 2012. http://isa-afp.org/entries/Girth_Chromatic.shtml, Formal proof development.