Properties of Random Graphs – Subgraph Containment

Lars Hupel

May 28, 2015

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 6
   3.1 Indicator variables and valid probability values . . . . . . . . 6
   3.2 Expectation and variance . . . . . . . . . . . . . . . . . . . . 6
   3.3 Sets of indicator variables . . . . . . . . . . . . . . . . . . . 8
4 Lemmas about undirected graphs 9
   4.1 Subgraphs . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
   4.2 Induced subgraphs . . . . . . . . . . . . . . . . . . . . . . . 11
   4.3 Graph isomorphism . . . . . . . . . . . . . . . . . . . . . . . 12
   4.4 Isomorphic subgraphs . . . . . . . . . . . . . . . . . . . . . 13
   4.5 Density . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   4.6 Fixed selectors . . . . . . . . . . . . . . . . . . . . . . . . . 15
5 Classes and properties of graphs 16
6 The subgraph threshold theorem 16
1 Introduction

Random graphs have been introduced by Erdős and Rényi in [?]. 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 [?, § 11.4] and [?, § 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 [?]. 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 Limits
~/src/HOL/Probability/Probability
../Girth-Chromatic/Girth-Chromatic-Misc
begin

lemma setsum-square:
  fixes a :: 'i ⇒ 'a :: {monoid-mult, semiring-0}
  shows (∑ i ∈ I. a i) * 2 = (∑ i ∈ I. ∑ j ∈ I. a i * a j)
  ⟨proof⟩

lemma setsum-split:
  assumes finite I ⇒ (∑ i ∈ I. if p i then f i else g i) = (∑ i | i ∈ I ∧ p i. f i) + (∑ i | i ∈ I ∧ ¬p i. g i)
  ⟨proof⟩

lemma setsum-split2:
  assumes finite I
  shows (∑ i | i ∈ I ∧ P i. if Q i then f i else g i) = (∑ i | i ∈ I ∧ P i ∧ Q i. f i)
\[ i + \left( \sum_{i \mid i \in I \land P,i \land \neg Q} g \right) \]

\section*{Proofs}

\subsection*{Lemma \textbf{setsum-upper}}
\begin{proof}
\textbf{Fixes} \( f : 'i \Rightarrow 'a \cdot \{\text{ordered-cancel-ab-semigroup-add}, \text{comm-monoid-add}\} \)
\textbf{Assumes} \( \text{finite } I \land i. i \in I \implies 0 \leq f i \)
\textbf{Shows} \( \left( \sum_{i \mid i \in I \land P,i} f i \right) \leq \text{setsum } f I \)
\end{proof}

\subsection*{Lemma \textbf{setsum-lower}}
\begin{proof}
\textbf{Fixes} \( f : 'i \Rightarrow 'a \cdot \{\text{ordered-cancel-ab-semigroup-add}, \text{comm-monoid-add}\} \)
\textbf{Assumes} \( \text{finite } I \land i. i \in I \implies 0 \leq f i x < f i \)
\textbf{Shows} \( x < \text{setsum } f I \)
\end{proof}

\subsection*{Lemma \textbf{setsum-lower-or-eq}}
\begin{proof}
\textbf{Fixes} \( f : 'i \Rightarrow 'a \cdot \{\text{ordered-cancel-ab-semigroup-add}, \text{comm-monoid-add}\} \)
\textbf{Assumes} \( \text{finite } I \land i. i \in I \implies 0 \leq f i x \leq f i \)
\textbf{Shows} \( x \leq \text{setsum } f I \)
\end{proof}

\subsection*{Lemma \textbf{setsum-left-div-distrib}}
\begin{proof}
\textbf{Fixes} \( f : 'i \Rightarrow \text{real} \)
\textbf{Shows} \( \left( \sum_{i \in I} f i / x \right) = \text{setsum } f I / x \)
\end{proof}

\subsection*{Lemma \textbf{powr-mono3}}
\begin{proof}
\textbf{Fixes} \( x : \text{real} \)
\textbf{Assumes} \( 0 < x < 1 \land b \leq a \)
\textbf{Shows} \( x < a \leq x powr b \)
\end{proof}

\subsection*{Lemma \textbf{card-union}}
\begin{proof}
\textbf{Assumes} \( \text{finite } A \implies \text{finite } B \implies \text{card } (A \cup B) = \text{card } A + \text{card } B - \text{card } (A \cap B) \)
\end{proof}

\subsection*{Lemma \textbf{card-1-element}}
\begin{proof}
\textbf{Assumes} \( \text{card } E = 1 \)
\textbf{Shows} \( \exists a. E = \{a\} \)
\end{proof}

\subsection*{Lemma \textbf{card-2-elements}}
\begin{proof}
\textbf{Assumes} \( \text{card } E = 2 \)
\textbf{Shows} \( \exists a b. E = \{a, b\} \land a \neq b \)
\end{proof}

\subsection*{Lemma \textbf{bij-lift}}
\begin{proof}
\textbf{Assumes} \( \text{bij-betw } f A B \)
\textbf{Shows} \( \text{bij-betw } (\lambda e. f e) (\text{Pow } A) (\text{Pow } B) \)
\end{proof}
\[\text{lemma } \text{card-inj-subs}: \text{ inj-on } f \, A \implies B \subseteq A \implies \text{card} \, (f \, ' \, B) = \text{card} \, B\]

\[\text{lemma } \text{image-comp-cong}: (\forall a. a \in A \implies f \, a = f \, (g \, a)) \implies f \, ' \, A = f \, ' \, (g \, ' \, A)\]

\[\text{abbreviation } \text{less-fun} :: (\text{nat} \Rightarrow \text{real}) \Rightarrow (\text{nat} \Rightarrow \text{real}) \Rightarrow \text{bool} \, (\text{infix } \ll 50) \text{ where} \]

\[f \ll g \equiv (\lambda n. f \, n \div g \, n) \longrightarrow 0\]

\[\text{context}\]

\[\text{fixes } f :: \text{nat} \Rightarrow \text{real}\]

\[\text{begin}\]

\[\text{lemma } \text{LIMSEQ-power-zero}: f \longrightarrow 0 \implies 0 < n \implies (\lambda x. f \, x ^ n :: \text{real}) \longrightarrow 0\]

\[\text{lemma } \text{LIMSEQ-cong}:\]

\[\text{assumes } f \longrightarrow x \forall \infty n. f \, n = g \, n\]

\[\text{shows } g \longrightarrow x\]

\[\text{proof}\]

\[\text{print-statement } \text{Lim-transform-eventually}\]

\[\text{lemma } \text{LIMSEQ-le-zero}:\]

\[\text{assumes } g \longrightarrow 0 \forall \infty n. 0 \leq f \, n \forall \infty n. f \, n \leq g \, n\]

\[\text{shows } f \longrightarrow 0\]

\[\text{proof}\]

\[\text{lemma } \text{LIMSEQ-const-mult}:\]

\[\text{assumes } f \longrightarrow a\]

\[\text{shows } (\lambda x. c \times f \, x) \longrightarrow c \times a\]

\[\text{proof}\]

\[\text{lemma } \text{LIMSEQ-const-div}:\]

\[\text{assumes } f \longrightarrow a \, c \neq 0\]

\[\text{shows } (\lambda x. f \, x / c) \longrightarrow a / c\]

\[\text{proof}\]

\[\text{end}\]

\[\text{lemma } \text{quot-bounds}:\]

\[\text{fixes } x :: 'a :: \text{linordered-field}\]

\[\text{assumes } x \leq x' \, y' \leq y \, 0 < y \, 0 \leq x \, 0 < y'\]

\[\text{shows } x / y \leq x' / y'\]

\[\text{proof}\]

4
lemma less-fun-bounds:
  assumes $f' \ll g' \forall n. f n \leq f' n \forall n. g' n \leq g n \forall n. 0 \leq f n \forall n. 0 < g n \forall n. 0 < g' n$
  shows $f \ll g$
⟨proof⟩

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⟩

lemma partition-set-of-intersecting-sets-by-card:
  assumes finite $A$
  shows $(\{B. A \cap B \neq \{\}\} = (\bigcup n \in \{1..\text{card } A\}. \{B. \text{card } (A \cap B) = n\})$
⟨proof⟩

lemma card-set-of-intersecting-sets-by-card:
  assumes $A \subseteq I$ finite $I k \leq n \leq \text{card } I k \leq \text{card } A$
  shows $\text{card } \{B. B \subseteq I \land B = n \land \text{card } (A \cap B) = k\} = (\text{card } A \text{ choose } k) \times ((\text{card } I - \text{card } A) \text{ choose } (n - k))$
⟨proof⟩

lemma card-dep-pair-set:
  assumes finite $A$ \land $a \subseteq A \implies \text{finite } (f a)$
  shows $\text{card } \{(a, b). a \subseteq A \land \text{card } a = n \land b \subseteq f a \land \text{card } b = g a\} = (\sum a \mid a \subseteq A \land \text{card } a = n. \text{card } (f a) \text{ choose } g a) \text{ (is card } ?S = ?C)$
⟨proof⟩

lemma setprod-cancel-nat:
  — Contributed by Manuel Eberl
  fixes $f : 'a \Rightarrow \text{nat}$
  assumes $B \subseteq A \text{ and } \text{finite } A \text{ and } \forall x \in B. f x \neq 0$
  shows $\prod A \div \prod B = \prod (A - B) \text{ (is } ?A / ?B = ?C)$
⟨proof⟩

lemma setprod-id-cancel-nat:
  — Contributed by Manuel Eberl
  fixes $X : \text{nat set}$
  assumes $B \subseteq A \text{ and } \text{finite } A \text{ and } 0 \notin B$
  shows $\prod A \div \prod B = \prod (A - B)$
⟨proof⟩

lemma (in prob-space) integrable-squareD:
  — Contributed by Johannes Hlzl
  fixes $X : \rightarrow \text{real}$
  assumes integrable $M (\lambda x. (X x)^2) X \in \text{borel-measurable } M$
  shows integrable $M X$
⟨proof⟩
3 Lemmas about probabilities

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

3.1 Indicator variables and valid probability values

abbreviation rind :: 'a set ⇒ 'a ⇒ real where
rind ≡ indicator

lemma product-indicator:
\( rind A x \ast rind B x = rind (A \cap B) x \)
⟨proof⟩

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

abbreviation valid-prob (p :: real) ≡ 0 ≤ p ∧ p ≤ 1
abbreviation nonzero-prob (p :: real) ≡ 0 < p ∧ p < 1

A function 'a ⇒ real is a ‘valid probability function’ iff each value in the image is valid, and similarly for ‘nonzero’.

abbreviation valid-prob-fun f ≡ (\(\forall n\). valid-prob (f n))
abbreviation nonzero-prob-fun f ≡ (\(\forall n\). nonzero-prob (f n))

lemma nonzero-fun-is-valid-fun: nonzero-prob-fun f ⇒ valid-prob-fun f
⟨proof⟩

3.2 Expectation and variance

context prob-space
begin

Note that there is already a notion of independent sets (see indep-set), but we use the following – simpler – definition:

definition indep A B ←→ prob (A \cap B) = prob A \ast prob B

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

lemma expectation-indicator:
\( A \in \text{events} \implies \text{expectation} (\text{rind} A) = \text{prob} A \)
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** $\forall a. 0 \leq X a$ and integrable $M X 0 < t$
- **Shows** $\Pr \{ a \in \text{space } M. t \leq X a \} \leq \text{expectation } X / t$

**Proof**

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

**Lemma (Variance-Expectation):**
- **Fixes** $X :: \lambda x. (X x) \rightarrow \text{real}$
- **Assumes** integrable $M (\lambda x. (X x)^2)$ and $X \in$ 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**

A corollary from the Markov inequality is Chebyshev’s inequality, which gives an upper bound for the deviation of a random variable from its expectation:

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

**Lemma (Chebyshev-Inequality):**
- **Fixes** $Y :: \lambda y. (Y y)^2$
- **Assumes** $Y$-int: integrable $M (\lambda y. (Y y)^2)$
- **Assumes** $Y$-borel: $Y \in$ borel-measurable $M$
- **Fixes** $s :: \text{real}$
- **Assumes** $s$-pos: $0 < s$
- **Shows** $\Pr \{ a \in \text{space } M. s \leq |Y a - \text{expectation } Y| \} \leq \text{variance } Y / s^2$

**Proof**

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

**Corollary (Chebyshev-Prob-Zero):**
- **Fixes** $Y :: \lambda y. (Y y)^2$
- **Assumes** $Y$-int: integrable $M (\lambda y. (Y y)^2)$
- **Assumes** $Y$-borel: $Y \in$ borel-measurable $M$
- **Assumes** $\mu$-pos: expectation $Y > 0$
- **Shows** $\Pr \{ a \in \text{space } M. Y a = 0 \} \leq \text{expectation } (\lambda y. (Y y)^2) / \text{expectation } Y^2 - 1$

**Proof**

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 ⇒ 'a set
  assumes A: A 'I ⊆ events

  assumes prob-non-zero: ∃ i ∈ I. 0 < prob (A i)

begin

We call the underlying sets A i for each i ∈ I, and the corresponding indicator variables X i. The sum is denoted by Y, and its expectation by μ.

definition X i = rind (A i)
definition Y x = (∑ i ∈ I. X i x)

definition μ = expectation Y

In the lecture notes, the following two relations are called ∼ and ⊸, respectively. Note that they are not the opposite of each other.

abbreviation ineq-indep :: 'i ⇒ 'i ⇒ bool where
  ineq-indep i j ≡ (i ≠ j ∧ indep (A i) (A j))

abbreviation ineq-dep :: 'i ⇒ 'i ⇒ bool where
  ineq-dep i j ≡ (i ≠ j ∧ ¬ indep (A i) (A j))

definition ∆ a = (∑ i ∈ I. ∑ j | j ∈ I ∧ i ≠ j. prob (A i ∩ A j))
definition ∆ d = (∑ i ∈ I. ∑ j | j ∈ I ∧ ineq-dep i j. prob (A i ∩ A j))

lemma ∆-zero:
  assumes i j. i ∈ I ⇒ j ∈ I ⇒ i ≠ j ⇒ indep (A i) (A j)
  shows ∆ d = 0
⟨proof⟩

lemma A-events[measurable]: i ∈ I ⇒ A i ∈ events
⟨proof⟩

lemma expectation-X-Y: μ = (∑ i∈I. expectation (X i))
⟨proof⟩

lemma expectation-X-non-zero: ∃ i ∈ I. 0 < expectation (X i)
⟨proof⟩

corollary μ-non-zero[simp]: 0 < μ
lemma \(\Delta_d\)-nonneg: \(0 \leq \Delta_d\)

〈proof〉

〈proof〉

corollary \(\mu\)-sq-non-zero[simp]: \(0 < \mu \cdot 2\)

〈proof〉

lemma \(Y\)-square-unfold: \((\lambda x. (Y \cdot x)^2) = (\lambda x. \sum_{i \in I} \sum_{j \in I} rind (A_i \cap A_j) \cdot x)\)

〈proof〉

lemma integrable-\(Y\)-sq[simp]: integrable \(M\) \((\lambda y. (Y \cdot y)^2)\)

〈proof〉

lemma measurable-\(Y\)[measurable]: \(Y \in \text{borel-measurable} \ M\)

〈proof〉

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

〈proof〉

lemma \(\Delta\)-expectation-\(X\): \(\Delta_a \leq \mu \cdot 2 + \Delta_d\)

〈proof〉

lemma prob-\(\mu\)-\(\Delta_d\): \(\text{prob} \{ a \in \text{space} \ M. Y \cdot a = 0 \} \leq 1 / (\mu + \Delta_d / \mu \cdot 2 - 1\)

〈proof〉

lemma prob-\(\mu\)-\(\Delta_d\): \(\text{prob} \{ a \in \text{space} \ M. Y \cdot a = 0 \} \leq 1/\mu + \Delta_d/\mu \cdot 2\)

〈proof〉

〈proof〉

end

end

4 Lemmas about undirected graphs

theory Ugraph-Lemmas

imports

Prob-Lemmas

../Girth-Chromatic/Girth-Chromatic

Lattices-Big

begin

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

definition complete :: nat set \Rightarrow ugraph where

complete \(V = (V, \text{all-edges} \ V)\)

lemma complete-wellformed: uwellformed \(\text{complete} \ V)\)

4 Lemmas about undirected graphs
If the set of vertices is finite, the set of edges in the complete graph is finite.

**Lemma** `all-edges-finite`: \( \text{finite } V \implies \text{finite } (\text{all-edges } V) \)

**Proof**

**Corollary** `complete-finite-edges`: \( \text{finite } V \implies \text{finite } (\text{uedges } (\text{complete } V)) \)

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

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

**Proof**

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

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

The complete graph is finite.

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

**Proof**

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

**Definition** `nonempty-graph G ≡ \text{uverts } G \neq \{\} \land \text{uedges } G \neq \{\}`

A random graph is both wellformed and finite.

**Lemma** `wellformed-and-finite`:
- Assumes \( E \in \text{Pow } S\text{-edges} \)
- Shows \( \text{finite-graph } (\text{edge-ugraph } E) \land \text{wellformed } (\text{edge-ugraph } E) \)

**Proof**

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

**Lemma** `cylinder-empty-prob`:
- \( A \subseteq S\text{-edges} \implies \text{prob } (\text{cylinder } S\text{-edges } A \{\}) = p^{-\text{card } A} \)

**Proof**

4.1 Subgraphs

**Definition** `subgraph :: ugraph ⇒ ugraph ⇒ bool` where
- \( \text{subgraph } G' G ≡ \text{uverts } G' \subseteq \text{uverts } G \land \text{uedges } G' \subseteq \text{uedges } G \)

**Lemma** `subgraph-refl`: \( \text{subgraph } G G \)

**Proof**

**Lemma** `subgraph-trans`: \( \text{subgraph } G'' G' \implies \text{subgraph } G' G \implies \text{subgraph } G'' G \)

**Proof**

**Lemma** `subgraph-antisym`: \( \text{subgraph } G G' \implies \text{subgraph } G' G \implies G = G' \)
lemma subgraph-complete:
  assumes uwellformed G
  shows subgraph G (complete (uverts G))
⟨proof⟩

corollary wellformed-all-edges: uwellformed G ⇒ uedges G ⊆ all-edges (uverts G)
⟨proof⟩

lemma subgraph-finite: [ finite-graph G; subgraph G' G ] ⇒ finite-graph G'
⟨proof⟩

corollary wellformed-finite:
  assumes finite (uverts G) and uwellformed G
  shows finite-graph G
⟨proof⟩

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

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

lemma subgraphs-finite:
  assumes finite-graph G
  shows finite (subgraphs G)
⟨proof⟩

corollary nonempty-subgraphs-finite: finite-graph G ⇒ finite (nonempty-subgraphs G)
⟨proof⟩

4.2 Induced subgraphs

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

lemma induced-is-subgraph:
  V ⊆ uverts G ⇒ subgraph (induced-subgraph V G) G
  V ⊆ uverts G ⇒ subgraph (induced-subgraph V G) (complete V)
⟨proof⟩

lemma induced-wellformed: uwellformed G ⇒ V ⊆ uverts G ⇒ uwellformed (induced-subgraph V G)
⟨proof⟩
**Lemma** *subgraph-union-induced:*

**Assumes** uverts $H_1 \subseteq S$ and uverts $H_2 \subseteq T$

**Assumes** uwellformed $H_1$ and uwellformed $H_2$

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

⟨proof⟩

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

**Assumes** uverts $H \subseteq V$ and uwellformed $H$ and $V \subseteq S$-verts

**Shows** $\text{prob } \{es \in \text{space } P. \text{subgraph } H \text{ (induced-subgraph } V \ (\text{edge-ugraph } es))\} = p \ ^{\text{card (uedges } H)} \ (\text{is prob } ? A = -)$

⟨proof⟩

### 4.3 Graph isomorphism

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

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

**Fun** $\text{map-ugraph :: (} \text{nat} \Rightarrow \text{nat} \Rightarrow \text{ugraph } \Rightarrow \text{ugraph}\text{ where} \text{map-ugraph } f \ (V, E) = (f \ ^{\prime} \ V, (\lambda e. f \ ^{\prime} \ e) \ ^{\prime} \ E)$

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

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

**Lemma** map-ugraph-id: $\text{map-ugraph } id = id$

⟨proof⟩

**Lemma** map-ugraph-trans: $\text{map-ugraph } (g \circ f) = (\text{map-ugraph } g) \circ (\text{map-ugraph } f)$

⟨proof⟩

**Lemma** map-ugraph-wellformed:

**Assumes** uwellformed $G$ and inj-on $f$ (uverts $G$)

**Shows** uwellformed $\ (\text{map-ugraph } f \ G)$

⟨proof⟩

**Lemma** map-ugraph-finite: finite-graph $G \implies \text{finite-graph } (\text{map-ugraph } f \ G)$
lemma map-ugraph-preserves-sub:
   assumes subgraph G₁ G₂
   shows subgraph (map-ugraph f G₁) (map-ugraph f G₂)
⟨proof⟩

lemma isomorphic-refl: uwellformed G ⇒ G ≃ G
⟨proof⟩

lemma isomorphic-trans:
   assumes G₁ ≃ G₂ and G₂ ≃ G₃
   shows G₁ ≃ G₃
⟨proof⟩

lemma isomorphic-sym:
   assumes G₁ ≃ G₂
   shows G₂ ≃ G₁
⟨proof⟩

lemma isomorphic-cards:
   assumes G₁ ≃ G₂
   shows card (uverts G₁) = card (uverts G₂) (is ?V)
   card (uedges G₁) = card (uedges G₂) (is ?E)
⟨proof⟩

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
to contain a triangle as an isomorphic subgraph.

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

lemma subgraph-is-subgraph-isomorphic: [ uwellformed G'; uwellformed G; sub-
   graph G' G ] ⇒⇒ G' ⊑ G
⟨proof⟩

lemma isomorphic-is-subgraph-isomorphic: G₁ ≃ G₂ ⇒⇒ G₁ ⊑ G₂
⟨proof⟩

lemma subgraph-isomorphic-refl: uwellformed G ⇒ G ⊑ G
⟨proof⟩

lemma subgraph-isomorphic-pre-iso-closed:
assumes $G_1 \simeq G_2$ and $G_2 \subseteq G_3$
shows $G_1 \subseteq G_3$
(proof)

lemma subgraph-isomorphic-pre-subgraph-closed:
assumes uwellformed $G_1$ and subgraph $G_1$ $G_2$ and $G_2 \subseteq G_3$
shows $G_1 \subseteq G_3$
(proof)


lemma subgraph-isomorphic-trans[trans]:
assumes $G_1 \subseteq G_2$ and $G_2 \subseteq G_3$
shows $G_1 \subseteq G_3$
(proof)

lemma subgraph-isomorphic-post-iso-closed: $[ H \subseteq G; G \simeq G' ] \implies H \subseteq G'$
(proof)


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 $G = \text{Lattices-Big.Max}(\text{density \{ nonempty-subgraphs } G)$

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 linorder-class.Max require non-emptiness of the target set, but we need that anyway to get a value out of it.

lemma subgraph-has-max-density:
assumes finite-graph $G$ and nonempty-graph $G$ and uwellformed $G$
shows $\exists G', \text{density } G' = \text{max-density } G \land \text{subgraph } G' G \land \text{nonempty-graph } G' \land \text{finite-graph } G' \land \text{uwellformed } G'$
(proof)
lemma max-density-is-max:
  assumes finite-graph G and finite-graph G' and nonempty-graph G' and uwell-formed G' and subgraph G' G
  shows density G' ≤ max-density G
  ⟨proof⟩

lemma max-density-gr-zero:
  assumes finite-graph G and nonempty-graph G and uwellformed G
  shows 0 < max-density G
  ⟨proof⟩

lemma isomorphic-density:
  assumes G₁ ≃ G₂
  shows density G₁ = density G₂
  ⟨proof⟩

lemma isomorphic-max-density:
  assumes G₁ ≃ G₂ and nonempty-graph G₁ and nonempty-graph G₂ and finite-graph G₁ and finite-graph G₂
  shows max-density G₁ = max-density G₂
  ⟨proof⟩

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.

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

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

lemma ex-fixed-selector:
  assumes uwellformed H and finite-graph H
  obtains f where is-fixed-selector H f
  ⟨proof⟩

lemma fixed-selector-induced-subgraph:
  assumes is-fixed-selector H f and card (uverts H) = card V and finite V
  assumes sub: subgraph (f V) (induced-subgraph V G) and V: V ⊆ uverts G and G: uwellformed G
  shows H ⊑ G

\(^1\)We call such a selector fixed because its result is deterministic.
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 ⇒ bool where
ugraph-property C ≡ ∀ G ∈ C. ∀ G′. G ≃ G′ → G′ ∈ C

abbreviation prob-in-class :: (nat ⇒ real) ⇒ ugraph-class ⇒ nat ⇒ real where
prob-in-class p c n ≡ probGn p n (λes. edge-space.edge-ugraph n es ∈ 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 ⇒ (nat ⇒ real) ⇒ bool where
is-threshold c t ≡ ugraph-property c ∧ (∀ p. nonzero-prob-fun p →
(p ≪ t → prob-in-class p c →→ 0) ∧
(t ≪ p → prob-in-class p c →→ 1))

lemma is-thresholdI[intro]:
  assumes ugraph-property c
  assumes ∨ p. [ nonzero-prob-fun p; p ≪ t ] → prob-in-class p c →→ 0
  assumes ∨ p. [ nonzero-prob-fun p; t ≪ p ] → prob-in-class p c →→ 1
  shows is-threshold c t
⟨proof⟩

end

6 The subgraph threshold theorem

theory Subgraph-Threshold
imports
  Ugraph-Properties
begin

lemma (in edge-space) measurable-pred[measurable]: Measurable.pred P Q
 ⟨proof⟩

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

The corresponding threshold function is

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

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

definition subgraph-threshold :: ugraph ⇒ nat ⇒ real where
subgraph-threshold \( H \) \( n \) = \( n \) powr \( -\left(\frac{1}{\max\text{-}\text{density } H}\right) \)

theorem
  assumes nonempty: nonempty-graph \( H \) and finite: finite-graph \( H \) and well-formed: uwellformed \( H \)
  shows is-threshold \( \{ G. \ H \subseteq G \} \) (subgraph-threshold \( H \))
 ⟨proof⟩

end