Roth's Theorem on Arithmetic Progressions

Chelsea Edmonds, Angeliki Koutsoukou-Argyraki and Lawrence C. Paulson
Computer Laboratory, University of Cambridge CB3 0FD
{cle47,ak2110,lp15}@cam.ac.uk

January 3, 2022

Abstract

We formalise a proof of Roth's Theorem on Arithmetic Progressions, a major result in additive combinatorics on the existence of 3-term arithmetic progressions in subsets of natural numbers. To this end, we follow a proof using graph regularity. We employ our recent formalisation of Szemerédi's Regularity Lemma, a major result in extremal graph theory, which we use here to prove the Triangle Counting Lemma and the Triangle Removal Lemma. Our sources are Yufei Zhao's MIT lecture notes "Graph Theory and Additive Combinatorics"[1] and W.T. Gowers's Cambridge lecture notes "Topics in Combinatorics".[2] We also refer to the University of Georgia notes by Stephanie Bell and Will Grodzicki "Using Szemerédi's Regularity Lemma to Prove Roth's Theorem".[3]

Contents

Acknowledgements

The authors were supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council.

[1] https://ocw.mit.edu/courses/mathematics/18-217-graph-theory-and-additive-combinatorics-fall-2019/lecture-notes/MIT18_217F19_ch3.pdf and https://yufeizhao.com/gtac/gtac17.pdf
[2] https://www.dpmms.cam.ac.uk/~par31/notes/tic.pdf
[3] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.432.327

1 Roth's Theorem on Arithmetic Progressions

theory *Roth-Arithmetic-Progressions*
imports *Szemeredi-Regularity.Szemeredi*
Random-Graph-Subgraph-Threshold.Subgraph-Threshold
Ergodic-Theory.Asymptotic-Density
HOL−Library.Ramsey HOL−Library.Nat-Bijection

begin

1.1 For the Library

declare *prod-encode-eq* [*simp*]
declare *prod-decode-eq* [*simp*]

lemma *mult-mod-cancel-right*:
fixes m :: *'a*::{*euclidean-ring-cancel*,*semiring-gcd*}
assumes *eq*: $(a * n)$ *mod* $m = (b * n)$ *mod* m **and** *coprime* m n
shows a *mod* $m = b$ *mod* m
⟨*proof*⟩

lemma *mult-mod-cancel-left*:
fixes m :: *'a*::{*euclidean-ring-cancel*,*semiring-gcd*}
assumes $(n * a)$ *mod* $m = (n * b)$ *mod* m **and** *coprime* m n
shows a *mod* $m = b$ *mod* m
⟨*proof*⟩

lemma *edge-density-le1*: *edge-density* X Y $G \leq 1$
⟨*proof*⟩

lemma *card-3-iff*: *card* $S = 3 \longleftrightarrow (\exists x\ y\ z.\ S = \{x,y,z\} \wedge x \neq y \wedge y \neq z \wedge x \neq z)$
⟨*proof*⟩

1.2 Miscellaneous Preliminaries

lemma *sum-prod-le-prod-sum*:
fixes a :: *'a* ⇒ *'b*::*linordered-idom*
assumes $\bigwedge i.\ i \in I \Longrightarrow a\ i \geq 0 \wedge b\ i \geq 0$
shows $(\sum i{\in}I.\ \sum j{\in}I.\ a\ i * b\ j) \leq (\sum i{\in}I.\ a\ i) * (\sum i{\in}I.\ b\ i)$
⟨*proof*⟩

lemma *real-mult-gt-cube*: $A \geq (X$ ::*real*$) \Longrightarrow B \geq X \Longrightarrow C \geq X \Longrightarrow X \geq 0 \Longrightarrow A * B * C \geq X\hat{}3$
⟨*proof*⟩

lemma *min-card-fin-X-elem*: *finite* $X \Longrightarrow x \in X \Longrightarrow$ *card* $X \geq 1$
⟨*proof*⟩

lemma *card-or-filter-max*:
assumes *finite A*
shows *card* $\{a \in A \,.\, P\ a \vee Q\ a\} \leq$ *card* $\{a \in A \,.\, P\ a\} +$ *card* $\{a \in A \,.\, Q\ a\}$
⟨*proof*⟩

lemma *triple-sigma-rewrite-card*:
assumes *finite X finite Y finite Z*
shows *card* $\{(x,\ y,\ z) \,.\, x \in X \wedge (y,\ z) \in Y \times Z \wedge P\ x\ y\ z\} = (\sum x \in X \,.\,$ *card* $\{(y,z) \in Y \times Z.\ P\ x\ y\ z\})$
⟨*proof*⟩

lemma *all-edges-between-Union1*:
all-edges-between (*Union* $\mathcal{X}$) $Y\ G = (\bigcup X \in \mathcal{X}.$ *all-edges-between* $X\ Y\ G)$
⟨*proof*⟩

lemma *all-edges-between-Union2*:
all-edges-between X (*Union* $\mathcal{Y}$) $G = (\bigcup Y \in \mathcal{Y}.$ *all-edges-between* $X\ Y\ G)$
⟨*proof*⟩

lemma *all-edges-between-disjoint1*:
assumes *disjoint R*
shows *disjoint* $((\lambda X.$ *all-edges-between* $X\ Y\ G)$ ‘ $R)$
⟨*proof*⟩

lemma *all-edges-between-disjoint2*:
assumes *disjoint R*
shows *disjoint* $((\lambda Y.$ *all-edges-between* $X\ Y\ G)$ ‘ $R)$
⟨*proof*⟩

lemma *all-edges-between-disjoint-family-on1*:
assumes *disjoint R*
shows *disjoint-family-on* $(\lambda X.$ *all-edges-between* $X\ Y\ G)\ R$
⟨*proof*⟩

lemma *all-edges-between-disjoint-family-on2*:
assumes *disjoint R*
shows *disjoint-family-on* $(\lambda Y.$ *all-edges-between* $X\ Y\ G)\ R$
⟨*proof*⟩

lemma *all-edges-between-mono1*:
$Y \subseteq Z \Longrightarrow$ *all-edges-between* $Y\ X\ G \subseteq$ *all-edges-between* $Z\ X\ G$
⟨*proof*⟩

lemma *all-edges-between-mono2*:
$Y \subseteq Z \Longrightarrow$ *all-edges-between* $X\ Y\ G \subseteq$ *all-edges-between* $X\ Z\ G$
⟨*proof*⟩

lemma *inj-on-mk-uedge*: $X \cap Y = \{\} \Longrightarrow$ *inj-on mk-uedge* (*all-edges-between* $X\ Y\ G$)

⟨*proof*⟩

lemma *uwellformed-alt*:
assumes *uwellformed G* {*x*, *y*} ∈ *uedges G*
shows {*x*, *y*} ⊆ *uverts G*
⟨*proof*⟩

lemma *uwellformed-alt-fst*:
assumes *uwellformed G* {*x*, *y*} ∈ *uedges G*
shows *x* ∈ *uverts G*
⟨*proof*⟩

lemma *uwellformed-alt-snd*:
assumes *uwellformed G* {*x*, *y*} ∈ *uedges G*
shows *y* ∈ *uverts G*
⟨*proof*⟩

lemma *all-edges-between-subset-times*: *all-edges-between X Y G* ⊆ (*X* ∩ ⋃(*uedges G*)) × (*Y* ∩ ⋃(*uedges G*))
⟨*proof*⟩

lemma *finite-all-edges-between′*:
assumes *finite* (*uverts G*) *uwellformed G*
shows *finite* (*all-edges-between X Y G*)
⟨*proof*⟩

lemma *card-all-edges-between*:
assumes *finite Y finite* (*uverts G*) *uwellformed G*
shows *card* (*all-edges-between X Y G*) = (∑ *y*∈*Y*. *card* (*all-edges-between X* {*y*} *G*))
⟨*proof*⟩

lemma *max-edges-graph*:
assumes *uwellformed G finite* (*uverts G*)
shows *card* (*uedges G*) ≤ (*card* (*uverts G*))^2
⟨*proof*⟩

lemma *all-edges-between-ss-uedges*: *mk-uedge* ‘ (*all-edges-between X Y G*) ⊆ *uedges G*
⟨*proof*⟩

lemma *all-edges-betw-D1*: (*x*, *y*) ∈ *all-edges-between X Y G* ⟹ *x* ∈ *X*
⟨*proof*⟩

lemma *all-edges-betw-D2*: (*x*, *y*) ∈ *all-edges-between X Y G* ⟹ *y* ∈ *Y*
⟨*proof*⟩

lemma *all-edges-betw-D3*: (*x*, *y*) ∈ *all-edges-between X Y G* ⟹ {*x*, *y*} ∈ *uedges G*

⟨*proof*⟩

lemma *all-edges-betw-I*: $x \in X \Longrightarrow y \in Y \Longrightarrow \{x, y\} \in uedges\ G \Longrightarrow (x, y) \in$ *all-edges-between* $X\ Y\ G$
⟨*proof*⟩

lemma *all-edges-between-E-diff*:
all-edges-between $X\ Y\ (V,E-E') =$ *all-edges-between* $X\ Y\ (V,E) -$ *all-edges-between* $X\ Y\ (V,E')$
⟨*proof*⟩

lemma *all-edges-between-E-Un*:
all-edges-between $X\ Y\ (V,E \cup E') =$ *all-edges-between* $X\ Y\ (V,E) \cup$ *all-edges-between* $X\ Y\ (V,E')$
⟨*proof*⟩

lemma *all-edges-between-E-UN*:
all-edges-between $X\ Y\ (V, \bigcup i \in I.\ E\ i) = (\bigcup i \in I.$ *all-edges-between* $X\ Y\ (V,E\ i))$
⟨*proof*⟩

lemma *all-edges-betw-prod-def*: *all-edges-between* $X\ Y\ G = \{(x, y) \in X \times Y\ .\ \{x, y\} \in uedges\ G\}$
⟨*proof*⟩

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

lemma *all-edges-preserved*: $[\![$*all-edges-between* $A\ B\ G' =$ *all-edges-between* $A\ B\ G$; $X \subseteq A$; $Y \subseteq B]\!]$
$\Longrightarrow$ *all-edges-between* $X\ Y\ G' =$ *all-edges-between* $X\ Y\ G$
⟨*proof*⟩

lemma *subgraph-edge-wf*:
assumes *uwellformed G uverts H* = *uverts G uedges H* $\subseteq$ *uedges G*
shows *uwellformed H*
⟨*proof*⟩

1.3 Preliminaries on Neighbors in Graphs

definition *neighbor-in-graph*:: *uvert* ⇒ *uvert* ⇒ *ugraph* ⇒ *bool*
where *neighbor-in-graph x y G* ≡ ($x \in$ (*uverts G*) ∧ $y \in$ (*uverts G*) ∧ $\{x,y\} \in$ (*uedges G*))

definition *neighbors* :: *uvert* ⇒ *ugraph* ⇒ *uvert set* **where**
neighbors x G ≡ $\{y \in$ *uverts G* . *neighbor-in-graph x y G*$\}$

definition *neighbors-ss*:: *uvert* ⇒ *uvert set* ⇒ *ugraph* ⇒ *uvert set* **where**

neighbors-ss x Y G ≡ {*y* ∈ *Y* . *neighbor-in-graph x y G*}

lemma *all-edges-betw-prod-def-neighbors*: *uwellformed G* ⟹
all-edges-between X Y G = {(*x*, *y*) ∈ *X* × *Y* . *neighbor-in-graph x y G*}
⟨*proof*⟩

lemma *all-edges-betw-sigma-neighbor*:
uwellformed G ⟹ *all-edges-between X Y G* = (*SIGMA x*:*X*. *neighbors-ss x Y G*)
⟨*proof*⟩

lemma *card-all-edges-betw-neighbor*:
assumes *finite X finite Y uwellformed G*
shows *card* (*all-edges-between X Y G*) = (∑ *x*∈*X*. *card* (*neighbors-ss x Y G*))
⟨*proof*⟩

1.4 Preliminaries on Triangles in Graphs

definition *triangle-in-graph*:: *uvert* ⇒ *uvert* ⇒ *uvert* ⇒ *ugraph* ⇒ *bool*
where *triangle-in-graph x y z G*
≡ ({*x*,*y*} ∈ *uedges G*) ∧ ({*y*,*z*} ∈ *uedges G*) ∧ ({*x*,*z*} ∈ *uedges G*)

definition *triangle-triples*
where *triangle-triples X Y Z G* ≡ {(*x*,*y*,*z*) ∈ *X* × *Y* × *Z*. *triangle-in-graph x y z G*}

lemma *card-triangle-triples-rotate*: *card* (*triangle-triples X Y Z G*) = *card* (*triangle-triples Y Z X G*)
⟨*proof*⟩

lemma *triangle-commu1*:
assumes *triangle-in-graph x y z G*
shows *triangle-in-graph y x z G*
⟨*proof*⟩

lemma *triangle-vertices-distinct1*:
assumes *wf*: *uwellformed G*
assumes *tri*: *triangle-in-graph x y z G*
shows *x* ≠ *y*
⟨*proof*⟩

lemma *triangle-vertices-distinct2*:
assumes *uwellformed G triangle-in-graph x y z G*
shows *y* ≠ *z*
⟨*proof*⟩

lemma *triangle-vertices-distinct3*:
assumes *uwellformed G triangle-in-graph x y z G*
shows *z* ≠ *x*

⟨*proof*⟩

lemma *triangle-in-graph-edge-point*:
assumes *uwellformed G*
shows *triangle-in-graph x y z G* ⟷ $\{y, z\} \in$ *uedges G* ∧ *neighbor-in-graph x y G* ∧ *neighbor-in-graph x z G*
⟨*proof*⟩

definition
unique-triangles G
≡ ∀ *e* ∈ *uedges G*. ∃!*T*. ∃ *x y z*. *T* = {*x*,*y*,*z*} ∧ *triangle-in-graph x y z G* ∧ *e* ⊆ *T*

definition *triangle-free-graph*:: *ugraph* ⇒ *bool*
where *triangle-free-graph G* ≡ ¬(∃ *x y z*. *triangle-in-graph x y z G*)

lemma *triangle-free-graph-empty*: *uedges G* = {} ⟹ *triangle-free-graph G*
⟨*proof*⟩

lemma *edge-vertices-not-equal*:
assumes *uwellformed G* {*x*,*y*} ∈ *uedges G*
shows *x* ≠ *y*
⟨*proof*⟩

lemma *edge-btw-vertices-not-equal*:
assumes *uwellformed G* (*x*, *y*) ∈ *all-edges-between X Y G*
shows *x* ≠ *y*
⟨*proof*⟩

lemma *mk-triangle-from-ss-edges*:
assumes (*x*, *y*) ∈ *all-edges-between X Y G* **and** (*x*, *z*) ∈ *all-edges-between X Z G* **and** (*y*, *z*) ∈ *all-edges-between Y Z G*
shows (*triangle-in-graph x y z G*)
⟨*proof*⟩

lemma *triangle-in-graph-verts*:
assumes *uwellformed G*
assumes *triangle-in-graph x y z G*
shows *x* ∈ *uverts G y* ∈ *uverts G z*∈ *uverts G*
⟨*proof*⟩

definition *triangle-set* :: *ugraph* ⇒ *uvert set set*
where *triangle-set G* ≡ { {*x*,*y*,*z*} | *x y z*. *triangle-in-graph x y z G*}

fun *mk-triangle-set* :: (*uvert* × *uvert* × *uvert*) ⇒ *uvert set*
where *mk-triangle-set* (*x*, *y*, *z*) = {*x*,*y*,*z*}

lemma *convert-triangle-rep-ss*:
fixes G :: *ugraph*
assumes $X \subseteq$ *uverts* G **and** $Y \subseteq$ *uverts* G **and** $Z \subseteq$ *uverts* G
shows *mk-triangle-set* ‘ $\{(x, y, z) \in X \times Y \times Z$. (*triangle-in-graph* x y z $G)\}$ $\subseteq$ *triangle-set* G
⟨*proof*⟩

lemma *finite-triangle-set*:
fixes G :: *ugraph*
assumes *fin*: *finite* (*uverts* G) **and** *wf*: *uwellformed* G
shows *finite* (*triangle-set* G)
⟨*proof*⟩

lemma *card-triangle-3*:
fixes G :: *ugraph*
assumes $t \in$ *triangle-set* G *uwellformed* G
shows *card* $t = 3$
⟨*proof*⟩

lemma *triangle-set-power-set-ss*: *uwellformed* $G \Longrightarrow$ *triangle-set* $G \subseteq$ *Pow* (*uverts* G)
⟨*proof*⟩

lemma *triangle-set-finite*:
assumes *finite* (*uverts* G)
assumes *uwellformed* G
shows *finite* (*triangle-set* G)
⟨*proof*⟩

lemma *triangle-in-graph-ss*:
fixes G :: *ugraph* **and** *Gnew* :: *ugraph*
assumes *uedges Gnew* $\subseteq$ *uedges* G
assumes *triangle-in-graph* x y z *Gnew*
shows *triangle-in-graph* x y z G
⟨*proof*⟩

lemma *triangle-set-graph-edge-ss*:
fixes G :: *ugraph* **and** *Gnew* :: *ugraph*
assumes *uwellformed* G
assumes *uedges Gnew* $\subseteq$ *uedges* G
assumes *uverts Gnew* $=$ *uverts* G
shows (*triangle-set Gnew*) $\subseteq$ (*triangle-set* G)
⟨*proof*⟩

lemma *triangle-set-graph-edge-ss-bound*:
fixes G :: *ugraph* **and** *Gnew* :: *ugraph*
assumes *uwellformed* G
assumes *finite* (*uverts* G)
assumes *uedges Gnew* $\subseteq$ *uedges* G

assumes *uverts Gnew = uverts G*
shows *card (triangle-set G) ≥ card (triangle-set Gnew)*
⟨*proof*⟩

1.5 The Triangle Counting Lemma and the Triangle Removal Lemma

We begin with some more auxiliary material to be used in the main lemmas.

lemma *regular-pairI*:
fixes *ε :: real* **and** *G :: ugraph* **and** *X :: uvert set* **and** *Y ::uvert set*
assumes *ε > 0* **and** *regular-pair X Y G ε* **and** *xss*: *X' ⊆ X* **and** *yss*: *Y' ⊆ Y* **and** *card X' ≥ ε * card X* **and** *(card Y' ≥ ε * card Y)*
shows *| edge-density X' Y' G − edge-density X Y G | ≤ ε*
⟨*proof*⟩

lemma *edge-density-zero*: *Y = {} ⟹ edge-density X Y G = 0*
⟨*proof*⟩

lemma *regular-pair-neighbor-bound*:
fixes *ε::real*
assumes *finG*: *finite (uverts G)*
assumes *xss*: *X ⊆ uverts G* **and** *yss*: *Y ⊆ uverts G* **and** *card X > 0*
and *wf*: *uwellformed G*
and *eg0*: *ε > 0* **and** *regular-pair X Y G ε* **and** *ed*: *edge-density X Y G ≥ 2*ε*
shows *card{x ∈ X. card (neighbors-ss x Y G) < (edge-density X Y G − ε)* card (Y)} < ε * card X*
(**is** *card (?X') < ε * -*)
⟨*proof*⟩

lemma *neighbor-set-meets-e-reg-cond*:
fixes *ε::real*
assumes *X ⊆ uverts G* **and** *Y ⊆ uverts G* **and** *enot0*: *ε > 0*
and *fin*: *finite X finite Y* **and** *uwellformed G*
and *rp1*: *regular-pair X Y G ε*
and *ed1*: *edge-density X Y G ≥ 2*ε*
and *card (neighbors-ss x Y G) ≥ (edge-density X Y G − ε) * card Y*
shows *card (neighbors-ss x Y G) ≥ ε * card (Y)*
⟨*proof*⟩

lemma *all-edges-btwn-neighbour-sets-lower-bound*:
fixes *ε::real*
assumes *X ⊆ uverts G* **and** *Y ⊆ uverts G* **and** *Z ⊆ uverts G* **and** *ε > 0*
and *finG*: *finite (uverts G)*
and *wf*: *uwellformed G* **and** *fin*: *finite X finite Y finite Z*
and *rp1*: *regular-pair X Y G ε* **and** *rp2*: *regular-pair Y Z G ε* **and** *rp3*: *regular-pair X Z G ε*
and *ed1*: *edge-density X Y G ≥ 2*ε* **and** *ed2*: *edge-density X Z G ≥ 2*ε* **and** *ed3*: *edge-density Y Z G ≥ 2*ε*

and *cond1*: *card* (*neighbors-ss x Y G*) ≥ (*edge-density X Y G* − *ε*) ∗ *card Y*
and *cond2*: *card* (*neighbors-ss x Z G*) ≥ (*edge-density X Z G* − *ε*) ∗ *card Z*
and *x* ∈ *X*
shows *card* (*all-edges-between* (*neighbors-ss x Y G*) (*neighbors-ss x Z G*) *G*)
≥ (*edge-density Y Z G* − *ε*) ∗ *card* (*neighbors-ss x Y G*) ∗ *card* (*neighbors-ss x Z G*)
(**is** *card* (*all-edges-between ?Y′ ?Z′ G*) ≥ (*edge-density Y Z G* − *ε*) ∗ *card ?Y′* ∗ *card ?Z′*)
⟨*proof*⟩

lemma *edge-density-implies-edge-exists*:
fixes *ε*::*real*
assumes *X* ⊆ *uverts G* **and** *Y* ⊆ *uverts G* **and** *ε* > *0* **and** *uwellformed G*
assumes *edge-density X Y G* ≥ *ε*
obtains *e* **where** *e* ∈ *all-edges-between X Y G*
⟨*proof*⟩

We are now ready to show the Triangle Counting Lemma (Theorem 3.13 in Zhao's notes):

theorem *triangle-counting-lemma*:
fixes *ε*::*real*
assumes *xss*: *X* ⊆ *uverts G* **and** *yss*: *Y* ⊆ *uverts G* **and** *zss*: *Z* ⊆ *uverts G* **and** *en0*: *ε* > *0*
and *finG*: *finite* (*uverts G*) **and** *wf*: *uwellformed G*
and *rp1*: *regular-pair X Y G ε* **and** *rp2*: *regular-pair Y Z G ε* **and** *rp3*: *regular-pair X Z G ε*
and *ed1*: *edge-density X Y G* ≥ *2*∗*ε* **and** *ed2*: *edge-density X Z G* ≥ *2*∗*ε* **and** *ed3*: *edge-density Y Z G* ≥ *2*∗*ε*
shows *card* (*triangle-triples X Y Z G*)
≥ (*1*−*2*∗*ε*)∗((*edge-density X Y G*) − *ε*)∗((*edge-density X Z G*) − *ε*) ∗((*edge-density Y Z G*) − *ε*)∗
(*card X*)∗(*card Y*)∗ (*card Z*)
⟨*proof*⟩

definition *regular-graph* :: *uvert set set* ⇒ *ugraph* ⇒ *real* ⇒ *bool*
where *regular-graph P G ε* ≡ ∀ *R S*. *R*∈*P* ⟶ *S*∈*P* ⟶ *regular-pair R S G ε*
for *ε*::*real*

A minimum density, but empty edge sets are excluded.

definition *edge-dense* :: *nat set* ⇒ *nat set* ⇒ *ugraph* ⇒ *real* ⇒ *bool*
where *edge-dense X Y G ε* ≡ *all-edges-between X Y G* = {} ∨ *edge-density X Y G* ≥ *ε*

definition *dense-graph* :: *uvert set set* ⇒ *ugraph* ⇒ *real* ⇒ *bool*
where *dense-graph P G ε* ≡ ∀ *R S*. *R*∈*P* ⟶ *S*∈*P* ⟶ *edge-dense R S G ε* **for** *ε*::*real*

definition *decent* :: *nat set* ⇒ *nat set* ⇒ *ugraph* ⇒ *real* ⇒ *bool*

where *decent X Y G η ≡ all-edges-between X Y G = {} ∨ (real (card X) ≥ η ∧ real (card Y) ≥ η)*

definition *decent-graph :: uvert set set ⇒ ugraph ⇒ real ⇒ bool*
where *decent-graph P G η ≡ ∀ R S. R∈P ⟶ S∈P ⟶ decent R S G η* **for** *ε::real*

The proof of the triangle counting lemma requires ordered triples. For each unordered triple there are six permutations, hence the factor of 1/6 here. This is mentioned briefly on pg 57 of Zhao's notes towards the end of the proof.

lemma *card-convert-triangle-rep*:
fixes *G :: ugraph*
assumes *X ⊆ uverts G* **and** *Y ⊆ uverts G* **and** *Z ⊆ uverts G* **and** *fin*: *finite (uverts G)*
and *wf*: *uwellformed G*
shows *card (triangle-set G) ≥ 1/6 * card {(x, y, z) ∈ X × Y × Z . (triangle-in-graph x y z G)}*
(**is** *- ≥ 1/6 * card ?TT*)
⟨*proof*⟩

lemma *card-convert-triangle-rep-bound*:
fixes *G :: ugraph* **and** *t :: real*
assumes *card {(x, y, z) ∈ X × Y × Z . (triangle-in-graph x y z G)} ≥ t*
assumes *X ⊆ uverts G* **and** *Y ⊆ uverts G* **and** *Z ⊆ uverts G* **and** *fin*: *finite (uverts G)*
and *wf*: *uwellformed G*
shows *card (triangle-set G) ≥ 1/6 *t*
⟨*proof*⟩

lemma *edge-density-eq0*:
assumes *all-edges-between A B G = {}* **and** *X ⊆ A Y ⊆ B*
shows *edge-density X Y G = 0*
⟨*proof*⟩

The following is the Triangle Removal Lemma (Theorem 3.15 in Zhao's notes).

theorem *triangle-removal-lemma*:
fixes *ε :: real*
assumes *egt*: *ε > 0*
shows *∃δ::real > 0. ∀ G. card(uverts G) > 0 ⟶ uwellformed G ⟶*
*card (triangle-set G) ≤ δ * card(uverts G) ^ 3 ⟶*
(∃ Gnew. triangle-free-graph Gnew ∧ uverts Gnew = uverts G ∧ (uedges Gnew ⊆ uedges G) ∧
*card (uedges G − uedges Gnew) ≤ ε * (card (uverts G))²)*
(**is** *∃δ::real > 0. ∀ G. - ⟶ - ⟶ - ⟶ (∃ Gnew. ?Φ G Gnew)*)
⟨*proof*⟩

1.6 Roth's Theorem

We will first need the following corollary of the Triangle Removal Lemma. This is Corollary 3.18 in Zhao's notes:

corollary *corollary-triangle-removal*:
fixes ε :: *real*
assumes $0 < \varepsilon$
shows $\exists N{>}0.\ \forall G.\ card(uverts\ G) > N \longrightarrow uwellformed\ G \longrightarrow unique\text{-}triangles\ G \longrightarrow$
$card\ (uedges\ G) \leq \varepsilon * (card\ (uverts\ G))^2$
⟨*proof*⟩

We are now ready to proceed to the proof of Roth's Theorem for Arithmetic Progressions.

definition *progression3* :: *'a*::*comm-monoid-add* ⇒ *'a* ⇒ *'a set*
where *progression3 k d* ≡ {*k*, *k*+*d*, *k*+*d*+*d*}

lemma *p3-int-iff*: *progression3* (*int k*) (*int d*) ⊆ *int* ' *A* ⟷ *progression3 k d* ⊆ *A*
⟨*proof*⟩

We assume that a set of naturals $A \subseteq \{... < N\}$ does not have any arithmetic progression. We will then show that A is of cardinality $o(N)$.

lemma *RothArithmeticProgressions-aux*:
fixes ε::*real*
assumes $\varepsilon > 0$
obtains *X* **where** $\forall N \geq X.\ \forall A \subseteq \{..{<}N\}.\ (\nexists k\ d.\ d{>}0 \wedge progression3\ k\ d \subseteq A) \longrightarrow card\ A < \varepsilon * real\ N$
⟨*proof*⟩

We finally present the main statement formulated using the upper asymptotic density condition.

theorem *RothArithmeticProgressions*:
assumes *upper-asymptotic-density A* > *0*
shows $\exists k\ d.\ d{>}0 \wedge progression3\ k\ d \subseteq A$
⟨*proof*⟩

end