# Transitive Union-Closed Families

Angeliki Koutsoukou-Argyraki[1,2] and Lawrence C. Paulson[2]

[1]Royal Holloway, University of London
[2]University of Cambridge

February 1, 2025

**Abstract**

We formalise a proof by Aaronson, Ellis and Leader showing that the Union-Closed Conjecture holds for the union-closed family generated by the cyclic translates of any fixed set.

## Contents



# 1 Transitive Union-Closed Families

A family of sets is union-closed if the union of any two sets from the family is in the family. The Union-Closed Conjecture is an open problem in combinatorics posed by Frankl in 1979. It states that for every finite, union-closed family of sets (other than the family containing only the empty set) there exists an element that belongs to at least half of the sets in the family. We formalise a proof by Aaronson, Ellis and Leader showing that the Union-Closed Conjecture holds for the union-closed family generated by the cyclic translates of any fixed set [1].

**theory** *Transitive-Union-Closed-Families*
**imports** *Pluennecke-Ruzsa-Inequality.Pluennecke-Ruzsa-Inequality*

**begin**

**no-notation** *equivalence.Partition* (**infixl** *′/ 75*)

**definition** *union-closed*:: *′a set set* ⇒ *bool*
**where** *union-closed* $\mathcal{F} \equiv (\forall A{\in}\mathcal{F}.\ \forall\ B{\in}\mathcal{F}.\ A \cup B \in \mathcal{F})$

**abbreviation** *set-difference* :: $['a\ set,'a\ set] \Rightarrow 'a\ set$ (**infixl** \ *65*)
  **where** $A \setminus B \equiv A-B$

**locale** *Family* = *additive-abelian-group* +
  **fixes** *R*
  **assumes** *finG*: *finite G*
  **assumes** *RG*: $R \subseteq G$
  **assumes** *R-nonempty*: $R \neq \{\}$

**begin**

**definition** *union-closed-conjecture-property*:: $'a\ set\ set \Rightarrow bool$
  **where** *union-closed-conjecture-property* $\mathcal{F}$
    $\equiv \exists \mathcal{X}\subseteq\mathcal{F}.\ \exists x\in G.\ x \in \bigcap\mathcal{X} \wedge card\ \mathcal{X} \geq card\ \mathcal{F}\ /\ 2$

**definition** $Neighbd \equiv \lambda A.\ sumset\ A\ R$

**definition** $Interior \equiv \lambda A.\ \{x\in G.\ sumset\ \{x\}\ R \subseteq A\}$

**definition** $\mathcal{F} \equiv Neighbd$ ' *Pow G*

We show that the family $\mathcal{F}$ as defined above and appears in the statement of the theorem [1] is actually a finite, nonempty union-closed family indeed.

**lemma** *card$\mathcal{F}$-gt0* [*simp*]: $card\ \mathcal{F} > 0$ **and** *finite$\mathcal{F}$*: *finite* $\mathcal{F}$
  **using** *$\mathcal{F}$-def finG* **by** *fastforce+*

**lemma** *union-closed* $\mathcal{F}$
**proof**−
  **have** $*:\forall\ A \subseteq G.\ \forall\ B \subseteq G.\ (sumset\ A\ R) \cup (sumset\ B\ R) = sumset\ (A \cup B)\ R$
    **by** (*simp add*: *sumset-subset-Un1*)
  **show** *?thesis* **using** $*$
    **by** (*auto simp*: *union-closed-def $\mathcal{F}$-def Neighbd-def*)
**qed**

**lemma** *cardG-gt0*: $card\ G > 0$
  **using** *RG R-nonempty card-0-eq finG* **by** *blast*

**lemma** *$\mathcal{F}$-subset*: $\mathcal{F} \subseteq Pow\ G$
  **by** (*simp add*: *Neighbd-def PowI $\mathcal{F}$-def image-subset-iff sumset-subset-carrier*)

## 1.1 Proof of the main theorem

**lemma** *card-Interior-le*:
  **assumes** $S \subseteq G$
  **shows** $card\ (Interior\ S) \leq card\ S$
**proof** −
  **obtain** *r* **where** $r \in R$
    **using** *R-nonempty* **by** *blast*
  **show** *?thesis*

**proof** (*intro card-inj-on-le*)
**let** *?f* = (λ*x*. *x* ⊕ *r*)
**show** *inj-on ?f* (*Interior S*) *?f* ' *Interior S* ⊆ *S*
**using** *RG* ‹*r* ∈ *R*› **by** (*auto simp*: *Interior-def inj-on-def*)
**show** *finite S*
**using** *assms finG finite-subset* **by** *blast*
**qed**
**qed**

**lemma** *Interior-subset-G* [*iff*]: *Interior S* ⊆ *G*
**using** *Interior-def* **by** *auto*

**lemma** *Neighbd-subset-G* [*iff*]: *Neighbd S* ⊆ *G*
**by** (*simp add*: *Neighbd-def sumset-subset-carrier*)

**lemma** *average-ge*:
**shows** ($\sum$ *S*∈$\mathcal{F}$.(*card S*)) / *card* $\mathcal{F}$ ≥ *card G* / *2*
**proof**−
**define** *f* **where** *f* ≡ λ*S*. *minusset* (*G* \ *Interior S*)

The following corresponds to (1) in the paper.

**have** *1*: *card S* + *card* (*f S*) ≥ *card G* **if** *S* ⊆ *G* **for** *S*
**proof**−
**have** *card* (*f S*) = *card G* − *card* (*Interior S*)
**unfolding** *f-def*
**by** (*metis Diff-subset Interior-subset-G card-Diff-subset card-minusset′ finG finite-subset*)
**with** *that* **show** *?thesis* **using** *card-Interior-le*
**by** (*metis* (*no-types*, *lifting*) *add.commute diff-le-mono2 le-diff-conv*)
**qed**

The following corresponds to (2) in the paper.

**have** *2*: *f S* = *sumset* (*minusset* (*G* \ *S*)) *R* **if** *S* ⊆ *G* **for** *S*
**proof**−
**have** ∗: *x* ∈ *f S* ⟷ *x* ∈ *sumset* (*minusset* (*G* \ *S*)) *R* **if** *x* ∈ *G* **for** *x*
**proof** −
**have** *x* ∈ *f S* ⟷ *inverse x* ∉ *Interior S*
**using** *that minusset.simps* **by** (*fastforce simp*: *f-def*)+
**also have** . . . ⟷ (*sumset* {*inverse x*} *R*) ∩ (*G**S*) ≠ {}
**using** *sumset-subset-carrier that* **by** (*auto simp*: *Interior-def*)
**also have** . . . ⟷ *x* ∈ *sumset* (*minusset* (*G**S*)) *R*
**proof**
**assume** *L*: *sumset* {*inverse x*} *R* ∩ (*G* \ *S*) ≠ {}
**then obtain** *r* **where** *r*: *inverse x* ⊕ *r* ∉ *S* **and** *r* ∈ *R*
**using** ‹*S* ⊆ *G*› ‹*x* ∈ *G*› **by** (*auto simp*: *sumset-eq minusset-eq*)
**then have** *inverse* (*inverse x* ⊕ *r*) ∈ *minusset* (*G* \ *S*)
**using** *RG that* **by** *auto*
**moreover have** *x* = *inverse* (*inverse x* ⊕ *r*) ⊕ *r*
**using** *RG* ‹*r* ∈ *R*› *that commutative inverse-composition-commute invertible-right-inverse2*

```
          by auto
        ultimately show x ∈ sumset (minusset (G \ S)) R
          by (metis RG ‹r ∈ R› minusset-subset-carrier subset-eq sumset.simps)
      next
        assume R: x ∈ sumset (minusset (G \ S)) R
        then obtain g r where *: g∈G g ∉ S r∈R x = inverse g ⊕ r
          by (metis Diff-iff minusset.simps sumset.cases)
        show sumset {inverse x} R ∩ (G \ S) ≠ {}
        proof
          assume sumset {inverse x} R ∩ (G \ S) = {}
          then have g ∉ sumset {inverse x} R
            using ‹g ∉ S› sumset-subset-carrier that by fastforce
          then have g ≠ local.inverse (local.inverse g ⊕ r) ⊕ r
            using * RG that by (auto simp: sumset-eq)
          with * RG that show False
          by (metis commutative invertible invertible-left-inverse2 invertible-right-inverse2
subset-eq)
        qed
      qed
      finally show ?thesis .
    qed
    show ?thesis
    proof
      show f S ⊆ sumset (minusset (G \ S)) R
      using * f-def minusset-subset-carrier by blast
    next
      show sumset (minusset (G \ S)) R ⊆ f S
      by (meson * subset-iff sumset-subset-carrier)
    qed
  qed
  then have f ‘ Pow G ⊆ ℱ
    by (auto simp: Neighbd-def ℱ-def minusset-subset-carrier)
```

The following corresponds to (3) in the paper.

```
  have 3: Neighbd (Interior (sumset A R)) = sumset A R
    if A ⊆ G for A
    using that by (force simp: sumset-eq Neighbd-def Interior-def)
```

"Putting everything together":

```
  moreover
  have sumset X R = sumset Y R
    if X ⊆ G Y ⊆ G
      minusset (G \ Interior (sumset X R)) = minusset (G \ Interior (sumset Y
R))
    for X Y
    using that 3
   by (metis Diff-Diff-Int Int-absorb2 Interior-subset-G inf-commute minus-minusset)
  ultimately have inj-on f ℱ
    by (auto simp: inj-on-def ℱ-def f-def Neighbd-def)
```

**moreover have** *f ‘ ℱ ⊆ ℱ*
  **using** *2 ℱ-def ‹f ‘ Pow G ⊆ ℱ›* **by** *force*
**moreover have** *ℱ ⊆ f ‘ ℱ*
  **by** (*metis ‹inj-on f ℱ› ‹f ‘ ℱ ⊆ ℱ› endo-inj-surj finiteℱ*)
**ultimately have** *bij-betw f ℱ ℱ*
  **by** (*simp add*: *bij-betw-def*)
**then have** *sum-card-eq*: (∑ *S*∈ℱ. *card* (*f S*)) = (∑ *S*∈ℱ. *card S*)
  **by** (*simp add*: *sum.reindex-bij-betw*)

**have** *card G / 2* = (*1* / (*2* ∗ *card ℱ*)) ∗ (∑ *S*∈ℱ. *card G*)
  **by** *simp*
**also have** ... ≤ (*1* / (*2* ∗ *card ℱ*)) ∗ (∑ *S*∈ℱ. *card S* + *card* (*f S*))
  **by** (*intro sum-mono mult-left-mono of-nat-mono 1*) (*auto simp*: *ℱ-def*)
**also have** ... = (*1* / *card ℱ*) ∗ (∑ *S*∈ℱ. *card S*)
  **by** (*simp add*: *sum-card-eq sum.distrib*)
**finally show** *?thesis*
  **by** *argo*
**qed**

We have thus shown that the average size of a set in the family $\mathcal{F}$ is at least $|G|/2$, proving the first part of Theorem 2 in the paper [1]. Using this, we will now show the main statement, i.e. that the Union-Closed Conjecture holds for the family $\mathcal{F}$.

**theorem** *Aaronson-Ellis-Leader-union-closed-conjecture*:
  **shows** *union-closed-conjecture-property ℱ*
**proof** −
— First, quite a big calculation not mentioned in the article: counting all the elements in two different ways.
**have** ∗: (∑ *S*∈ℱ.(*card S*)) = (∑ *x*∈*G*. *card* {*S*∈ℱ. *x*∈*S*})
    **using** *finiteℱ ℱ-subset*
  **proof** *induction*
    **case** *empty*
    **then show** *?case*
      **by** *simp*
  **next**
    **case** (*insert S 𝒢*)
    **then have** *A*: {*T*. (*T* = *S* ∨ *T*∈𝒢) ∧ *x*∈*T*}
          = {*T*∈𝒢. *x*∈*T*} ∪ (*if x*∈*S then* {*S*} *else* {})
      **for** *x*
      **by** *auto*
    **have** *B*: *card* {*T*. (*T* = *S* ∨ *T*∈𝒢) ∧ *x*∈*T*}
        = *card* {*T*∈𝒢. *x*∈*T*} + (*if x*∈*S then 1 else 0*)
      **for** *x*
      **by** (*simp add*: *A card-insert-if insert*)
    **have** *S* = (⋃*x*∈*G*. *if x* ∈ *S then* {*x*} *else* {})
      **using** *insert.prems* **by** *auto*
    **then have** *card S* = *card* (⋃*x*∈*G*. *if x* ∈ *S then* {*x*} *else* {})
      **by** *simp*
    **also have** ... = (∑ *i*∈*G*. *card* (*if i* ∈ *S then* {*i*} *else* {}))

```
      by (intro card-UN-disjoint) (auto simp: finG)
    also have ... = (∑x∈G. if x ∈ S then 1 else 0)
      by (force intro: sum.cong)
    finally have C: card S = (∑x∈G. if x ∈ S then 1 else 0) .
    show ?case
      using insert by (auto simp: sum.distrib B C)
  qed

  have 1/2 ≤ (sum card ℱ) / (card ℱ ∗ card G)
    using mult-right-mono [OF average-ge, of 1 / card G]
    using cardG-gt0 by (simp add: divide-simps split: if-splits)
  also have ... = (∑x∈G. ((card {S∈ℱ. x∈S}) / (card ℱ))) / card G
    by (simp add: ∗ sum-divide-distrib)
  finally have ∗∗: 1/2 ≤ (∑x∈G. card {S∈ℱ. x∈S} / card ℱ) / card G .
    — There is a typo in the paper (bottom of page): instead of x ∈ S it says x ∈
ℱ.
  show ?thesis
  proof (rule ccontr) — Contradict the inequality proved above
    assume ¬ union-closed-conjecture-property ℱ
    then have A: ⋀𝒳 x. ⟦𝒳⊆ℱ; x∈G; x ∈ ⋂𝒳⟧ ⟹ card 𝒳 < card ℱ / 2
      by (fastforce simp: union-closed-conjecture-property-def)
    have (∑x∈G. real (card {S∈ℱ. x∈S})) < (∑x∈G. card ℱ / 2)
    proof (intro sum-strict-mono)
      fix x :: 'a
      assume x ∈ G
      then have card {S∈ℱ. x∈S} < card ℱ / 2
        by (intro A) auto
      then show real (card {S∈ℱ. x∈S}) < real (card ℱ) / 2
        by blast
    qed (use unit-closed finG in auto)
    also have ... = card ℱ ∗ (card G / 2)
      by simp
    finally have B: (∑x∈G. real (card {S∈ℱ. x∈S})) < card ℱ ∗ (card G / 2) .
    have (∑x∈G. card {S∈ℱ. x∈S} / card ℱ) / card G < 1/2
      using divide-strict-right-mono [OF B, of card ℱ ∗ card G]
      using cardG-gt0
      by (simp add: divide-simps sum-divide-distrib)
    with ∗∗ show False
      by argo
  qed
qed

end

end
```

## References

[1] J. Aaronson, D. Ellis, and I. Leader. A note on transitive union-closed families. 28(2), 2021. doihttps://doi.org/10.37236/9956.