Leaning into geometric group theory

Formalizing Cayley graphs, word metrics, quasi-isometries, and the Milnor–Svarc bridge in Lean 4.

Laith Amjad, Alex Buzdugan, Yuqian Shen, Akili Parekh

Supervisor: George Domat

LOG(M)

UNIVERSITY OF MICHIGAN

Why this project?

Geometric group theory studies finitely generated groups by turning them into metric spaces and comparing them.
This project formalizes that passage in Lean:

- Cayley graphs record multiplication by generators.
- Word metrics convert algebraic length into distance.
- Quasi-isometries encode coarse equivalence.
- Milnor–Svarc transfers geometric control back to the group.

Core definitions

Generating set. A subset $S \subseteq G$ generates G if every element of G is a finite product of elements of $S \cup S^{-1}$.
Cayley graph. The graph $\Gamma(G, S)$ has vertex set G and an edge $g \to gs$ for each $s \in S \cup S^{-1}$.
Word metric. The word length $|g|_S$ is the shortest length of a word representing g, and the induced metric is

$$d_S(g, h) = |g^{-1}h|_S.$$

Quasi-isometry. A map $f : X \to Y$ is a quasi-isometry if distances are distorted by uniform multiplicative and additive constants and the image is coarsely dense.
Cayley's theorem. Every group G embeds into $\mathrm{Sym}(G)$ via the left regular action, explaining the symmetry of Cayley graphs.

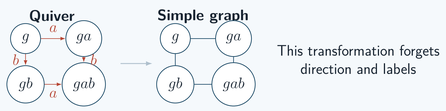
Lean design choices

Two graph models are useful for different proof tasks.

- A **quiver** preserves directed, generator-labelled steps.
- A **simple graph** exposes paths, connectivity, and graph lemmas already available in mathlib.

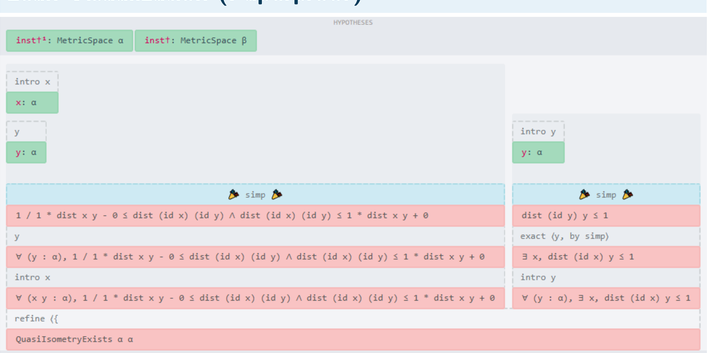
That split keeps the formal objects close to the mathematics while reusing graph infrastructure.

Quiver versus simple graph.



The quiver remembers generators. The simple graph keeps only adjacency for paths and connectivity.

Lean Visualization (Paperproof)



The formalization should preserve the large-scale viewpoint: groups become metric spaces, actions become orbit maps, and Milnor–Svarc turns geometric information into algebraic information.

A minimal Lean skeleton

Lean 4 sketch

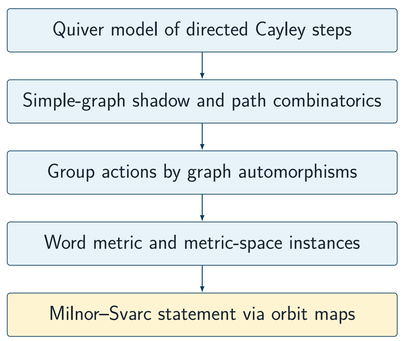
```
structure CayleyGraph (G : Type*) (S : Set G) where
  elt : G

instance [Group G] (S : Set G) : Quiver (CayleyGraph G S) where
  Hom g h :={s // s \in S ∧g.elt * s = h.elt}

structure QuasiIsometry (X Y : Type*)
    [MetricSpace X] [MetricSpace Y] (f : X →Y) where
  A : Real
  hA : 1 ≤A
  B : Real
  hB : 0 ≤B
  C : Real
  hC : 0 ≤C
  dist_le : ∀x x', dist (f x) (f x') ≤A * dist x x' + B
  le_dist : ∀x x', (1 / A) * dist x x' - B ≤dist (f x) (f x')
  quasi_dense : ∀y : Y, ∃x : X, dist y (f x) ≤C
```

This is not the full proof, but it captures the main formal objects.

Formalization pipeline



Milnor–Svarc target

If G acts properly discontinuously and cocompactly by isometries on a proper geodesic metric space X, then:

- G is finitely generated.
- For any basepoint $x_0 \in X$, the orbit map $g \mapsto g \cdot x_0$ is a quasi-isometry.

This is the hinge of the project: geometric control of an action becomes coarse information about the group.

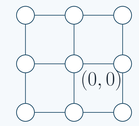
Simple Cayley graph pictures

1. The group $\mathbb{Z}$ with generator $\{1\}$ and $\{2, 3\}$.



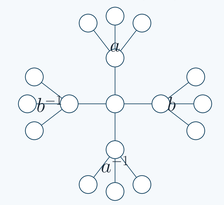
Graph distance agrees with the standard word metric on integers.

2. The square lattice for $\mathbb{Z}^2$ with generators e_1, e_2.



The lattice picture shows that $\mathbb{Z}^2$ looks Euclidean at large scale.

3. A 4-regular tree for the free group $F_2 = \langle a, b \rangle$.



This tree models a free group at large scale: branching is uniform and loops disappear.

Orbit maps and consequences

For a geometric action of G on X, Lean packages the orbit-map proof into reusable ingredients:

- properness of the action,
- cocompactness as coarse surjectivity,
- Lipschitz and reverse bounds for the orbit map.

This interface supports later work on growth, hyperbolicity, and ends.

References

References

[1] M. Bridson and A. Haefliger. *Metric Spaces of Non-Positive Curvature*. Springer, 1999.

[2] P. de la Harpe. *Topics in Geometric Group Theory*. University of Chicago Press, 2000.

[3] The mathlib Community. *mathlib4: Lean mathematics library*. https://github.com/leanprover-community/mathlib4

[4] Paperproof Team. *paperproof*. https://reservoir.lean-lang.org/@Paper-Proof/paperproof/versions

[5] Completed Lean 4 code for this project is available at https://github.com/AndrewShen871/GGT-in-Lean.

Figure 1: Our work on GitHub