Formalizing a classification theorem of low-dimensional solvable Lie algebras in Lean

Paul Schwahn (Universidade Estadual de Campinas)
joint with Viviana del Barco, Gustavo Infanti, Exequiel Rivas

CICM 2025

7 Oct 2025

Overview of the talk

- Who are we?
- Technical background: Formalization, Lean and Mathlib
- Mathematical subject: Lie algebras
- The classification and how to formalize it
- Outlook

The team

Started with limited knowledge about formalization of maths.

Mathematicians:

- Viviana del Barco: professor at Unicamp, differential geometer
- Gustavo Infanti: undergraduate student at Unicamp
- Paul Schwahn: postdoc at Unicamp, differential geometer

Computer scientist:

- Exequiel Rivas: researcher at Tallinn University of Technology, some past experience in Agda/Coq/F*.

Formalization, Lean and Mathlib

- Formalization $\approx$ verifying mathematical proofs by turning them into code (inside a *proof assistant*).
- A statement follows from the axioms if its proof *typechecks*.
- We have chosen *Lean 4* as our proof assistant.
- *Mathlib*: A large community-driven library of definitions/theorems formalized in Lean, focusing on classical mathematics. Tactics for automation.

Classifications in mathematics

- A common problem is the *classification* of a given type of mathematical objects.
- Given a suitable equivalence relation (e.g. isomorphism) on a category of objects, *classification* usually means providing a non-redundant, exhaustive list of representants, ensuring each object under consideration is equivalent to exactly one item on the list.
- Historically, classifications by hand have often suffered from:
  - Redundancy: The same structure appearing multiple times under different guises.
  - Incompleteness: Missing cases.
- This motivates formalizing classification theorems in proof assistants, ensuring correctness and completeness through machine-checked proofs.

Classifications in formalized mathematics

- Focusing on Lean, some formalized classification theorems are:
    - the structure theorem for finitely generated abelian groups (any such group is isomorphic to some $\mathbb{Z}^r \times \prod_i C_{p_i^{n_i}}$),
    - a classification theorem for finite Z-groups (a finite group is a Z-group iff it is isomorphic to a semidirect product of two cyclic subgroups of coprime order),
    - Harper–Wu 2025: the classification of groups of order pq for p, q prime (C_{p^2}, $C_p \times C_p$, C_{pq}, $C_q \rtimes C_p$, where $p < q$).
- That's about it.
- We turned to a classification of *Lie algebras*.

Lie algebras: Background

- In mathematics, a *group* is an abstraction of the set of symmetries of some object.
- Symmetries appear everywhere: in geometry, analysis, physics, nature, ...
- Symmetries are usually *discrete* or *continuous*.

Figure: The 16 symmetries of a regular octagon.

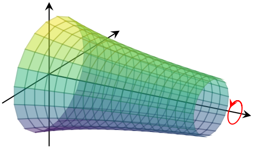


Figure: The rotational symmetry of a surface of revolution.

Lie algebras: Background

- Continuous symmetries are organized in *Lie groups*: groups which are also (differentiable) manifolds such that the group operations are differentiable.
- Most of the structure of a Lie group is already determined by its *Lie algebra*, which consists of infinitesimal symmetries/symmetry generators.

Figure: Sophus Lie, 1842–1899.

Lie algebras: Background

Definition

A *Lie algebra* over a field $\mathbb{K}$ is a $\mathbb{K}$-vector space L together with a map $[\,\cdot\,,\cdot\,] : L \times L \to L$ (called the *Lie bracket*), which

- is *bilinear*: $[ax + y, z] = a[x, z] + [y, z]$ and $[x, ay + z] = a[x, y] + [x, z]$ for all $a \in \mathbb{K}$, $x, y, z \in L$,
- is *skew-symmetric*: $[x, x] = 0$ for all $x \in L$,
- satisfies the *Jacobi identity*: $[x, [y, z]] = [[x, y], z] + [y, [x, z]]$ for all $x, y, z \in L$.

A standard example of a Lie algebra is $\mathfrak{gl}(n, \mathbb{K})$, the space of all $n \times n$-matrices over $\mathbb{K}$ with bracket

$$[A, B] := AB - BA, \qquad A, B \in \mathfrak{gl}(n, \mathbb{K}).$$

Lie algebras: Background

- Like all algebraic structures, Lie algebras have substructures:
  - A *subalgebra* of L is a vector subspace $L' \leq L$ which is closed under the bracket: $\forall x, y \in L',\ [x, y] \in L'$.
  - An *ideal* of L is a vector subspace $L' \leq L$ which is absorbing under the bracket: $\forall x \in L, \forall y \in L',\ [x, y] \in L'$.
- An important tool to study a Lie algebra L is its *derived series* of ideals:

$$\mathcal{D}^0(L) := L,$$
$$\mathcal{D}^{k+1}(L) := [\mathcal{D}^k(L), \mathcal{D}^k(L)] \text{ for } k \in \mathbb{N}_0.$$

$\mathcal{D}^1(L) = [L, L]$ is also called the *commutator ideal* of L.

Lie algebras: Background

Definition

A Lie algebra L is called

- *simple* if the only ideals of L are $\{0\}$ and L itself and if $[\cdot, \cdot] \neq 0$.
- *solvable* if the derived series terminates eventually, i.e. if there is k such that $\mathcal{D}^k(L) = \{0\}$.

- These two properties are mutually exclusive.
- Levi 1905: Any Lie algebra can be "decomposed" into solvable and simple parts (in characteristic 0).
- If $\dim_{\mathbb{K}} L = 3$ (for any field $\mathbb{K}$), then L is either solvable or simple.

Lie algebras in Lean

- Nash 2022: Addition of Lie algebras into Mathlib, good deal of theory, ongoing project to classify simple Lie algebras (over algebraically closed fields of characteristic 0).
  - Oliver Nash: *Formalising Lie algebras*, CPP 2022, pp. 239–250, New York, 2022. Association for Computing Machinery
- In mathlib, algebraic structures are (usually) encoded using *typeclasses* (unbundled), and substructures using *records* (bundled).

Lie algebras in Lean

- A Lie algebra structure on `L : Type*` is a typeclass:

```
class Bracket (L M : Type*) where
  bracket : L → M → M

class LieRing (L : Type*) extends
    AddCommGroup L, Bracket L L where [...]

class LieAlgebra (K L : Type*) [CommRing K]
    [LieRing L] extends Module K L where [...]
```

- A Lie subalgebra is a record:

```
structure LieSubalgebra (K : Type u)
    (L : Type v) [CommRing K] [LieRing L]
    [LieAlgebra K L] extends Submodule K L :
    Type v where [...]
```

The statement to be formalized

Theorem

Let L be a solvable Lie algebra over a field $\mathbb{K}$, with $\dim_{\mathbb{K}} L \leq 3$.

1. If $\dim_{\mathbb{K}} L = 1$, then $L \cong \mathbb{K}$ (abelian).
2. If $\dim_{\mathbb{K}} L = 2$, then $L \cong \mathbb{K}^2$ (abelian) or $L \cong \mathfrak{aff}(\mathbb{K})$.
3. If $\dim_{\mathbb{K}} L = 3$, then L is isomorphic to one of:
   - $\mathbb{K}^3$ (abelian)
   - $\mathfrak{heis}_3(\mathbb{K})$ (Heisenberg)
   - $\mathfrak{aff}(\mathbb{K}) \oplus \mathbb{K}$
   - $\mathfrak{hyp}_3(\mathbb{K})$ (hyperbolic)
   - L_α for some $\alpha \in \mathbb{K}^\times$
   - M_δ for some $\delta \in \mathbb{K}^\times/(\mathbb{K}^\times)^2$

The Lie algebras listed above are pairwise non-isomorphic.

The statement to be formalized

$\dim_{\mathbb{K}}$	**Notation**	**Non-zero brackets**
1	$\mathbb{K}$	–
2	$\mathbb{K}^2$	–
	$\mathfrak{aff}(\mathbb{K})$	$[b_0, b_1] = b_1$
3	$\mathbb{K}^3$	–
	$\mathfrak{heis}_3(\mathbb{K})$	$[b_1, b_2] = b_0$
	$\mathfrak{aff}(\mathbb{K}) \oplus \mathbb{K}$	$[b_1, b_2] = b_1$
	$\mathfrak{hyp}_3(\mathbb{K})$	$[b_0, b_1] = b_1,\ [b_0, b_2] = b_2$
	$L_\alpha,\ \alpha \in \mathbb{K}^\times$	$[b_0, b_1] = b_2,\ [b_0, b_2] = \alpha b_1$
	$M_{[\alpha]},\ \alpha \in \mathbb{K}^\times$	$[b_0, b_1] = b_2,\ [b_0, b_2] = \alpha b_1 + b_2$

Table: Notation for low-dimensional Lie algebras. Here (b_i) is a special basis, and $[b_i, b_j] = -[b_j, b_i]$ is implicit. The entire bracket is then determined by bilinearity.

The formalization

- We characterize the cases in the classification using Lie-algebraic invariants such as $\dim_{\mathbb{K}}[L, L]$.
- The first step is to construct a suitable (vector space) basis for the Lie algebra in question...

```
variable {K L : Type*} [Field K] [LieRing L]
    [LieAlgebra K L]
lemma case1a (dim3 : Module.finrank K L = 3)
    (h₁ : Module.finrank K (commutator K L) = 1)
    (h : IsTwoStepNilpotent K L) :
  ∃ B : Basis (Fin 3) K L, ⁅B 1, B 2⁆ = B 0 ∧
    ⁅B 0, B 1⁆ = 0 ∧ ⁅B 0, B 2⁆ = 0
lemma case1b (dim3 : Module.finrank K L = 3)
    (h₁ : Module.finrank K (commutator K L) = 1)
    (h : ¬ IsTwoStepNilpotent K L) :
  ∃ B : Basis (Fin 3) K L, ⁅B 0, B 1⁆ = 0 ∧
    ⁅B 0, B 2⁆ = 0 ∧ ⁅B 1, B 2⁆ = B 1
```

The formalization

- …then construct an isomorphism to a predefined Lie algebra from this basis.
- This construction depends on arbitrary *choices*, which is why we don't keep the isomorphism as data.

```
theorem Dim3.classification (h : Module.finrank K L = 3)
    (hs : LieAlgebra.IsSolvable L) :
    Nonempty (L ≃ₗ⁅K⁆ (Dim3.Abelian K)) ∨
    Nonempty (L ≃ₗ⁅K⁆ (Heisenberg K)) ∨
    Nonempty (L ≃ₗ⁅K⁆ (AffinePlusAbelian K)) ∨
    Nonempty (L ≃ₗ⁅K⁆ (Hyperbolic K)) ∨
    (∃ α, α ≠ 0 ∧ Nonempty (L ≃ₗ⁅K⁆ (Family K α 0))) ∨
    (∃ α, α ≠ 0 ∧ Nonempty (L ≃ₗ⁅K⁆ (Family K α 1)))
```

- Separately, we formalize theorems to guarantee that the different entries are pairwise non-isomorphic.

The semidirect product

- In order to conduct our formalization project, we provided many auxiliar theorems and constructions that were not in Mathlib. One such construction is the *semidirect product* of Lie algebras.
- This is the one that appears in the Levi decomposition: using it, all Lie algebras are built out of simple and solvable ones.

$$(L, [\,\cdot\,,\cdot\,]_L), \quad (J, [\,\cdot\,,\cdot\,]_J), \quad \varphi : L \to \operatorname{Der} J \quad \rightsquigarrow \quad L \ltimes_\varphi J$$

The semidirect product

Definition

- Let $(J, [\cdot\,,\cdot]_J)$ be a Lie algebra. A *derivation* of J is a linear endomorphism $D \in \mathfrak{gl}(J)$ such that

$$\forall x, y \in J, \qquad D[x,y]_J = [Dx, y]_J + [x, Dy]_J.$$

  The derivations of J form a Lie subalgebra $\operatorname{Der} J \leq \mathfrak{gl}(J)$.

- Let $(L, [\cdot\,,\cdot]_L)$ and $(J, [\cdot\,,\cdot]_J)$ be two Lie algebras over a field $\mathbb{K}$, and $\varphi : L \to \operatorname{Der} J$ a homomorphism of Lie algebras. The *semidirect product* $L \ltimes_\varphi J$ is the vector space product $L \times J$ together with the Lie bracket

$$[(l_1, j_1), (l_2, j_2)] := ([l_1, l_2]_L, [j_1, j_2]_J + \varphi(l_1)j_2 - \varphi(l_2)j_1),$$

  for any $l_1, l_2 \in L$, $j_1, j_2 \in J$.

The semidirect product

```
variable {K : Type*} (L J: Type*) [CommRing K]
  [LieRing L] [LieRing J] [LieAlgebra K L]
  [LieAlgebra K J] (φ : L →ₗ⁅K⁆ LieDerivation K J J)

def LieSemidirectProduct := L × J

notation:35 L " ⋉[" φ:35 "] " J:35 =>
  LieSemidirectProduct L J φ

instance : Bracket (L ⋉[φ] J) (L ⋉[φ] J) where
  bracket := fun a b ↦ ⟨⁅a.1, b.1⁆,
    φ a.1 b.2 - φ b.1 a.2 + ⁅a.2, b.2⁆⟩
```

Conclusion and outlook

So far:

- We formalized necessary results on Lie ideals, semidirect product, etc.
- We obtained an admit-free proof of the classification theorem for solvable Lie algebras of dimension ≤ 3 (over arbitrary fields).
- This may be used as a blueprint for further formalization projects on classifications in mathematics.

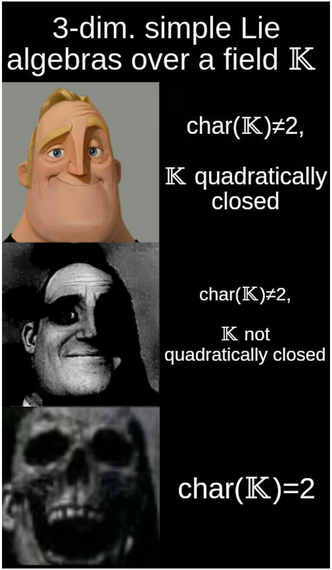
Currently:

- Refine and refactor.
- Upstream our results to Mathlib.

Conclusion and outlook

Further work:

- Extend classification to solvable Lie algebras of higher dimension. (E.g. in dimension 4 the classification is known over arbitrary fields; de Graaf 2004.)
- Formalize classification(s) of simple Lie algebras in dimension 3. (This depends crucially on the algebraic properties of the field.)



Thank you!


WHEN THE
Tactic `rewrite` failed: motive is not type correct:
conv =>
⊢
(GOAL STATE)
imgflip.com