

# Formalising the SOS Decomposition of LCS Games in Lean

## Project Status Report

### Week By Week Recap.

---

#### Week 1 & 2

##### Introduction to Lean

- Got familiar with Proof assistant (cs-428 in rocq)
- Setup Lean
- Basic Proof in Lean

##### LCS & The Mermin-Peres game

- Read : “Characterization of Binary Constraint System Games”  
Got familiar with the Mermin-Peres game, and the notion of LCS games.
- Read : “entanglement and non-locality in games and graphs”, section 1.  
Understand the historical context of non-local games.
- Review CHSH decomposition in SOS to prove tsirelson bound.

#### Week 3

##### Formalising a LCS Strategy

- Working in  $\ast$ -ring [StarRing] instead of a Hilbert space.
- Adapting from the CHSH.lean file.
- Setup a formal definition of a commuting measurement systems  $\{E_{i,x}\}$ .
- Formally defined the LCSStrategy

##### 4.7 SOS approach to solution group

- Assimilate the construction of the observable for a LCS strategy.

#### Week 4

##### Starting Proofs, Mermin-Peres & Documentation

- Made&Published documentation
- Reworked the LCS framework (geometry of the game)
- Defined an LCS game
- Proved some measurement Lemmas
- Tried Mermin-Peres square strategy.

#### Week 5

##### Observable to Strategy

- Defined A strategy given in terms of Observable.
- Attempt to prove that given a strategy defined in terms of observable, we can construct a strategy defined in terms of projectors.
- Proved that  $A_j^i B_j$  are valid observables (Introduce InducedMeasurementSystem)
- Reformulated the proof that Alice observable commute for different variables of the same equation.

##### Prove 4.7.1 & 4.7.2

- Prove 4.7.1 & 4.7.2 of the paper (fully automated by Lean).

#### Week 6

##### Observable to Strategy

- Prove equivalence of Strategy given in terms of Observable and Strategy given in terms of Projectors.
- Add MathLib style documentation

##### Prep work for SOS decomposition

- Define Local Winning/Loss Operator for main proof
- Introduce ObservableToProjector Lemmas
- Reorganize codebase

#### Week 7 + Easter Break (Milestone 1&2)

##### Show that the Magic Square Grid is a valid strategy

- Setup pauli matrices algebra
- Define the game layout and the grid of obserables
- Define a Bipartite Observable Strategy
- Prove commutation to show that the Mermin-Peres square strategy is valid.

##### Formalize The SOS Decomposition of an LCS game

- Add auxiliary lemmas of commutation over Alice and Bob observables
- Define Alice product of obserables for a given equation.
- Decomposed the proof in 5 steps.
- Add custom notation for readability.
- Sectionned codebase for docs.

#### Week 8

##### Update Manual&Docs

- Update manual with the work done for the first two milestones.

##### Compare with other libraries

- Compare project with other lean quantum information library.
- Plan following weeks and define the next milestone.

#### Week 9

##### Group Theory

- Explore mathlib PresentedGroup and FreeGroup.
- Define a Linear System Object in Lean from the Layout and Game definition.
- Implement a mapping from an LCS game to a group presentation of the solution group.

## Week 10

### Deriving Global equalities from EPR annihilating local loss operators

- Use milestone 2 to rewrite into the Sum of Square decomposition of the game.
- Use self-adjoint property to show each term of the sum of square must be 0 for a perfect strategy.
- Extract from the condition of each term being 0 identities used to prove the representation of the solution group.

---

## Week 11

### Representation of the Solution Group

- Maps each generator of the solution group to an operator defined from the strategy, this is a mapping to the group of unitary matrices.
- Prove that each of the relations of the group presentation are satisfied by the image of this mapping, thus proving that we have a homomorphism from the solution group to a group of unitary matrices.

#### —Constructing the Solution Group Representation via Matrix

(Milestone 3)

#### Homomorphisms—

•  
•  
•

# Meetings

**Monday 11.15am in BC110** to debrief previous week and plan the next one.

## Week 10

### Summary of the week

0. refactor project structure.
1. Add a linear system layer, to follow notation of the paper more closely.
2. Define the alphabet of the solution group as the set of variables of the linear system and a special generator  $J$ .
3. Define first a freeGroup from the alphabet.
4. add helper definitions to define the relations of the group presentation, mostly the last constraint that required extra work.
5. pack all the relations into a single set of relations (define by a membership predicate to test if a word is in the relation)
6. Pass this relations to the mathlib PresentedGroup construction to get the solution group.
7. Uses this definition with Magic Square game and check the relations generated.

Next step if we want to prove homomorphism, we need a function  $f$  to map each generator to an operator and we need a proof for each relation that the image still satisfies.

**Lean specific details** *Empty Private* section for the representation, make sure it does not appear in the API.

### Question

- How is the solution group presentation and the concrete homomorphism to a group of pauli matrices related to the SOS decomposition of the game ?
- What would be a good 3rd milestone for the project ?

## Week 7 & Easter Break

*Finished Milestone 1 & 2*

### Lean general discovery

- Cauchy Sequence for Real number, Complex over rationnal possibly.
- LLM being very verbose, two solutions
  1. Introduce custom notation (example for SOS equivalence)
  2. Intrdouce macro / custom notation (example in pauli & comm for MagicSquare)
- Module tactic to simplify trivial steps.
- Section and docstring as well as private lemmas to simplify. (doc4-gen)

### Implementation specific

- Pauli algebras instead of importing them.
- Bipartite Observable Strategy that lifts into two seperate distince spaces for alice and bob.

### Where to go next

- Write report of this milestone in manual style, simplify implementation?

## Week 6

### 1. Add Condition to the ObservableStrategy Game

- Missing in the description of the strategy was that each of Alice observable must commute with each of Bob observable.

### 2. Define ObservableToProjector with Lemmas to prove the construction is a valid PVM

- ### 3. In Equivalence. lean prove that the strategy defined in terms of observable is equivalent to the strategy defined in terms of projector.

- Easy for Bob, Long for Alice requires one or more lemmas per projector

#### 4. **Introduce Local Winning/Loss Operator for main proof**

#### 5. **Restructure the codebase**

Introduce Game/ Strategy/ and moved everything specific to a Projector based strategy in its own file, while keeping general lemmas about measurement and observable in their own files.

Goal :

- Read & Understand Proofs in Observable and WinningCondition.
- Try to finalize the Observable->Strategy file to show that the Mermin-Peres square strategy is valid.
- Start Main proof if time allows.
- Catch up with the documentation.

### **Week 5**

#### 1. **Showing that $A_j^i$ and $B_j$ are valid observables.**

*Observable.lean*

- Defining IsObservable as element that are self-adjoint and square to identity.
  - Proving that  $A_j^i$  and  $B_j$  are valid observables so that we can use these properties in the main proof.
  - To do this we introduce inducesMeasurementSystem
    1. If we have a valid Projector family we can construct another family by mapping outcomes to 0 and 1.
    2. Prove that for a binary projector family,  $P(0) - P(1)$  is an observable.
    3. So we prove that  $A_j^i$  is valid by showing that it is an induced measurement system where the partition function is given by the value of the variable in the equation, thus the observable that is the difference of the two projectors is exactly  $A_j^i$ .
2. **Proving 4.7.1 & 4.7.2 of the paper.**

*WinningCondition.lean*

- Both Lemmas are needed in the main proof.
  - Proofs were written fully by LLMs and then simplified with another pass of LLMs.
3. **Generalize last week attempt to show that the Mermin-Peres square strategy is valid.**

*ObservableStrategy.lean*

- Define a strategy given in terms of observable and try to prove that we can construct a strategy given in terms of projectors, which is the form of strategy we need for the proof.
  - Endgoal is to use this to show that the Mermin-Peres square strategy is valid.
  - LLMs switched to Module instead of Algebra.
  - File is not worth looking at right now.
4. **General remarks.**
- Tried Codex and Antigravity, ran out of budget in one hour.

### **Week 4 :**

#### 1. **Made&Published documentation**

Available for consultation, to use while debriefing and throughout the weeks to check on the progress. The source raw files are also available at src/

*Use docs for the rest of the debrief.*

#### 2. **Reworked the LCS framework**

added geometry of the game to remove long signature.

#### 3. **Defined an LCS game**

as just a function from index of constraints to the value of the constraints

#### 4. **Tried Mermin-Peres square strategy**

Multiple challenges:

- Noncomputable because of complex numbers.
- Working with matrices
- Proving commutation.
- Gave up for now.

#### 5. **Proved some measurement Lemmas to get to Alice observable commutes.**

- Proved that Alice measurement commute for the same question but different answer.
- Starting to define the 4.7.1 & 4.7.2

#### 6. **Noticed Typo in the Original Paper**

### **Week 3 :**

1. **Defined a strategy for a LCS game in Lean.**
2. **Getting familiar with the notation of the paper and the construction of the observable.**

### **Week 1 & 2**

#### 1. **Clarify Project Goal and Scope**

Clarify that the goal is to formalize a proof that is general to LCS game and not just the Mermin-Peres square. The proof gives a condition on the existence of a perfect strategy for an LCS game.

#### 2. **Learn about the Context of the Mermin-Peres square and LCS games**

# Bibliography