Seymour’s Theorem Formalization

Project Status Update

Martin Dvořák, Ivan Sergeev

03 July 2025

Overall Status

Current Status of 3-Sum

01	3-Sum of Matrices	• Basic definitions • API
02	Re-signing of Matrices	• Re-signing 3x3 matrices • Re-signing matrices based on 3x3 submatrices • Total unimodularity
03	Canonical Signing of 3-Sum	• Re-signing of summands • Canonical signing of 3-sum • Total unimodularity
04	Family of 3-Sum-Like Matrices	• Structural definition • 3-sums are generalized • Total unimodularity
05	3-Sum of Matroids	• Standard representations • Matroids having a standard representation • Regularity

List of Contributions

Left block in canonically signed 3-sum is TU	Rida Hadamani
3-sum-like matrices are TU	Evgenia Karunus
Form of rows and columns of bottom-left block in canonically signed 3-sum	Alex Meiburg
Bottom-left block of canonically signed 3-sum = sum of outer products of vectors	Ivan Sergeev
Correctness of canonical signing of 3-sum (resulting form, is indeed signing)	Ivan Sergeev
Pivoting in top-left block of 3-sum-like matrix yields a 3-sum-like matrix	Ivan Sergeev & Martin Dvorak
TUness of extension of TU left summand matrix with parallel and zero rows	Martin Dvorak
Merging pull requests + reviewing and optimizing code + maintaining blueprint	Martin Dvorak

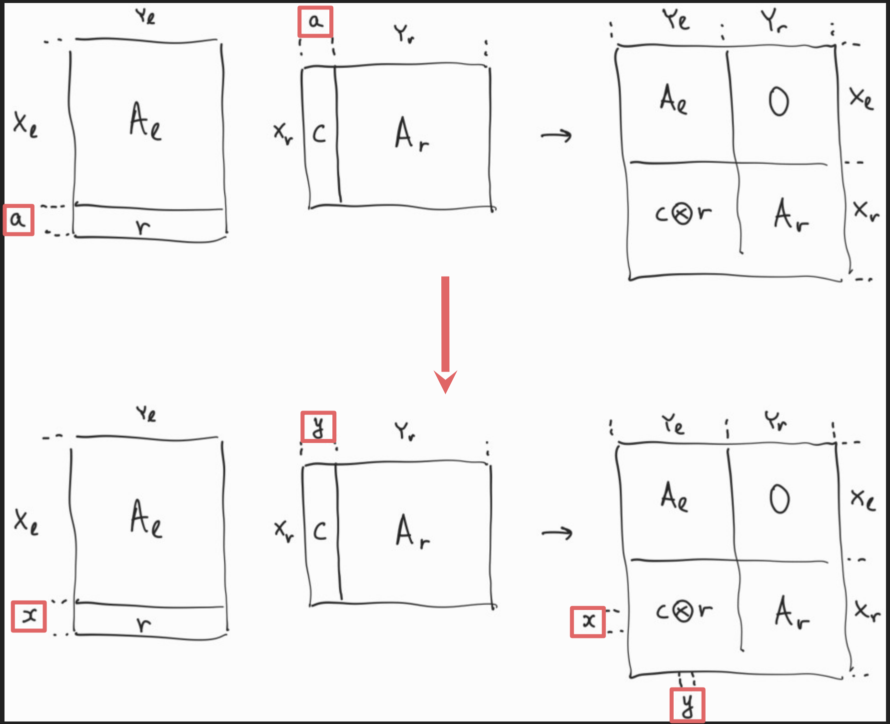
MVP Milestone Reached!

Verification of Verification: 2-Sum

```
def standardRepr2sumComposition {α : Type} [DecidableEq α] {a : α} {Sₗ Sᵣ : StandardRepr α Z2}
    (ha : Sₗ.X ∩ Sᵣ.Y = {a}) (hXY : Sᵣ.X ⫗ Sₗ.Y) :
```

```
noncomputable def standardReprSum2 {α : Type} [DecidableEq α] {Sₗ Sᵣ : StandardRepr α Z2} {x y : α}
    (hXX : Sₗ.X ∩ Sᵣ.X = {x}) (hYY : Sₗ.Y ∩ Sᵣ.Y = {y}) (hXY : Sₗ.X ⫗ Sᵣ.Y) (hYX : Sₗ.Y ⫗ Sᵣ.X) :
```

Verification of Verification: 2-Sum



Verification of Verification: Standard Representation Level

```lean
/-- `StandardRepr`-level 1-sum of two matroids.
    It checks that everything is disjoint (returned as `.snd` of the output). -/
def standardRepr1sumComposition {Sₗ Sᵣ : StandardRepr α Z2} (hXY : Sₗ.X ⫗ Sᵣ.Y) (hYX : Sₗ.Y ⫗ Sᵣ.X) :
    StandardRepr α Z2 × Prop :=
  ⟨
    ⟨
      Sₗ.X ∪ Sᵣ.X,
      Sₗ.Y ∪ Sᵣ.Y,
      by simp only [Set.disjoint_union_left, Set.disjoint_union_right]; exact ⟨⟨Sₗ.hXY, hYX.symm⟩, ⟨hXY, Sᵣ.hXY⟩⟩,
      (matrix1sumComposition Sₗ.B Sᵣ.B).toMatrixUnionUnion,
      inferInstance,
      inferInstance,
    ⟩,
    Sₗ.X ⫗ Sᵣ.X ∧ Sₗ.Y ⫗ Sᵣ.Y
  ⟩
```

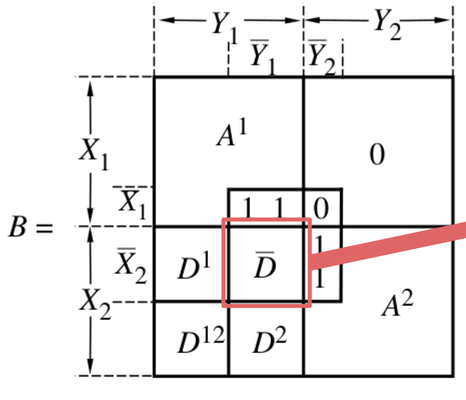
```lean
/-- `StandardRepr`-level 1-sum of two matroids. Returns the result only if valid. -/
noncomputable def standardReprSum1 {Sₗ Sᵣ : StandardRepr α Z2} (hXY : Sₗ.X ⫗ Sᵣ.Y) (hYX : Sₗ.Y ⫗ Sᵣ.X) :
    Option (StandardRepr α Z2) :=
  open scoped Classical in if
    Sₗ.X ⫗ Sᵣ.X ∧ Sₗ.Y ⫗ Sᵣ.Y
  then
    some ⟨
      -- row indices
      Sₗ.X ∪ Sᵣ.X,
      -- col indices
      Sₗ.Y ∪ Sᵣ.Y,
      -- row and col indices are disjoint
      by rw [Set.disjoint_union_right, Set.disjoint_union_left, Set.disjoint_union_left]
         exact ⟨⟨Sₗ.hXY, hYX.symm⟩, ⟨hXY, Sᵣ.hXY⟩⟩,
      -- standard representation matrix
      (matrixSum1 Sₗ.B Sᵣ.B).toMatrixUnionUnion,
      -- decidability of row indices
      inferInstance,
      -- decidability of col indices
      inferInstance⟩
  else
    none
```

Verification of Verification: Matroid Level

```
structure Matroid.Is1sumOf (M : Matroid α) (Mₗ Mᵣ : Matroid α) where
  S : StandardRepr α Z2
  Sₗ : StandardRepr α Z2
  Sᵣ : StandardRepr α Z2
  hSₗ : Finite Sₗ.X
  hSᵣ : Finite Sᵣ.X
  hM : S.toMatroid = M
  hMₗ : Sₗ.toMatroid = Mₗ
  hMᵣ : Sᵣ.toMatroid = Mᵣ
  hXY : Sₗ.X ⫗ Sᵣ.Y
  hYX : Sₗ.Y ⫗ Sᵣ.X
  IsSum : (standardRepr1sumComposition hXY hYX).fst = S
  IsValid : (standardRepr1sumComposition hXY hYX).snd
```

```
/-- Binary matroid `M` is a result of 1-summing `Mₗ` and `Mᵣ` in some way. -/
def Matroid.Is1sumOf (M : Matroid α) (Mₗ Mᵣ : Matroid α) : Prop :=
  ∃ S Sₗ Sᵣ : StandardRepr α Z2,
  ∃ hXY : Sₗ.X ⫗ Sᵣ.Y,
  ∃ hYX : Sₗ.Y ⫗ Sᵣ.X,
  standardReprSum1 hXY hYX = some S
  ∧ Finite Sₗ.X
  ∧ Finite Sᵣ.X
  ∧ S.toMatroid = M
  ∧ Sₗ.toMatroid = Mₗ
  ∧ Sᵣ.toMatroid = Mᵣ
```

Verification of Verification: Extension of 3-Sum



Matrix B of 3-sum

For this book, 3-sums are of considerable importance. A good choice for the matrix $\overline{B}$ representing $\overline{M}$ turns out to be

$$\overline{B} = \begin{array}{|c|c|} \hline C^1 & 0 \\ \hline \overline{D} & C^2 \\ \hline \end{array} = \begin{array}{|cc|c|} \hline 1 & 1 & 0 \\ \hline \multicolumn{2}{|c|}{\overline{D}} & \begin{array}{c} 1 \\ 1 \end{array} \\ \hline \end{array} \tag{8.3.9}$$

Matrix $\overline{B}$ of 3-sum

where $\overline{D}$ is any 2×2 GF(2)-nonsingular matrix. If $\overline{D}$ is the 2×2 identity matrix, then by (5.2.8), $\overline{B}$ represents up to indices $M(W_3)$, which is the graphic matroid of the wheel with three spokes. If $\overline{D}$ contains exactly three 1s, the only other choice, then by one GF(2)-pivot, say in $C^1 = [1\ 1]$, we obtain the former case. Thus, in all instances, $\overline{M}$ is an $M(W_3)$ minor of M.

Current: *D* is either the identity or contains exactly three ones

Desired: *D* is invertible

Next Steps: Timeline

Jul 07–11	Finalize 3-sum
Jul 14–18	Paper draft 1
Jul 21–25	Paper draft 1
Jul 28–Aug 1	Paper draft 2
Aug 04–08	Paper draft 3
Aug 11–15	Paper draft 4
Aug 18–22	Paper draft 5
Aug 25–29	Backup

Venue: CPP 2026

Abstract: 5 September 2025

Paper: 12 September 2025

Conference: January 12–13 2026