Some graph products and their expansion properties

Andrew Brown, EPFL

Joint work with Amin Shokrollahi.

Introduction

• Graph products have recently been used to construct **explicit families of expander graphs** (The zig-zag product [RVW]).

• This is a **recursive** construction that uses **graph products**.

• **Question**: Can we, in a similar way, use products of codes to recursively construct explicit families of good binary codes?

• It turns out that the problem of finding good binary codes can be rephrased as finding Cayley graphs over $(\mathbb{F}_2^k, +)$ that are good expanders

Expander graphs

- Different ways to characterize expander graphs.

- The most intuitive is that any set of nodes must have many neighbors (**combinatorics**)

- There is also an **algebraic** characterization: Look at $\lambda(\mathcal{G})$, the **second largest eigenvalue** (in absolute value) of the normalized adjacency matrix of the graph.

- Smaller $\lambda(\mathcal{G})$ means better expansion

- A **constant degree expander family** is a family $\{\mathcal{G}_i\}_i$ of $[n_i, d, \lambda_i]$-graphs with $\lim_{i\to\infty} n_i = \infty$ and $\lambda_i \leq \lambda$ for some fixed $\lambda < 1$.

- Random regular graphs are good expanders.

- **Applications:** Derandomization, cryptography, circuit complexity, topology, etc...

Code - Expander connection

• **Family of good codes:** A family $\{\mathcal{C}_i\}_i$ of codes with parameters $[n_i, k_i, d_i]$, with $k_i/n_i \leq R$ and $d_i/n_i \leq \delta$ for some $R, \delta < 1$ ($\lim_{i\to\infty} n_i = \infty$).

• Different ways to relate expander graphs to error correcting codes:

• **Expander codes** (Sipser, Spielman). From a family of expander graphs, construct a family of good codes.

• Since there are known explicit constructions for the required expander families, this leads to explicit constructions of good codes.

• Codes described by their **Tanner graph**

Code - Expander connection

• **Cayley graph:** Given a group G and a generating set S. We consider the graph with:

Nodes: elements of G

Edges: $g_1 \sim g_2 \iff \exists s \in S : g_2 = g_1 + s$.

• Take the $k \times n$ generator matrix of binary code $\mathcal{C}$. It has rank k.

• So its n columns generate $(\mathbb{F}_2^k, +)$. We let $\mathcal{G}(\mathcal{C})$ be the **Cayley graph** of $(\mathbb{F}_2^k, +)$ with respect to this generating set.

• **Theorem.** The parameters are the following:

$$\Big[n, k, d\Big]\text{-code} \quad \rightarrow \quad \Big[2^k, n, 1 - \frac{2d}{n}\Big]\text{-graph}$$

• So good codes lead to good expanders.

- **Recall:**
  - We are looking to define code products
  - We have a correspondance:

$$\text{Code} \leftrightarrow \text{Cayley graph over } \mathbb{F}_2^k$$

- **Obvious idea:** What about applying the zig-zag to the Cayley graphs? *Problem:* The result is no longer an $\mathbb{F}_2^k$-Cayley graph.

- Need a graph product that preserves this property.

- A graph product that does this: **Tensor product**

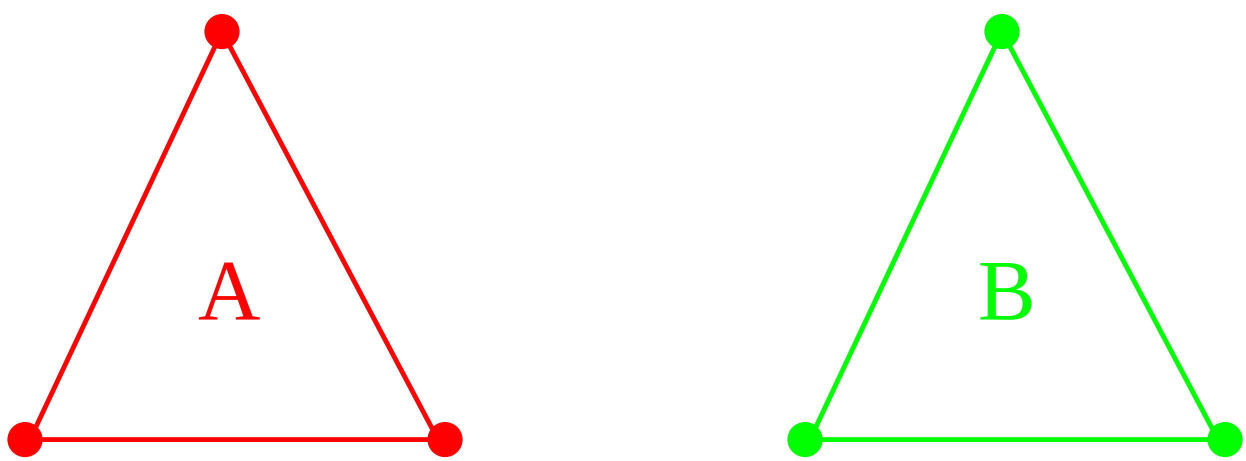
Graph tensoring

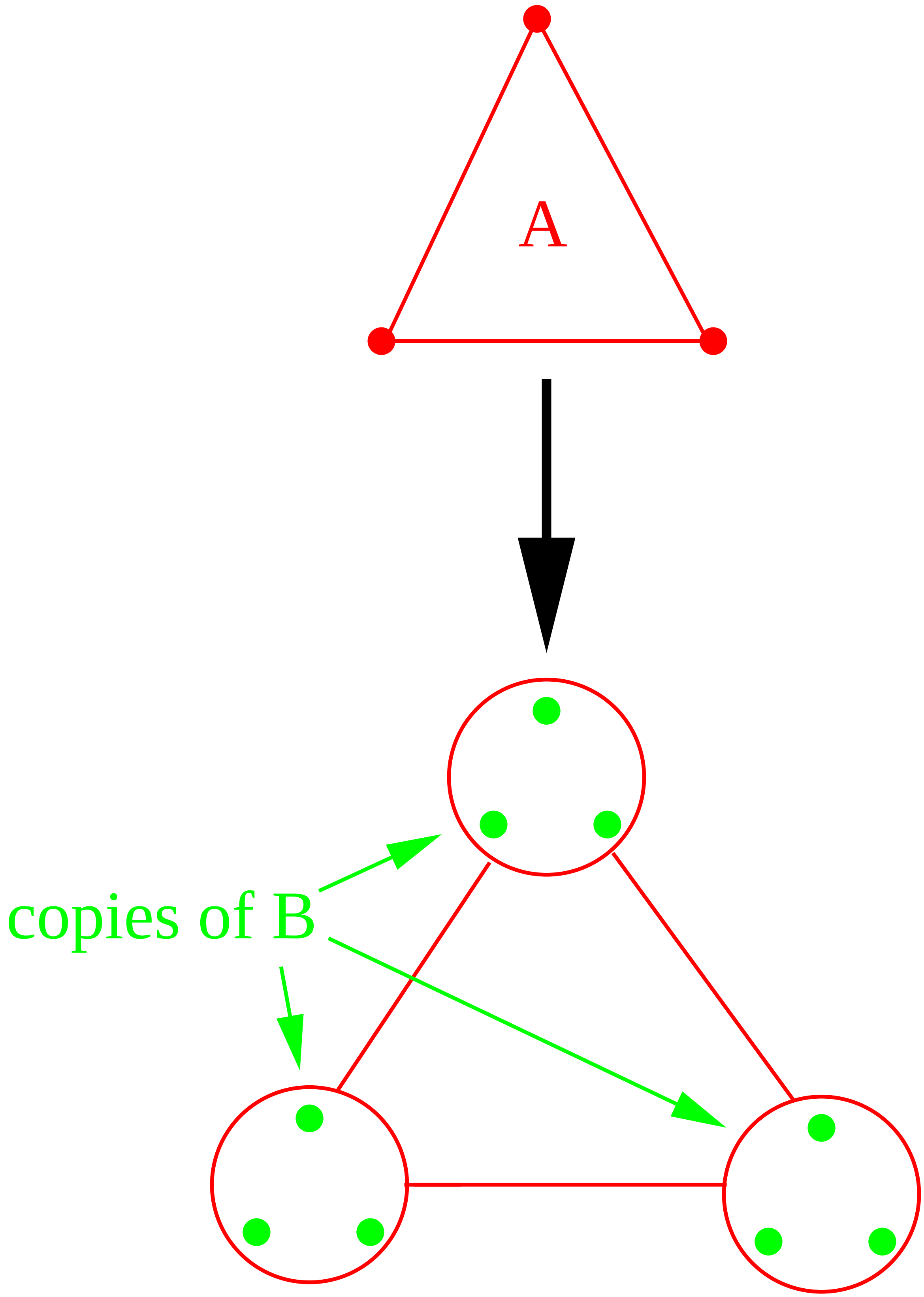
- A, B graphs with node sets

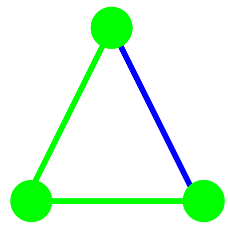
$$[n_A] = \{1, \ldots, n_A\}$$
$$[n_B] = \{1, \ldots, n_B\}$$

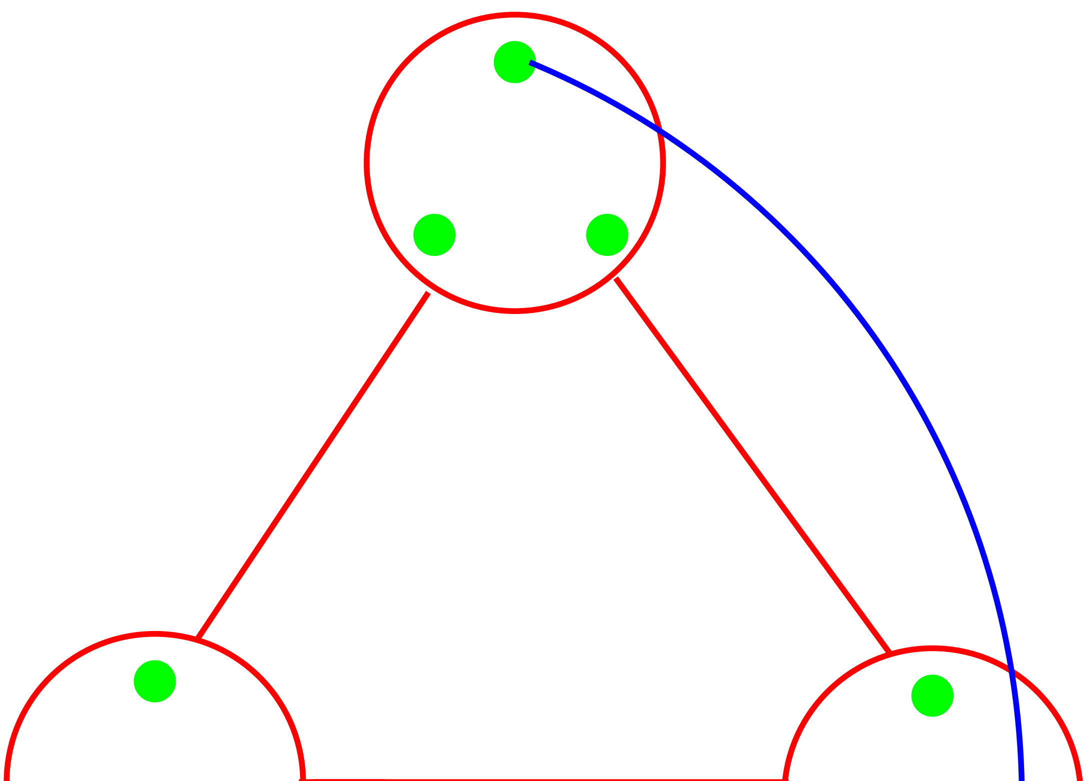
- $A \otimes B$:
  - Nodes: $[n_A] \times [n_B]$
  - Edges: $(a, b) \sim (a', b') \iff \begin{array}{l} a \sim_A a' \\ b \sim_B b' \end{array}$

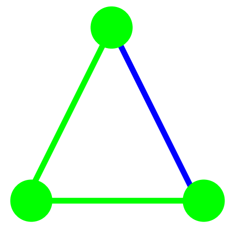
Tensor product $A \otimes B$

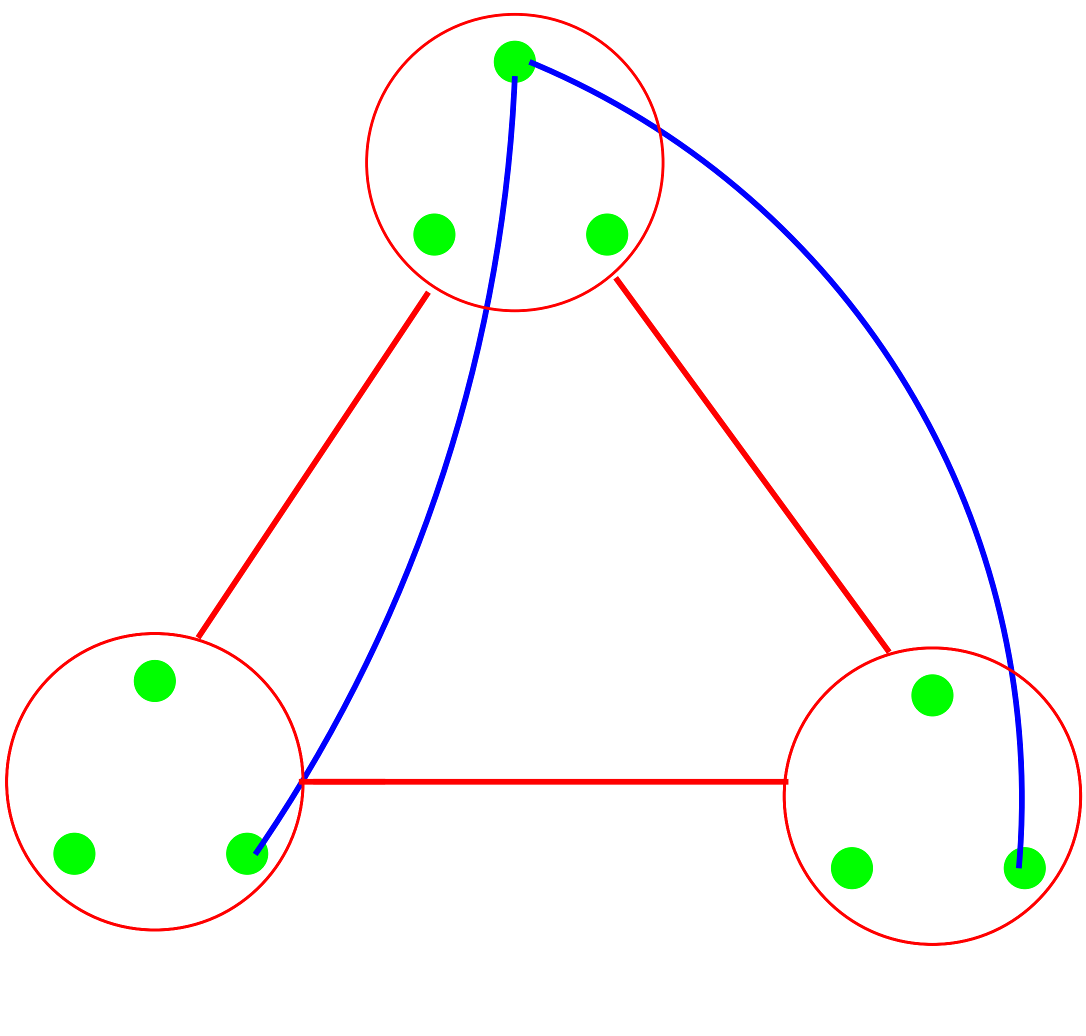


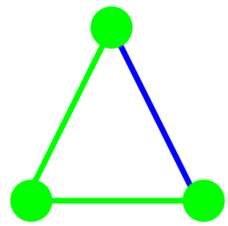

A
copies of B

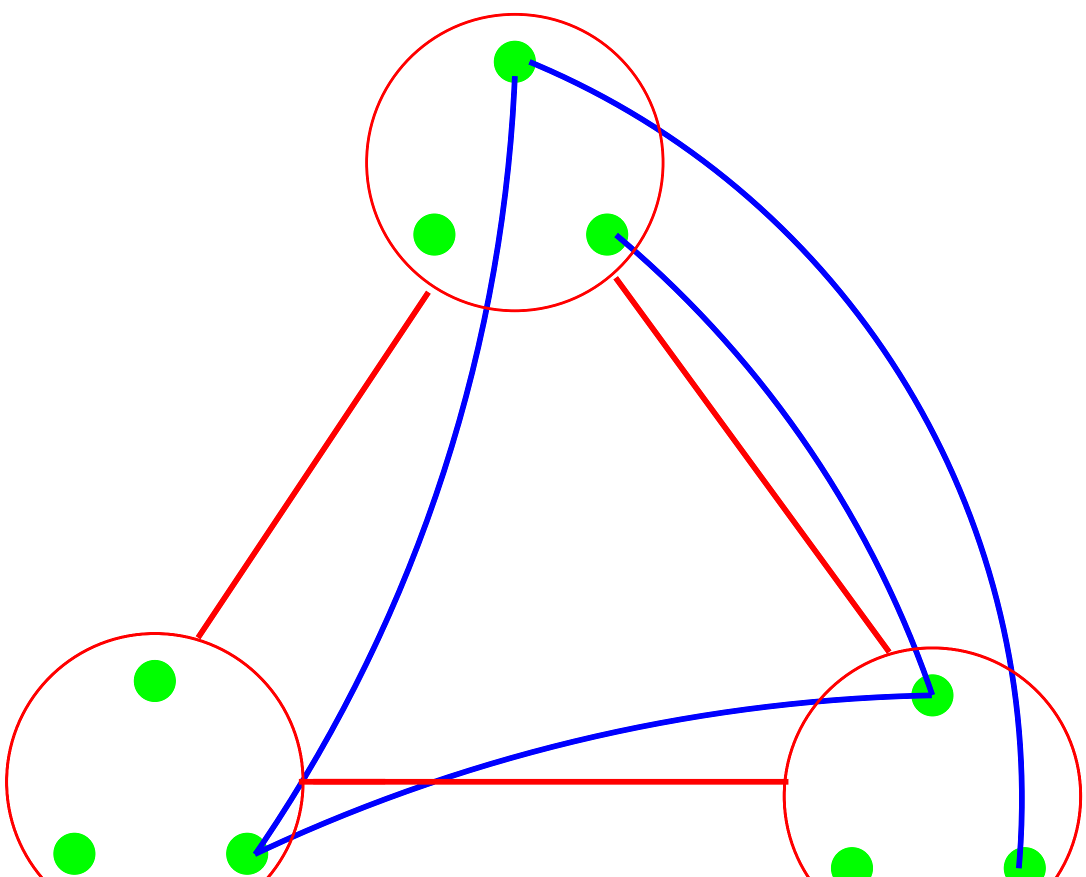


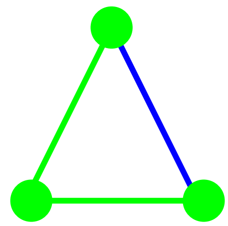


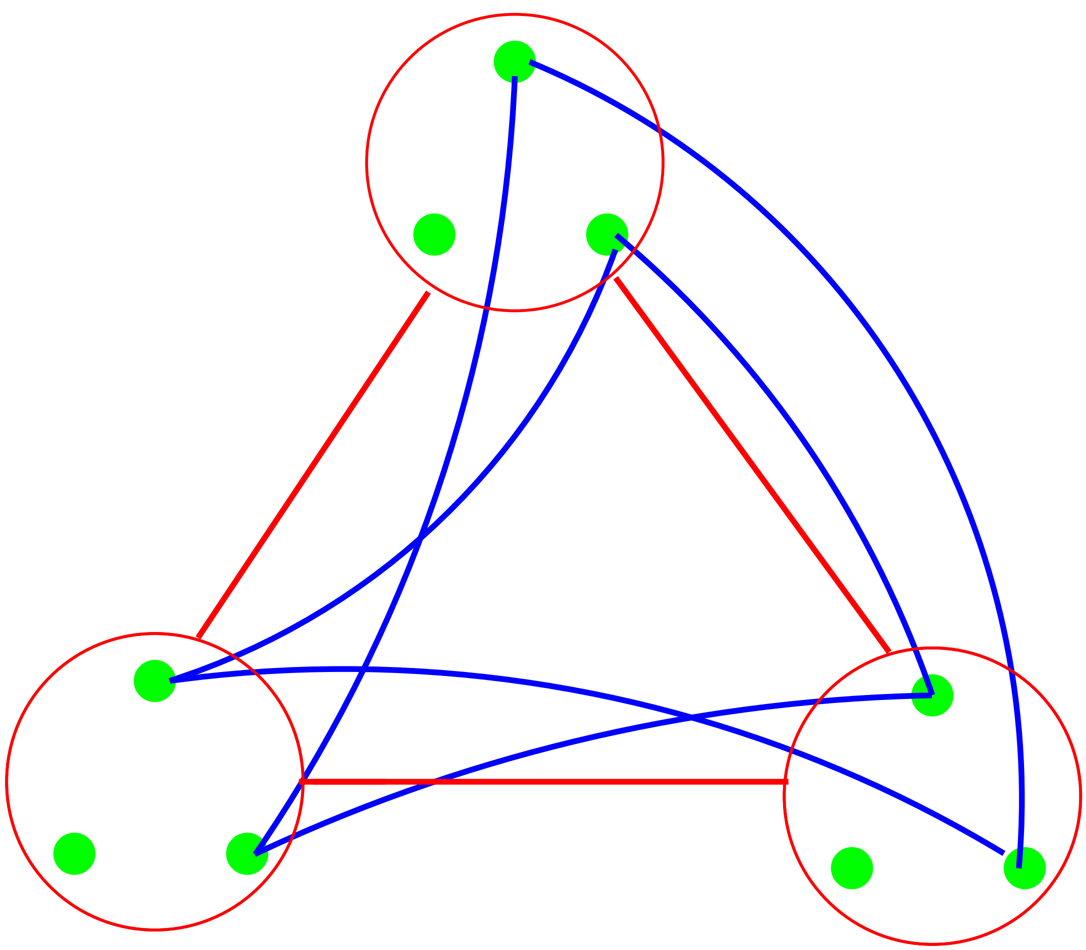


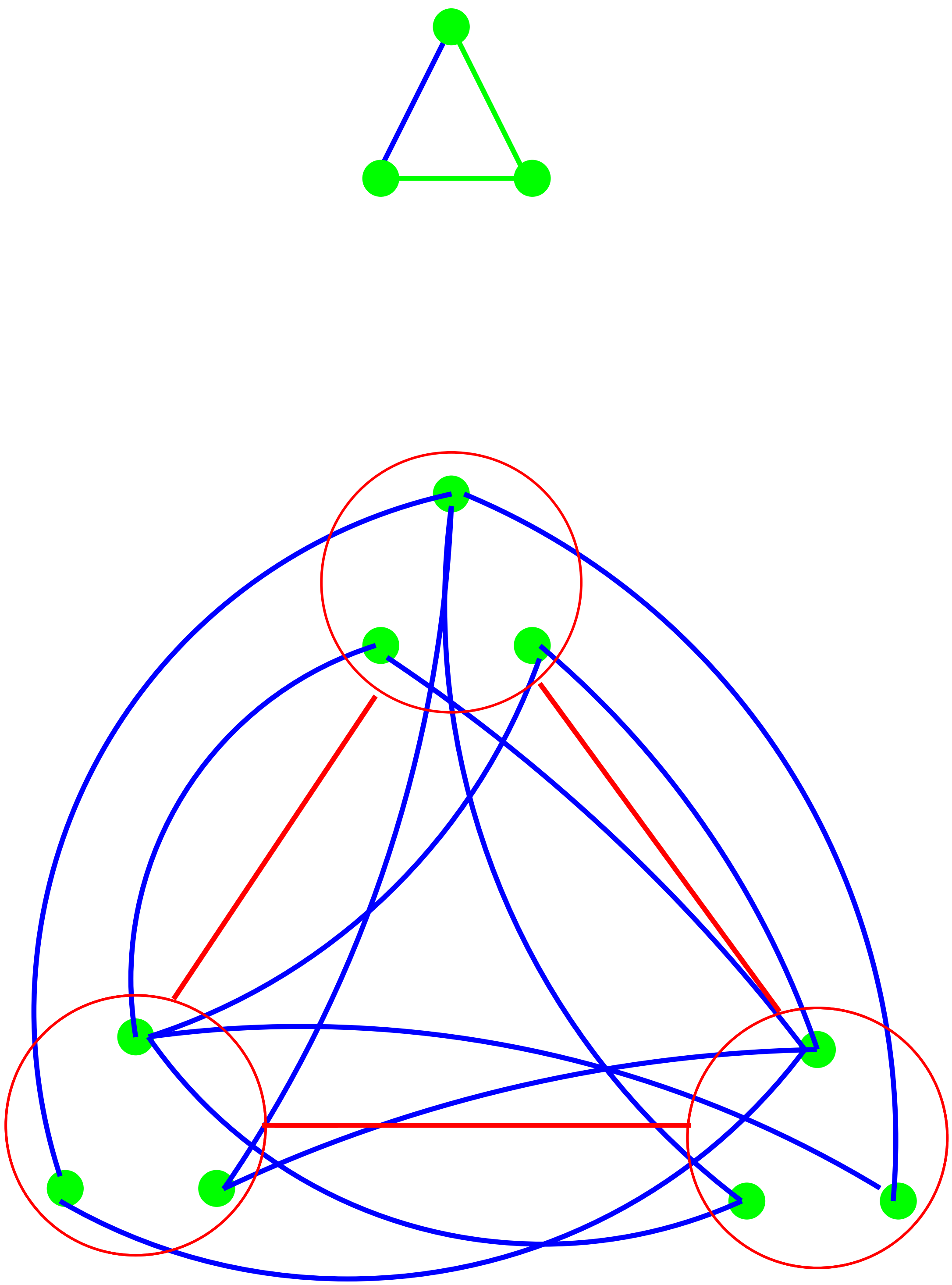


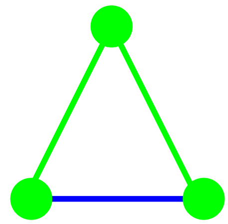


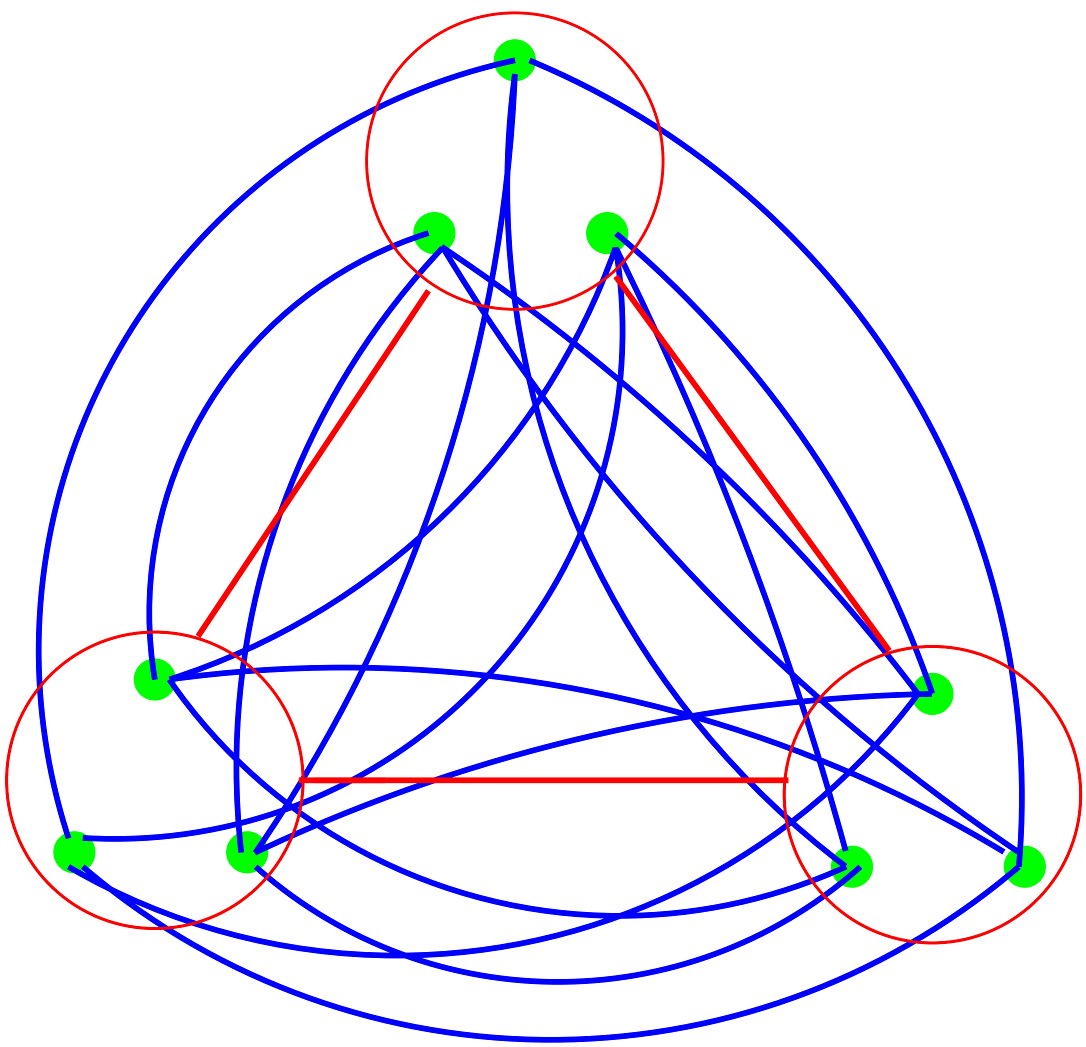


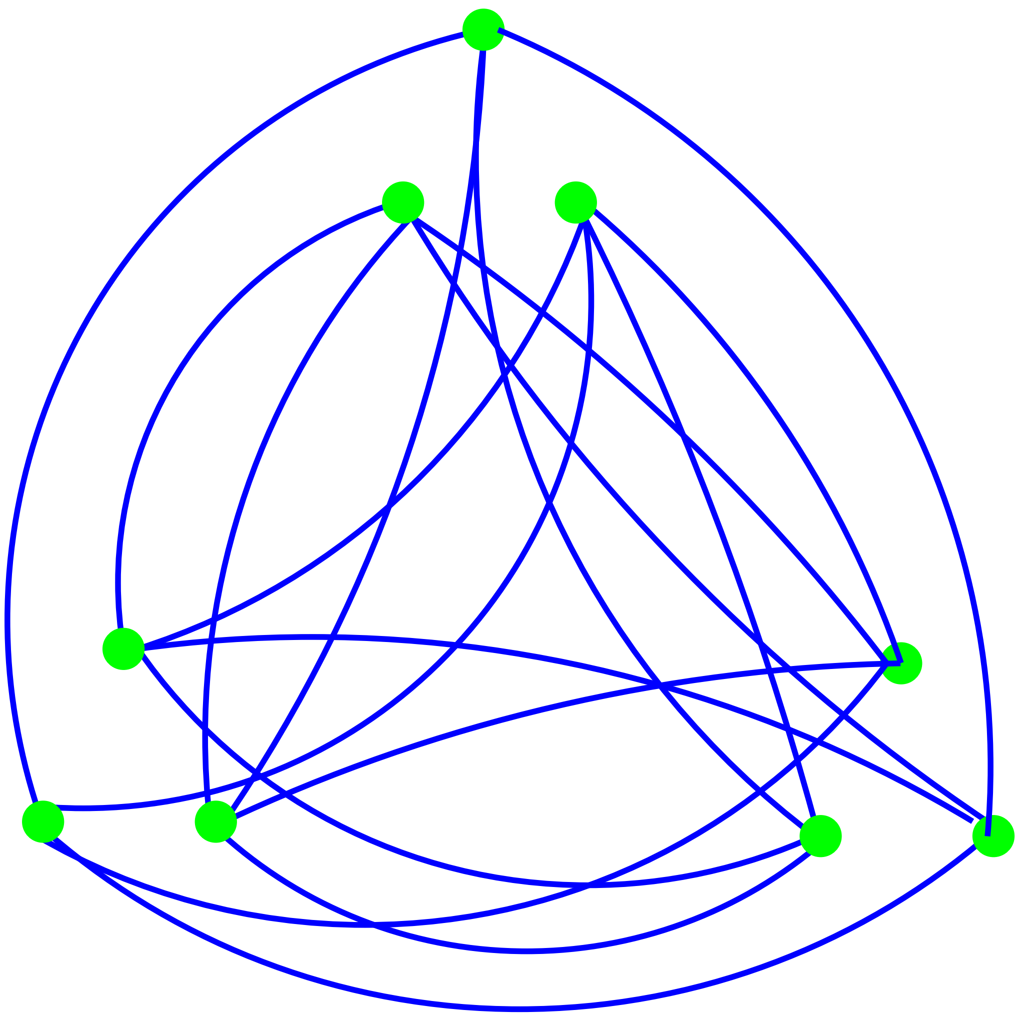


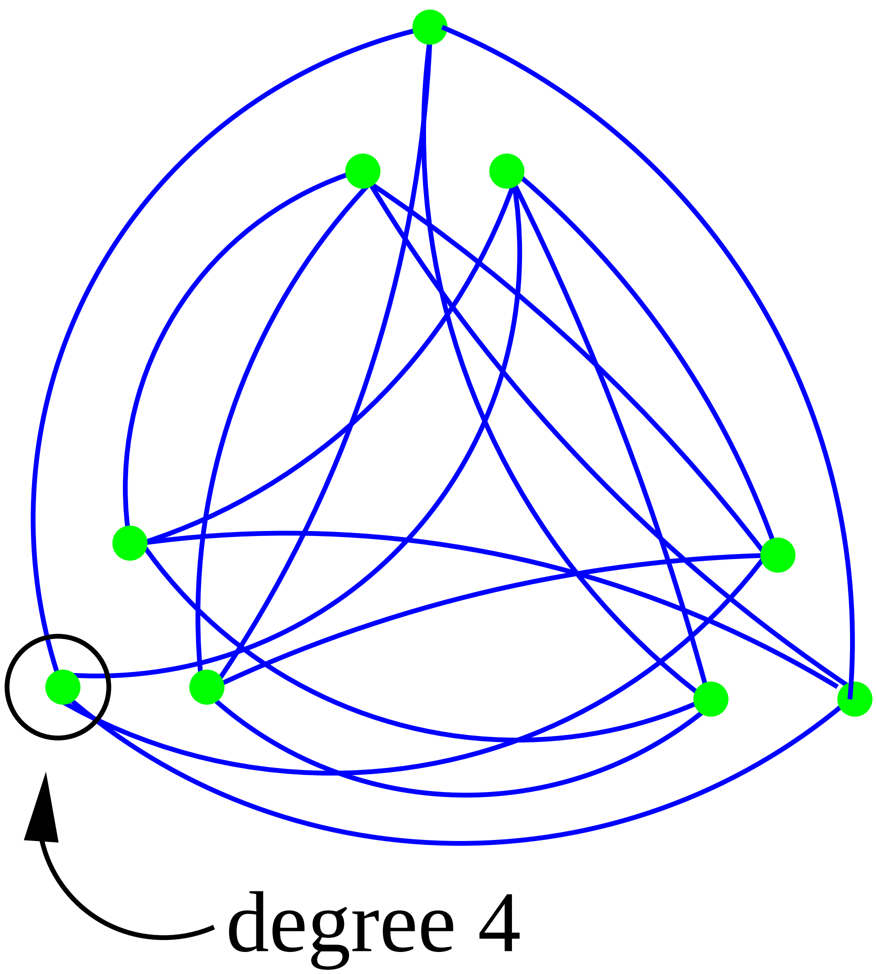











degree 4

Graph tensoring

- Parameters $[n_A \cdot n_B, d_A \cdot d_B, \mathsf{max}(\lambda_A, \lambda_B)]$

- Increases the size of the graph (dimension): **good**
- Maintains the second eigenvalue (distance): **good**
- But also increases the degree a lot (length): **bad**

- **Problem** for codes: Degree increases too much ($\Longrightarrow$ length of code increases faster than dimension).

- **Idea:** Remove some edges from $A \otimes B$ in a clever way.

Reducing the degree

- **Graph squaring:** $\mathcal{G}^2$ has the same nodes as $\mathcal{G}$, take all paths of length 2 as edges.

- August 2005: Rozenman and Vadhan presented a new operation **derandomized squaring** Ⓢ. This involves squaring a graph, and then *removing some edges* according to a second graph.

- Reduces degree at the cost of slightly worse expansion

- Can be seen as a projection of the zig-zag product

$$A \circledS (C^2) = P\Big[(A \text{ⓩ} C)^2\Big]$$

- We wanted to remove edges from the tensor product (without losing too much expansion)

- We can use this idea to come up with **derandomized tensoring**: Take the tensor product of two graphs, and remove edges according to a third **bipartite graph**.

Derandomized tensoring (1)

• A, B graphs with node sets $[n_A]$, $[n_B]$, degrees d_A, d_B.

• Assume **edge colorings**

$$\varphi_A : E(A) \to [d_A],$$

and likewise φ_B.

• For a Cayley graph: 1 color $\leftrightarrow$ 1 generator

• Suppose we have a bipartite graph C with d_A left nodes, and d_B right nodes.

• So there is a correspondance:

colors of $A \leftrightarrow$ left nodes of C
colors of $B \leftrightarrow$ right nodes of C

• $A \otimes_C B$: node set $[n_A] \times [n_B]$

• Edges: $(a, b) \sim (a', b') \iff$

$$\begin{aligned} a &\sim_A a' \\ b &\sim_B b' \\ \varphi_A(a, a') &\sim_C \varphi_B(b, b') \end{aligned}$$

Derandomized tensoring (2)

- $A \otimes_C B$:
  - Number of nodes $= n_A \cdot n_B$
  - Degree $= |\mathrm{Edges}(C)|$

- If C is biregular of left and right degrees ℓ, r:

$$\text{Degree} = d_A \cdot \ell = d_B \cdot r.$$

- If C is the **complete bipartite graph** then

$$A \otimes_C B = A \otimes B.$$

- **In terms of codes** this involves appending certain columns from the two generator matrices.

Expansion properties

• What are the expansion properties of this product?

Theorem. Suppose without loss of generality that $\lambda_B \leq \lambda_A$. Suppose also that C is biregular. Then

$$\lambda_{A \otimes_C B} \leq \max\Big(\lambda_A, \lambda_B, m(\lambda_A, \lambda_B, \lambda_C)\Big),$$

where we let

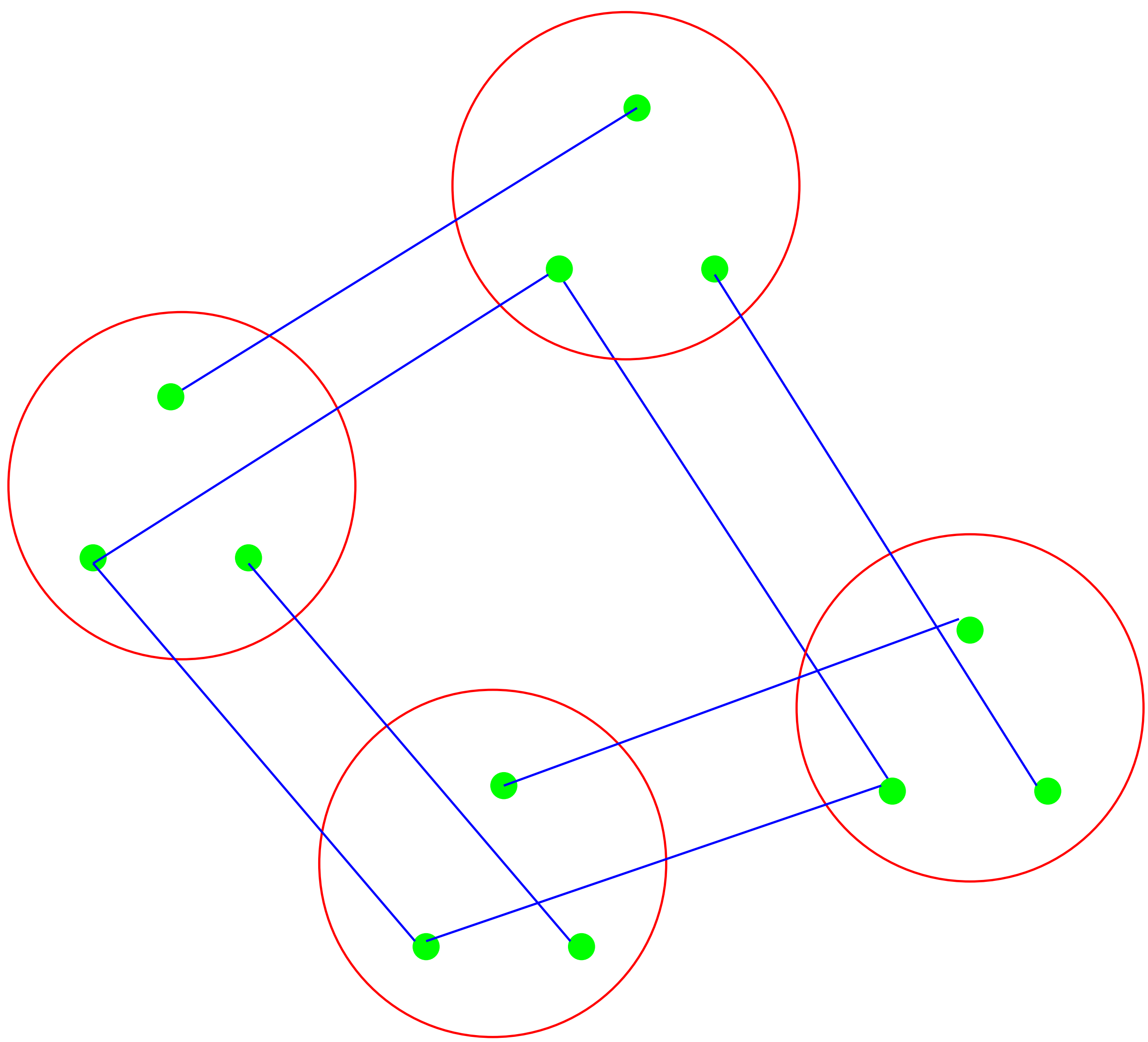
$$\begin{aligned} f(a,b,c) &= ab + c\sqrt{(1-a^2)(1-b^2)}, \\ g(b,c) &= \left(\frac{c^2}{b^2} - c^2 + 1\right)^{-1/2}, \\ m(a,b,c) &= f\Big(\min(a, g(b,c)), b, c\Big). \end{aligned}$$

• **Simpler case**: If $\lambda_A = \lambda_B$ then

$$\lambda_{A \otimes_C B} \leq \max\Big(\lambda_A, \lambda_A^2 + \lambda_C \cdot (1 - \lambda_A^2)\Big)$$

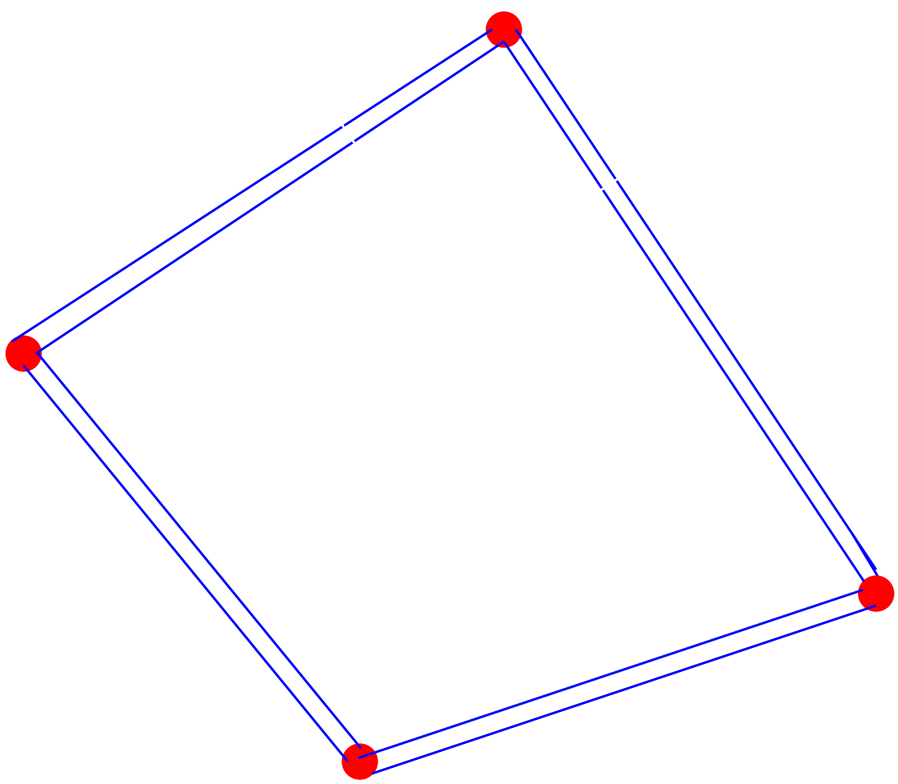
Projection

- The analysis is done by viewing $A \otimes_C B$ as a **projection** of a larger graph.



Projection

- The analysis is done by viewing $A \otimes_C B$ as a **projection** of a larger graph.



Proof of Theorem

• We view our graph over $[n_A] \times [n_B]$ as **projection** of a graph over $[n_A] \times [n_B] \times [\underbrace{d_A + d_B}_{d}]$.

• **Normal tensoring:** $A \otimes B = \widehat{A} \cdot \widehat{B}$, where

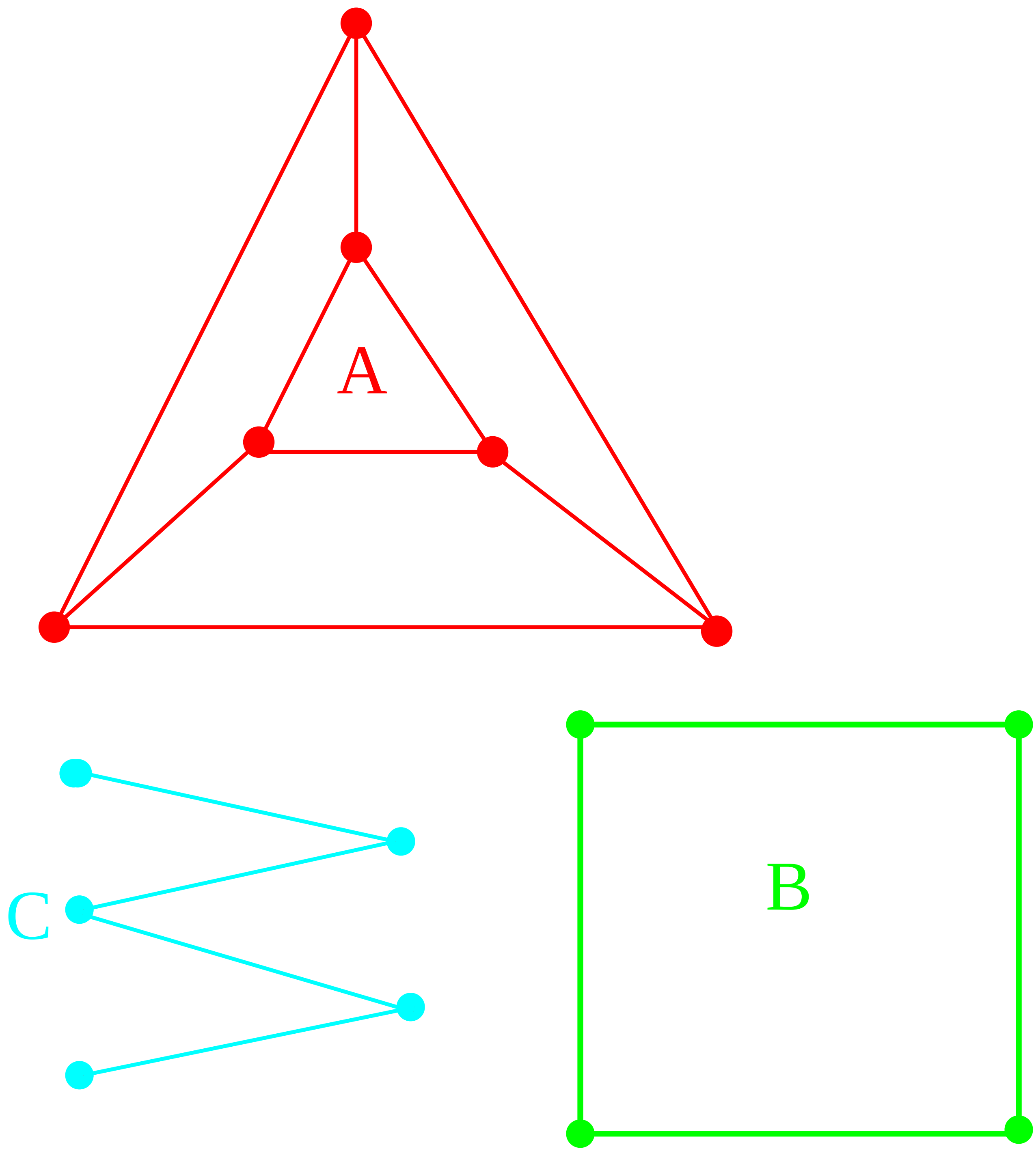
$$\widehat{A} = A \otimes Id(n_B)$$
$$\widehat{B} = Id(n_A) \otimes B$$

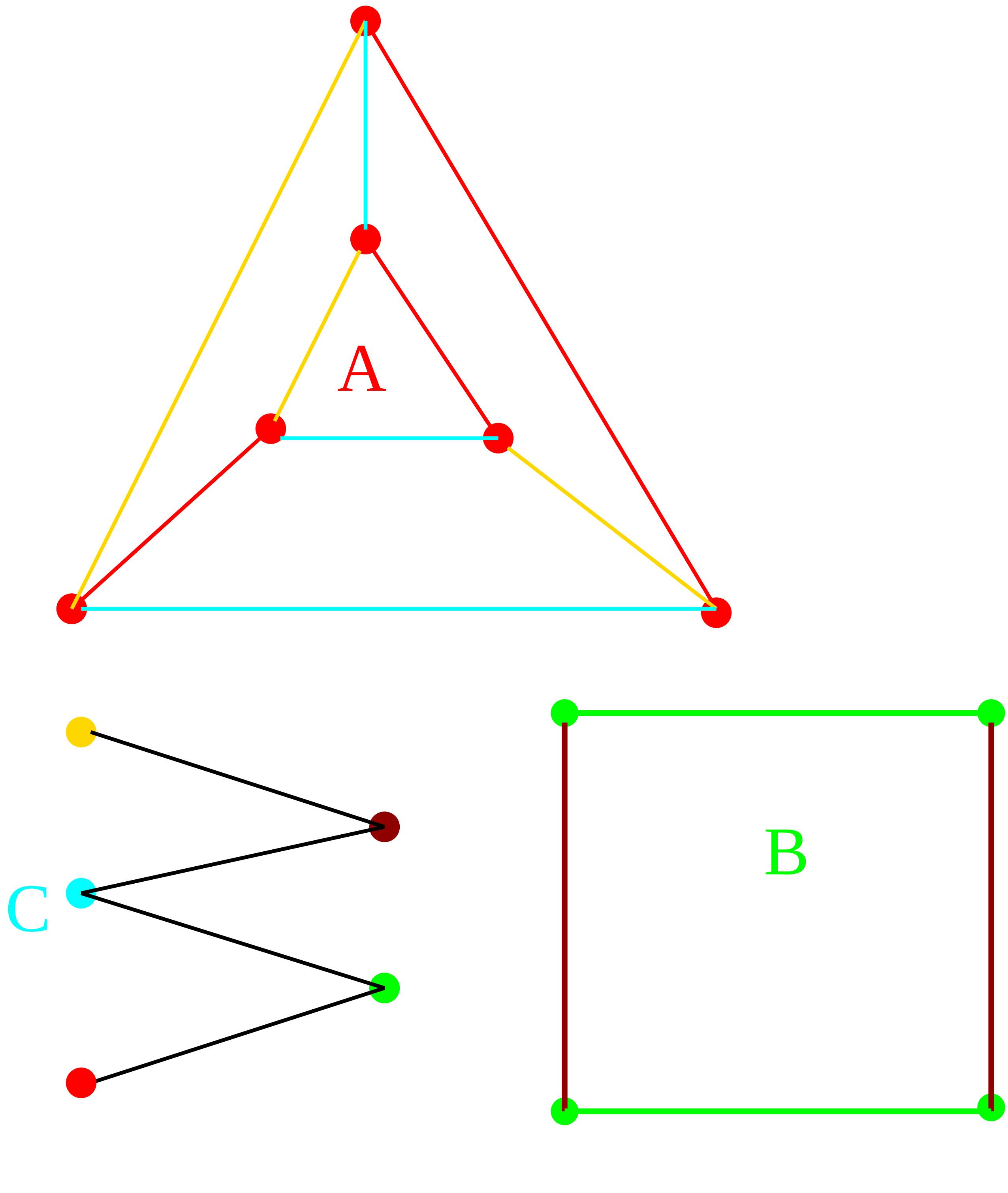
• **Derandomized tensoring:**

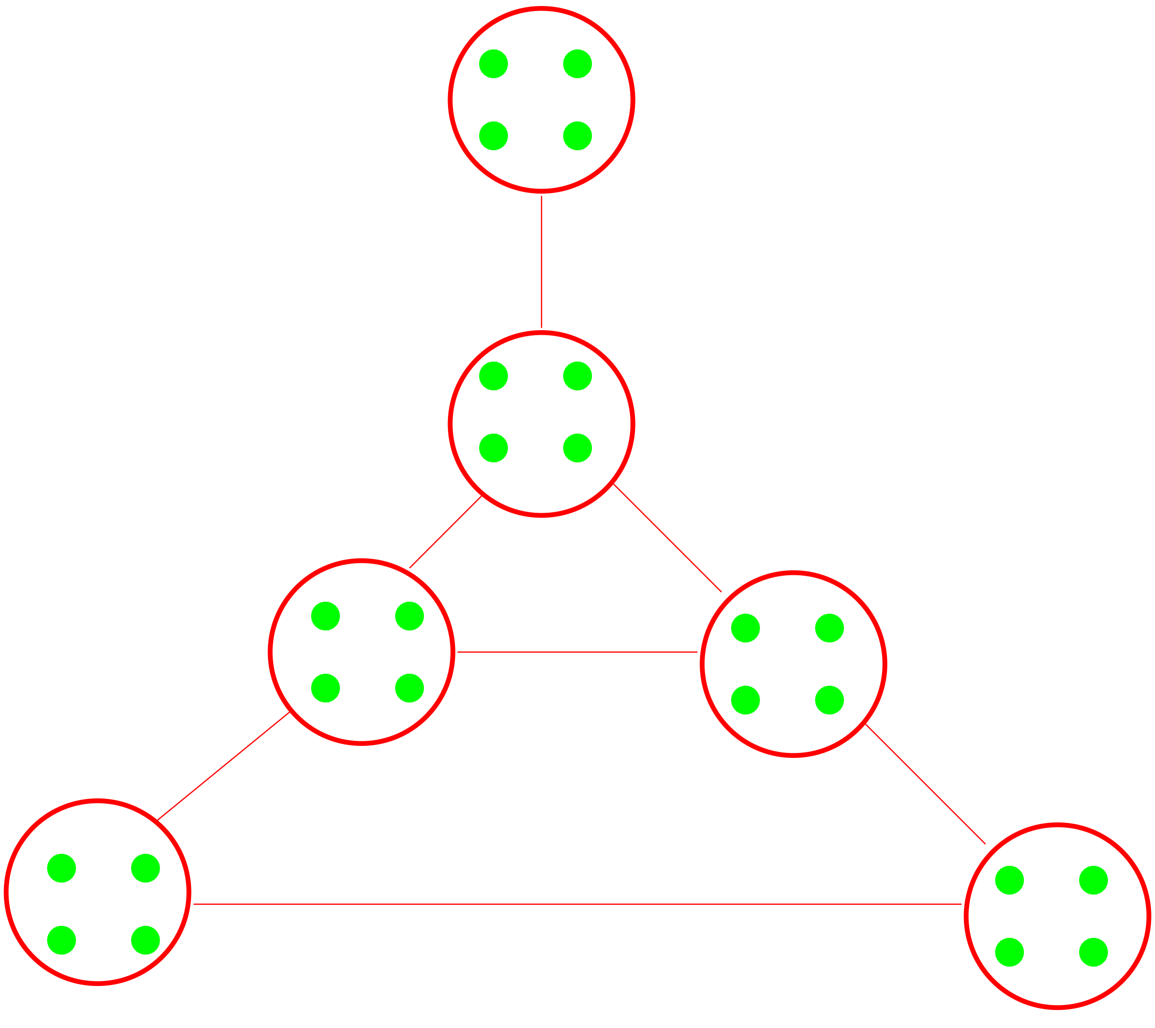
$$A \otimes_C B = \mathrm{Proj}[\widehat{X} \cdot \widehat{C} \cdot \widehat{X}],$$

where • $\widehat{X}$ depends on A and B,

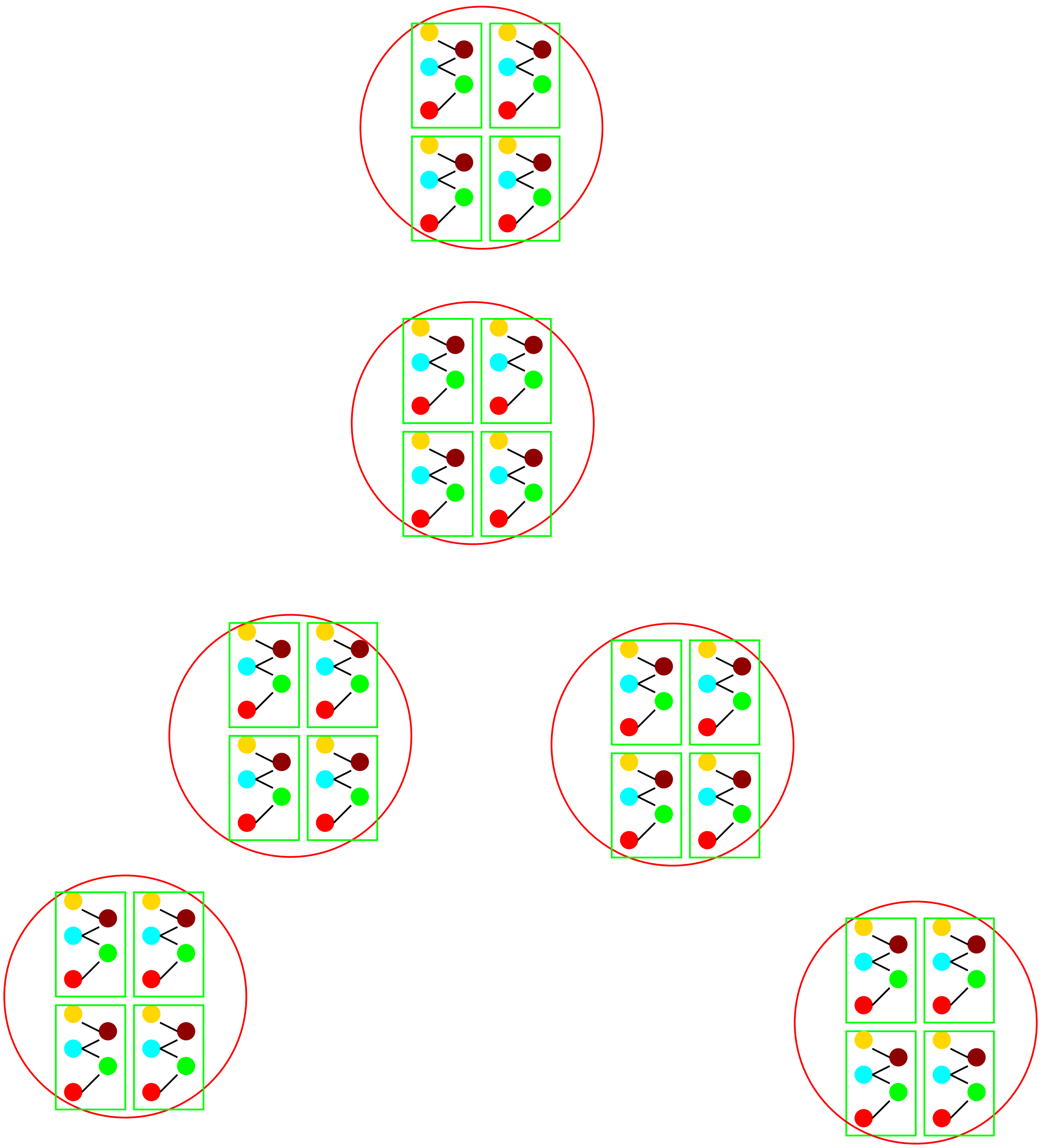
• $\widehat{C} = Id(n_A n_B) \otimes C$.

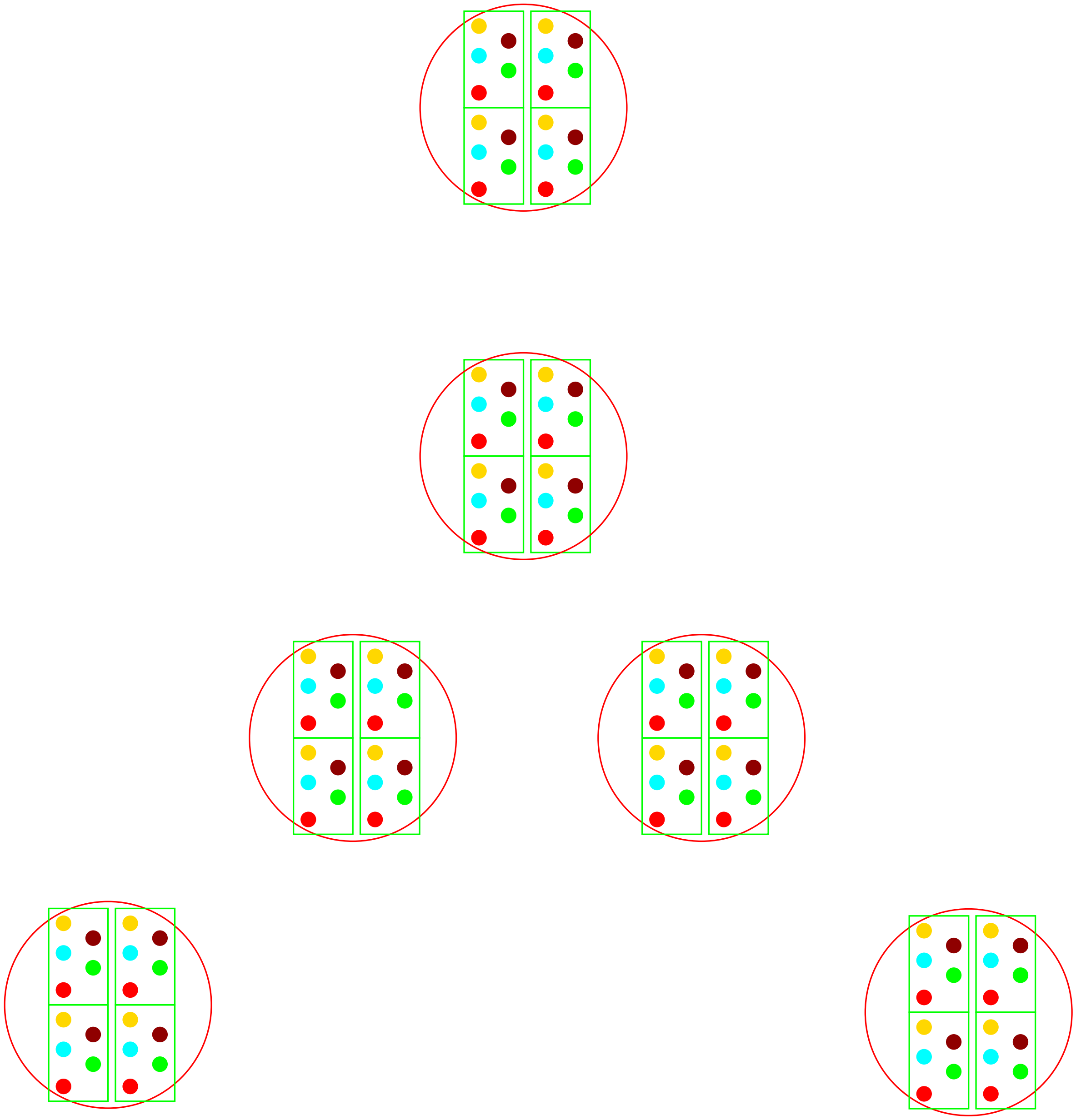

A
C
B

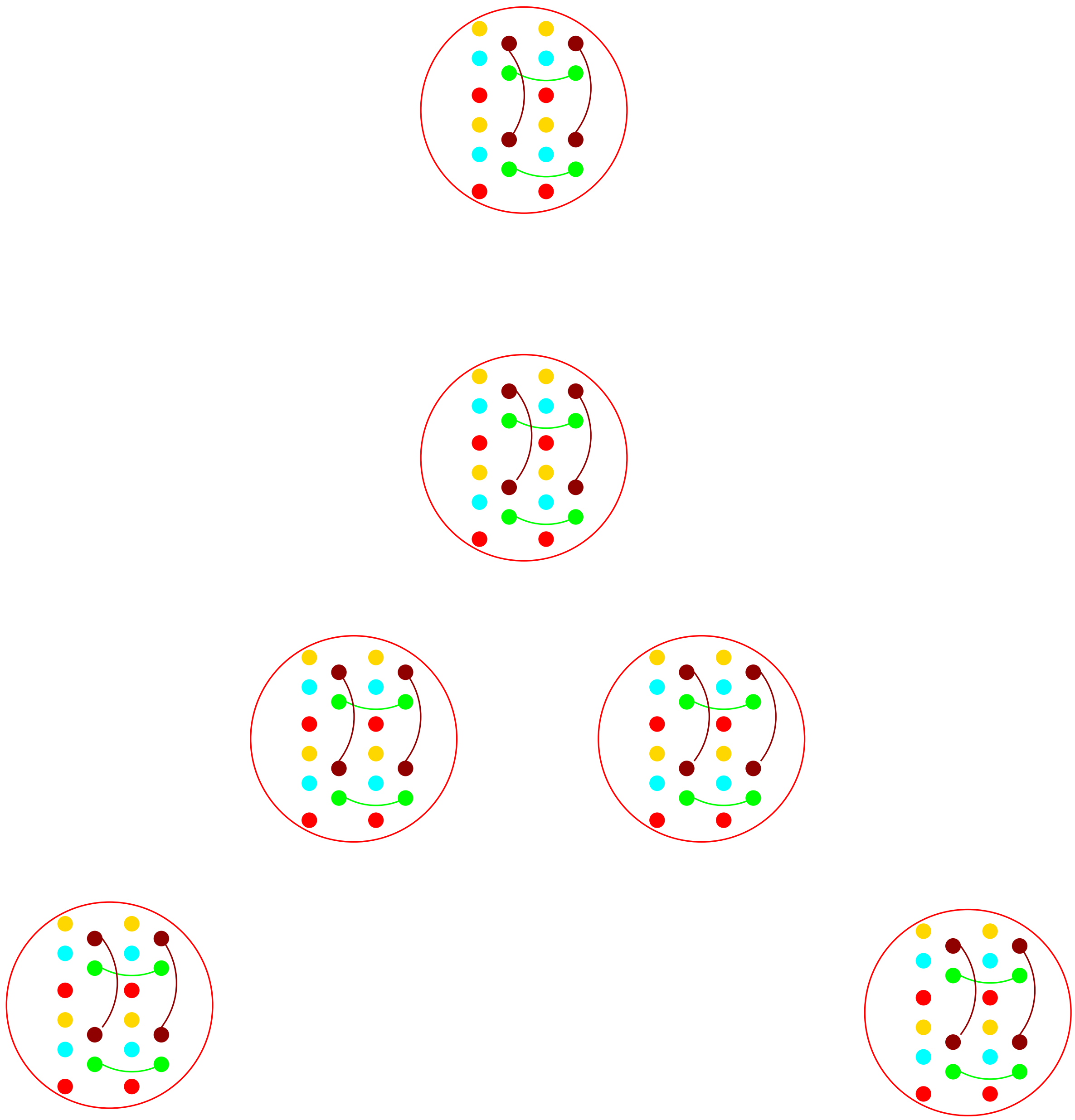

A
C
B

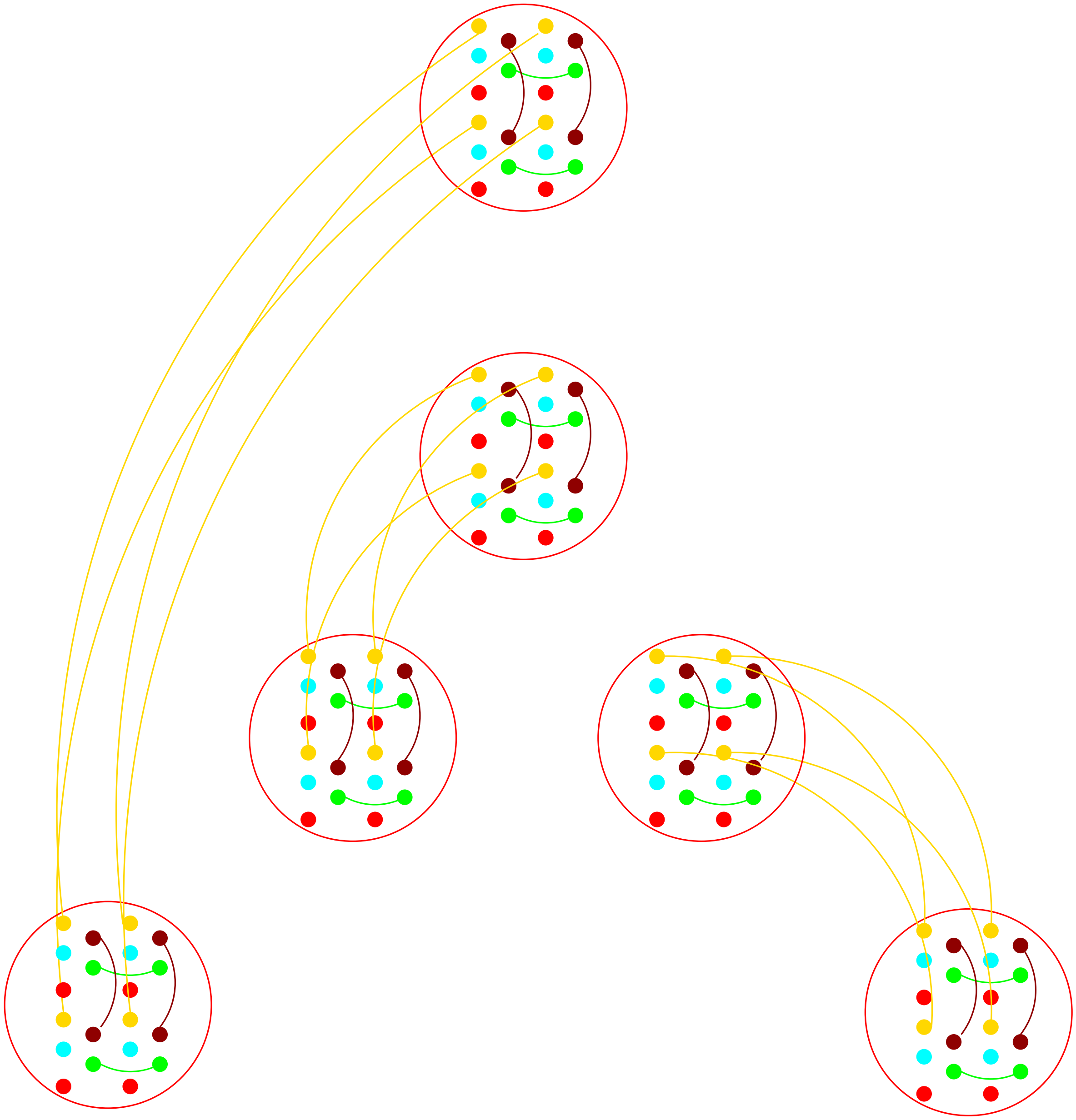


• Graph $\widehat{C}$

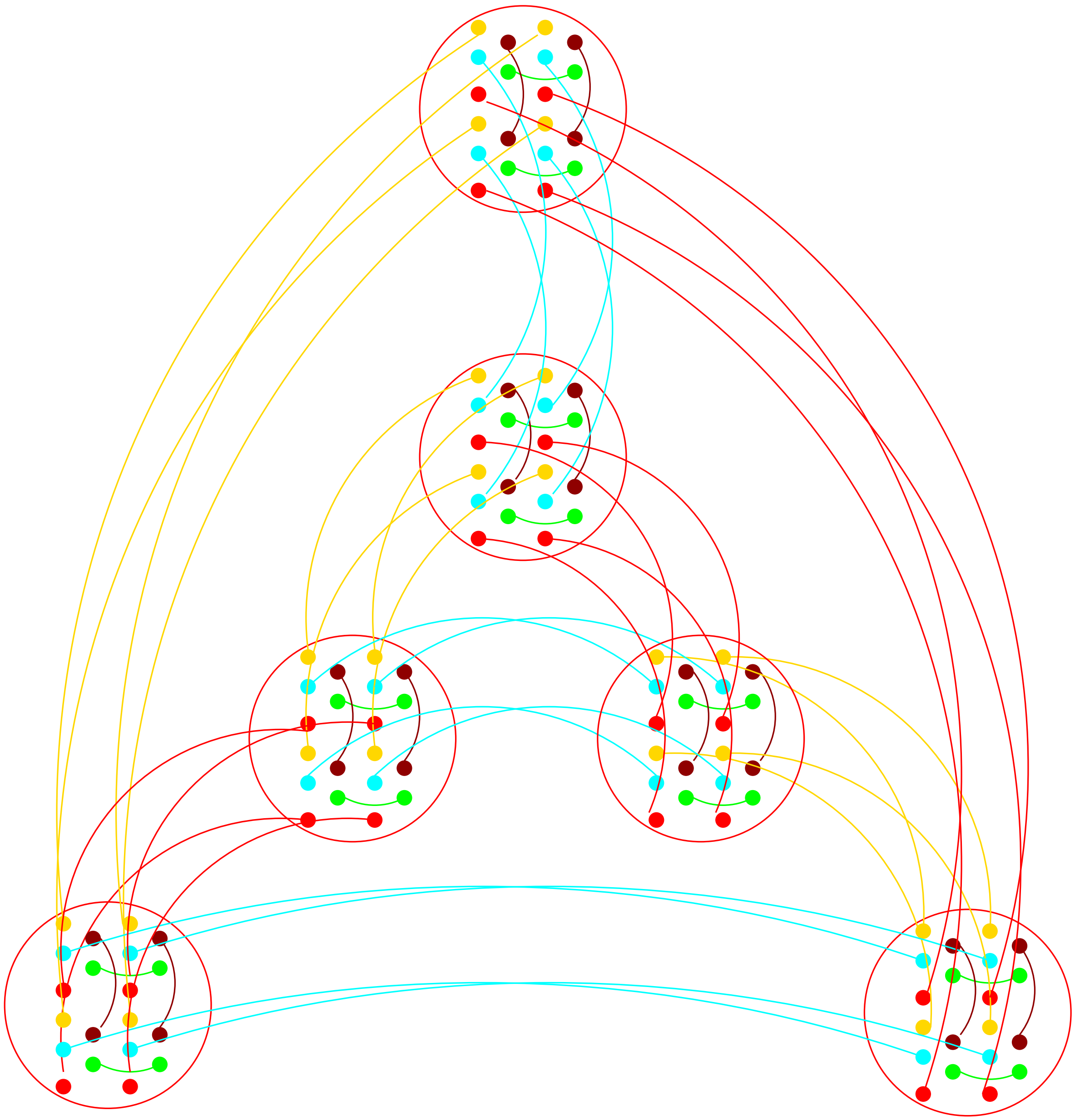








Graph $\widehat{X}$



Proof of Theorem

- **Lemma.** Let S be the space

$$S = (1_{n_A})^{\perp} \otimes (1_{n_B})^{\perp} \otimes 1_d^{\parallel}$$

The second eigenvalue of this projection is

$$\lambda\Big(\text{Proj}[\hat{X}\hat{C}\hat{X}]\Big) = \max_{x \in S} \frac{\Big|\langle \hat{X}\hat{C}\hat{X} \cdot x, x\rangle\Big|}{\langle x, x\rangle}.$$

- We decompose S into

$$\underbrace{\Big(1_{n_A}^{\perp} \otimes 1_{n_B}^{\parallel} \otimes 1_d^{\parallel}\Big)}_{S_1} \oplus \underbrace{\Big(1_{n_A}^{\parallel} \otimes 1_{n_B}^{\perp} \otimes 1_d^{\parallel}\Big)}_{S_2}$$

$$\oplus \underbrace{\Big(1_{n_A}^{\perp} \otimes 1_{n_B}^{\perp} \otimes 1_d^{\parallel}\Big)}_{S_3}.$$

- Show that

$$x_1 \in S_1 \Longrightarrow |\langle \hat{X}\hat{C}\hat{X} \cdot x, x\rangle| \leq \lambda_A \cdot \langle x, x\rangle$$
$$x_2 \in S_2 \Longrightarrow |\langle \hat{X}\hat{C}\hat{X} \cdot x, x\rangle| \leq \lambda_B \cdot \langle x, x\rangle$$
$$x_3 \in S_3 \Longrightarrow |\langle \hat{X}\hat{C}\hat{X} \cdot x, x\rangle| \leq m(\lambda_A, \lambda_B, \lambda_C) \cdot \langle x, x\rangle$$

- Deduce that if $x \in S$ then

$$\frac{|\langle \hat{X}\hat{C}\hat{X} \cdot x, x\rangle|}{\langle x, x\rangle} \leq \max\Big(\lambda_A, \lambda_B, m(\lambda_A, \lambda_B, \lambda_C)\Big)$$

Extensions

• This idea can be also be used to get a different analysis of the derandomized square

$$\lambda\Big(A \circledS C\Big) \leq \lambda_A^2 + \lambda_C \cdot (1 - \lambda_A^2)$$

• We can also create a derandomized zig-zag product

$$\lambda\Big(A \text{ⓩ}_C B\Big) \leq \lambda_A + \lambda_B + \lambda_B^2 + \lambda_C \cdot (1 - \lambda_B^2),$$

smaller degree than the original zig-zag product, at the cost of slightly worse expansion.

Conclusion

- There is a coding theoretic motivation behind finding graph products with good expansion properties and small degree.

- We can define derandomized version of known products, decreasing the degree a lot while only slightly worsening the expansion.

- The analysis is done by looking at the product as a projection of a larger graph, whose adjacency matrix we can express easily.

- These tools can be used to obtain bounds the expansion of other graph products.