Graph Isomorphism problem, Weisfeiler-Leman algortihm and coherent configurations

M. Muzychuk

Ben-Gurion University of the Negev,
Israel

10th PhD Summer School in Discrete Mathematics , Rogla
Slovenia, June 2022

Content

- Graph isomorphism problem and Weisfeiler-Leman algorithm.
- Coherent configurations and coherent (cellular) algebras.
- Association schemes.
- Cayley schemes and Schur rings.
- Schur rings and Cayley graphs isomorphism problem.

References. Coherent configurtion and algebras

[Hig70] D. G. Higman, Coherent configurations I, Rend. Sem. Mat. Univ. Padova 44(1970), 1-25

[Hig87] D. G. Higman,Coherent algebras, Linear Alg. and Its Applications 93(1987),209-239

[Wei76] B. Weisfeiler, On Construction and Identification of Graphs, LNM 558 (1976)

KRRT99] M. Klin, C. Rücker, G. Rücker and G. Tinhofer, Algebraic Combinatorics in Mahematical Chemistry. Methods and Algorithms. I. Permutation Groups and Coherent (Cellular) Algebras. Match (40), 1999, `shttp:\match.pmf.kg.ac.rs\electronic_versions\Match40\match40_7-138.pdf`

[CP] G. Chen and I. Ponomarenko, Lectures on coherent configurations (2019), `http://www.pdmi.ras.ru/~inp/ccNOTES.pdf`.

References. Association Schemes

[BI84] E. Bannai, T. Ito, Algebraic Combinatorics I: Association Schemes, Benjamin/Cummings, Menlo Park, CA, 1984.

[Ba04] R.A. Bailey, Association Schemes: Designed Experiments, Algebra and Combinatorics, Cambridge University Press, Cambridge, 2004

[BCN89] A.E. Brouwer, A.M. Cohen, A. Neumaier, Distance-Regular Graphs, Springer-Verlag, Berlin, 1989

[Z96] P.-H. Zieschang, An algebraic approach to association schemes, Springer-Verlag, Berlin,1996

[Z05] P.-H. Zieschang, Theory of Association Schemes, Springer-Verlag, Berlin, 2005.

Binary relations

Let $R, S \subseteq \Omega^2$ be binary relations. Then

- $S^* := \{(\alpha, \beta) \mid (\beta, \alpha) \in S\}$;
- S is symmetric (antisymmetric) if $S = S^*$ ($S \cap S^* = \emptyset$ resp.);
- $\alpha S := \{\beta \mid (\alpha, \beta) \in S\}, S\alpha := \alpha S^*$;
- $D(S) := \{\alpha \in \Omega \mid \alpha S \neq \emptyset\}, R(S) := D(S^*)$;
- $RS = \{(\alpha, \beta) \mid \alpha R \cap S\beta \neq \emptyset\}$;
- $R^+ = \bigcup_{i=1}^{\infty} R^i$ is the transitive closure of R;
- $1_\Omega := \{(\omega, \omega) \mid \omega \in \Omega\}$

Each permutation $g \in \mathrm{Sym}(\Omega)$ is considered as a binary relation. Thus $\alpha g = \{\alpha^g\}$ and $g^* = g^{-1}$.

Partitions.

- $\mathcal{P} \vdash \Omega$ means that $\mathcal{P}$ is a partition of Ω.
- $\mathcal{P} \sqsubseteq \mathcal{C} \iff \mathcal{C}$ is a refinement of $\mathcal{P}$ (in particular, $|\mathcal{P}| \leq |\mathcal{C}|$);
- Lattice operations are denoted as $\mathcal{P} \vee \mathcal{C}$ and $\mathcal{P} \wedge \mathcal{C}$;
- if $\mathcal{P} \vdash \Omega$ then $\mathcal{P}^{\cup}$ denotes the set of all possible unions of elements in $\mathcal{P}$;
- $\mathcal{C} \vdash \Omega^2 \implies \mathcal{C}^* := \{C^* \mid C \in \mathcal{C}\}$;

Graphs.

In what follows graph is a pair $\Gamma = (\Omega, E)$ where Ω is a finite set of vertices and $E \subset \Omega \times \Omega$ is the set of arcs; Γ is undirected if $E^* = E$.

Graphs.

In what follows graph is a pair $\Gamma = (\Omega, E)$ where Ω is a finite set of vertices and $E \subset \Omega \times \Omega$ is the set of arcs; Γ is undirected if $E^* = E$.

Definition.

Graphs $\Gamma_1 = (\Omega_1, E_1)$ and $\Gamma_2 = (\Omega_2, E_2)$ are called isomorphic, $\Gamma_1 \cong \Gamma_2$, if there is a bijection $f : \Omega_1 \to \Omega_2$ such that

$$\forall \alpha_1, \beta_1 \in \Omega_1 : \quad (\alpha_1^f, \beta_1^f) \in E_2 \quad \Leftrightarrow \quad (\alpha_1, \beta_1) \in E_1.$$

Such a bijection is called an isomorphism from Γ_1 to Γ_2; the set of all of them is denoted by $\mathrm{Iso}(\Gamma_1, \Gamma_2)$.The set $\mathrm{Iso}(\Gamma_1, \Gamma_1)$ is known as the automorphism group of Γ_1, notation $\mathrm{Aut}(\Gamma_1)$.

Graphs.

In what follows graph is a pair $\Gamma = (\Omega, E)$ where Ω is a finite set of vertices and $E \subset \Omega \times \Omega$ is the set of arcs; Γ is undirected if $E^* = E$.

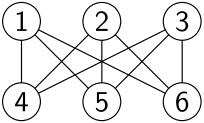
Definition.

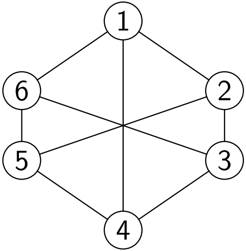
Graphs $\Gamma_1 = (\Omega_1, E_1)$ and $\Gamma_2 = (\Omega_2, E_2)$ are called isomorphic, $\Gamma_1 \cong \Gamma_2$, if there is a bijection $f : \Omega_1 \to \Omega_2$ such that

$$\forall \alpha_1, \beta_1 \in \Omega_1 : \quad (\alpha_1^f, \beta_1^f) \in E_2 \quad \Leftrightarrow \quad (\alpha_1, \beta_1) \in E_1.$$

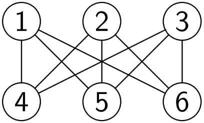
Such a bijection is called an isomorphism from Γ_1 to Γ_2; the set of all of them is denoted by $\mathrm{Iso}(\Gamma_1, \Gamma_2)$.The set $\mathrm{Iso}(\Gamma_1, \Gamma_1)$ is known as the automorphism group of Γ_1, notation $\mathrm{Aut}(\Gamma_1)$.

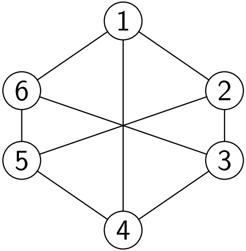
Example





Example

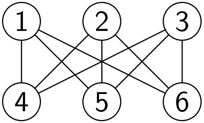


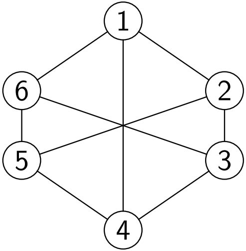


An isomorphism

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 5 & 2 & 4 & 6 \end{pmatrix}$$

Example





An isomorphism

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 5 & 2 & 4 & 6 \end{pmatrix}$$

$$\mathrm{Aut}(\Gamma) = (S_3 \times S_3).S_2.$$

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.
- An exhaustive search of all the possible bijections runs in exponential time $O(n!)$.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.
- An exhaustive search of all the possible bijections runs in exponential time $O(n!)$.
- At present it is not known whether GI$\in$**P**.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.
- An exhaustive search of all the possible bijections runs in exponential time $O(n!)$.
- At present it is not known whether GI$\in$**P**.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.
- An exhaustive search of all the possible bijections runs in exponential time $O(n!)$.
- At present it is not known whether GI$\in$**P**.

It does not matter whether the graphs Γ and Γ' are

-undirected, directed, regular, colored etc.

The Graph Isomorphism Problem (GI).

GI is to find the computational complexity of the problem:

given two graphs Γ_1 and Γ_2 test whether or not $\Gamma_1 \cong \Gamma_2$.

- Given graphs Γ_1 and Γ_2 of order n, and a bijection $f : \Omega_1 \to \Omega_2$ one can test in time $O(n^2)$ whether $f \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.
- Therefore GI$\in$**NP**.
- An exhaustive search of all the possible bijections runs in exponential time $O(n!)$.
- At present it is not known whether GI$\in$**P**.

It does not matter whether the graphs Γ and Γ' are

-undirected, directed, regular, colored etc.

Theorem (L.Babai, 2015).

The isomorphism of n-vertex graphs can be tested in time $\exp((\log n)^c)$.

Some problems equivalent to the GI (R.Mathon,1979).

The following problems are equivalent to the GI:

- ICOUNT: given Γ and Γ' find $|\,\mathrm{Iso}(\Gamma, \Gamma')|$,

Some problems equivalent to the GI (R.Mathon,1979).

The following problems are equivalent to the GI:

- ICOUNT: given Γ and Γ' find $|\mathrm{Iso}(\Gamma, \Gamma')|$,
- ACOUNT: given Γ find $|\mathrm{Aut}(\Gamma)|$,

Some problems equivalent to the GI (R.Mathon,1979).

The following problems are equivalent to the GI:

- ICOUNT: given Γ and Γ' find $|\mathrm{Iso}(\Gamma, \Gamma')|$,
- ACOUNT: given Γ find $|\mathrm{Aut}(\Gamma)|$,
- AGEN: given Γ find generators of the group $\mathrm{Aut}(\Gamma)$,

Some problems equivalent to the GI (R.Mathon,1979).

The following problems are equivalent to the GI:

- ICOUNT: given Γ and Γ' find $|\,\mathrm{Iso}(\Gamma, \Gamma')|$,
- ACOUNT: given Γ find $|\,\mathrm{Aut}(\Gamma)|$,
- AGEN: given Γ find generators of the group $\mathrm{Aut}(\Gamma)$,
- APART: given Γ find $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$.

Some problems equivalent to the GI (R.Mathon,1979).

The following problems are equivalent to the GI:

- ICOUNT: given Γ and Γ' find $|\mathrm{Iso}(\Gamma, \Gamma')|$,
- ACOUNT: given Γ find $|\mathrm{Aut}(\Gamma)|$,
- AGEN: given Γ find generators of the group $\mathrm{Aut}(\Gamma)$,
- APART: given Γ find $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$.

Isomorphism problem for colored graphs.

Definition.

A triple $\Gamma = (\Omega, Y, c)$ where $c : \Omega^2 \to Y$ is a surjection, is called a colored graph with the coloring function c and color classes $c^{-1}(y)$, $y \in Y$. Each colored graph determines a partition $\mathcal{C} := \{c^{-1}(y) \mid y \in Y\}$ of Ω^2.

Two colored graphs $\Gamma = (\Omega, Y, c)$ and $\Gamma' = (\Delta, Z, d)$ are isomorphic iff there exist bijections $f : \Omega \to \Delta, \phi : Y \to Z$ s.t.

$$d(\alpha^f, \beta^f) = c(\alpha, \beta)^\phi.$$

Isomorphism problem for colored graphs.

Definition.

A triple $\Gamma = (\Omega, Y, c)$ where $c : \Omega^2 \to Y$ is a surjection, is called a colored graph with the coloring function c and color classes $c^{-1}(y)$, $y \in Y$. Each colored graph determines a partition $\mathcal{C} := \{c^{-1}(y) \,|\, y \in Y\}$ of Ω^2.

Two colored graphs $\Gamma = (\Omega, Y, c)$ and $\Gamma' = (\Delta, Z, d)$ are isomorphic iff there exist bijections $f : \Omega \to \Delta, \phi : Y \to Z$ s.t.

$$d(\alpha^f, \beta^f) = c(\alpha, \beta)^\phi.$$

Notice that ϕ is uniquely determined by f. For this reason we define $f^* := \phi$.

The adjacency matrix $A(\Gamma)$ of $\Gamma = (\Omega, Y, c)$ is defined as follows:

$$(A(\Gamma))_{\alpha,\beta} = c(\alpha, \beta), \alpha, \beta \in \Omega.$$

Isomorphism problem for colored graphs.

We also set $\mathrm{Iso}(\Omega, Y, c)$ to be the group of all isomorphisms from (Ω, Y, c) to itself and $\mathrm{Aut}(\Omega, Y, c)$ to be the normal subgroup of $\mathrm{Iso}(\Omega, Y, c)$ which does not permute the colors (that is $f^* = 1_Y$).

Proposition

Let (Ω, Y, c) be a colored graph and $\mathcal{C} := \{c^{-1}(y) \,|\, y \in Y\}$ be the corresponding partition. Then

$$\mathrm{Iso}(\Omega, Y, c) = \{g \in \mathrm{Sym}(\Omega) \,|\, \mathcal{C}^g = \mathcal{C}\},$$
$$\mathrm{Aut}(\Omega, Y, c) = \{g \in \mathrm{Sym}(\Omega) \,|\, \forall_{C \in \mathcal{C}} C^g = C\}.$$

Theorem.

Isomorphism problem for colored graphs is polynomially equivalent to GI.

Cayley Graphs and their Isomorphisms.

A Cayley graph over a finite group H defined by a connection set $S \subseteq H$ has H as a set of nodes and arc set

$$\mathrm{Cay}(H, S) := \{(x, y) \,|\, xy^{-1} \in S\}$$

. A circulant graph is a Cayley graph over a cyclic group.

Definition

Two Cayley graphs $\mathrm{Cay}(H, S)$ and $\mathrm{Cay}(K, T)$ are Cayley isomorphic if there exists a group isomorphism $f : H \to K$ which is a graph isomoprhism too, that is

$$\mathrm{Cay}(H, S)^f = \mathrm{Cay}(K, T) \iff S^f = T.$$

The graphs $\mathrm{Cay}(\{\pm 1\}, \mathbb{Z}_5)$ and $\mathrm{Cay}(\{\pm 2\}, \mathbb{Z}_5)$ are Cayley isomorphic.

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H,S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.
- Now $f \in \mathrm{Iso}(\Gamma, \mathrm{Cay}(H, S))$:

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.
- Now $f \in \mathrm{Iso}(\Gamma, \mathrm{Cay}(H, S))$:

  $(\alpha, \beta) \in E \ \Leftrightarrow$

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.
- Now $f \in \mathrm{Iso}(\Gamma, \mathrm{Cay}(H, S))$:

  $(\alpha, \beta) \in E \ \Leftrightarrow \ (\omega^{\bar{\alpha}}, \omega^{\bar{\beta}}) \in E \ \Leftrightarrow$

Cayley representations of graphs.

An automorphism of a Cayley graph $\mathrm{Cay}(H, S)$ contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff $\mathrm{Aut}(\Gamma)$ contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.
- Now $f \in \mathrm{Iso}(\Gamma, \mathrm{Cay}(H, S))$:

  $$(\alpha, \beta) \in E \ \Leftrightarrow \ (\omega^{\bar{\alpha}}, \omega^{\bar{\beta}}) \in E \ \Leftrightarrow \ (\omega^{\bar{\alpha}\bar{\beta}^{-1}}, \omega) \in E \ \Leftrightarrow$$

Cayley representations of graphs.

An automorphism of a Cayley graph Cay(H, S) contains a regular subgroup H_R consisting of right translations $h_R, h \in H$:

$$x^{h_R} = xh, x \in H.$$

Theorem (Sabidussi)

A graph $\Gamma = (\Omega, E)$ is isomorphic to a Cayley graph over a group H iff Aut(Γ) contains a regular subgroup isomorphic to H.

Proof.

- Pick a base point $\omega \in \Omega$.
- Define a bijection $f : \Omega \to H$, $\alpha \mapsto \bar{\alpha}$ where $\omega^{\bar{\alpha}} = \alpha$.
- Set $S = \{\bar{\alpha} : \ \alpha \in E\omega\}$.
- Now $f \in \mathrm{Iso}(\Gamma, \mathrm{Cay}(H, S))$:

$$(\alpha, \beta) \in E \ \Leftrightarrow \ (\omega^{\bar{\alpha}}, \omega^{\bar{\beta}}) \in E \ \Leftrightarrow \ (\omega^{\bar{\alpha}\bar{\beta}^{-1}}, \omega) \in E \ \Leftrightarrow \ \bar{\alpha}\bar{\beta}^{-1} \in S.$$

Isomorphism problems for Cayley graphs.

Given $\Gamma = \mathrm{Cay}(H, S)$ and $\Gamma' = \mathrm{Cay}(H, S')$:

- IMAP: find $f \in \mathrm{Iso}(\Gamma, \Gamma')$ (if it exists),
- ICOUNT: find $|\,\mathrm{Iso}(\Gamma, \Gamma')|$,
- ACOUNT:find $|\,\mathrm{Aut}(\Gamma)|$,
- AGEN: find generators of the group $\mathrm{Aut}(\Gamma)$,
- CGR: given a graph Θ find whether it's a Cayley graph over a group H.

Isomorphism problem for finite groups.

Construction. Let K be a finite group.

Define a graph $\Gamma(K)$ with vertex set $K \times K$ and edges:

$$(a, b) \sim (c, d) \iff a = c \vee b = d \vee ab = cd.$$

Isomorphism problem for finite groups.

Construction. Let K be a finite group.

Define a graph $\Gamma(K)$ with vertex set $K \times K$ and edges:

$$(a, b) \sim (c, d) \iff a = c \vee b = d \vee ab = cd.$$

Theorem (E. Moorhouse) $K_1 \cong K_2 \iff \Gamma(K_1) \cong \Gamma(K_2)$.

Isomorphism problem for finite groups.

Construction. Let K be a finite group.

Define a graph $\Gamma(K)$ with vertex set $K \times K$ and edges:

$$(a, b) \sim (c, d) \iff a = c \vee b = d \vee ab = cd.$$

Theorem (E. Moorhouse) $K_1 \cong K_2 \iff \Gamma(K_1) \cong \Gamma(K_2)$.

Exercise. Prove that $\Gamma(K)$ is a Cayley graph over $K \times K$.

Exercise. Prove that $\Gamma(\mathbb{Z}_4) \not\cong \Gamma(\mathbb{Z}_2 \times \mathbb{Z}_2)$.

$\mathbb{Z}_4 \to$

0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

$\mathbb{Z}_2 \times \mathbb{Z}_2 \to$

0	1	2	3
1	0	3	2
2	3	0	1
3	2	1	0

Isomorphism problem for finite groups.

Construction. Let K be a finite group.

Define a graph $\Gamma(K)$ with vertex set $K \times K$ and edges:

$$(a, b) \sim (c, d) \iff a = c \vee b = d \vee ab = cd.$$

Theorem (E. Moorhouse) $K_1 \cong K_2 \iff \Gamma(K_1) \cong \Gamma(K_2)$.

Exercise. Prove that $\Gamma(K)$ is a Cayley graph over $K \times K$.

Exercise. Prove that $\Gamma(\mathbb{Z}_4) \ncong \Gamma(\mathbb{Z}_2 \times \mathbb{Z}_2)$.

$\mathbb{Z}_4 \to$

0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

$\mathbb{Z}_2 \times \mathbb{Z}_2 \to$

0	1	2	3
1	0	3	2
2	3	0	1
3	2	1	0

The isomorphism of groups of order n can be tested in time $n^{O(\log n)}$.

Naive vertex classification.

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

- To find $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$ put vertices α and β in the same class iff $d_\Gamma(\alpha) = d_\Gamma(\beta)$.

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

- To find $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$ put vertices α and β in the same class iff $d_\Gamma(\alpha) = d_\Gamma(\beta)$.
- Iteratively, put vertices α and β in the same class iff $d_\Gamma(\alpha, C) = d_\Gamma(\beta, C)$ for all color classes C.

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

- To find $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$ put vertices α and β in the same class iff $d_\Gamma(\alpha) = d_\Gamma(\beta)$.
- Iteratively, put vertices α and β in the same class iff $d_\Gamma(\alpha, C) = d_\Gamma(\beta, C)$ for all color classes C.

Comments.

- The algorithm correctly finds $\mathrm{Orb}(\mathrm{Aut}(\Gamma))$ for the class of trees (G.Tinhofer, 1985), for almost all graphs (L.Babai, P.Erdös, S.Selkow, 1980).

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

- To find Orb(Aut(Γ)) put vertices α and β in the same class iff $d_\Gamma(\alpha) = d_\Gamma(\beta)$.
- Iteratively, put vertices α and β in the same class iff $d_\Gamma(\alpha, C) = d_\Gamma(\beta, C)$ for all color classes C.

Comments.

- The algorithm correctly finds Orb(Aut(Γ)) for the class of trees (G.Tinhofer, 1985), for almost all graphs (L.Babai, P.Erdös, S.Selkow, 1980).
- The algorithm fails when Γ is a regular graphs and the group Aut(Γ) is intransitive.

Naive vertex classification.

Vertex partition by valences.

Denote by $d_\Gamma(\alpha)$ the valency of the vertex α in the graph Γ;the valency of α in a color class is denoted by $d_\Gamma(\alpha, C)$.

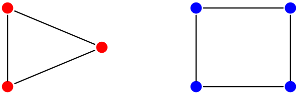
- To find Orb(Aut(Γ)) put vertices α and β in the same class iff $d_\Gamma(\alpha) = d_\Gamma(\beta)$.
- Iteratively, put vertices α and β in the same class iff $d_\Gamma(\alpha, C) = d_\Gamma(\beta, C)$ for all color classes C.

Comments.

- The algorithm correctly finds Orb(Aut(Γ)) for the class of trees (G.Tinhofer, 1985), for almost all graphs (L.Babai, P.Erdös, S.Selkow, 1980).
- The algorithm fails when Γ is a regular graphs and the group Aut(Γ) is intransitive.

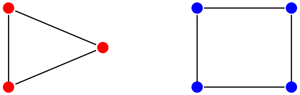
The Weisfeiler-Leman algorithm, 1968.

No automorphism moves red points to blue ones.



The Weisfeiler-Leman algorithm, 1968.

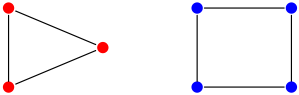
No automorphism moves red points to blue ones.



To distinguish vertices we need to color edges of Γ.

The Weisfeiler-Leman algorithm, 1968.

No automorphism moves red points to blue ones.

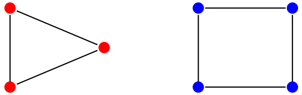


To distinguish vertices we need to color edges of Γ.

Algorithm. Set $\mathcal{C} = \{1_\Omega\} \cup \{E\} \cup \{(\Omega \times \Omega) \setminus E\}$.

The Weisfeiler-Leman algorithm, 1968.

No automorphism moves red points to blue ones.



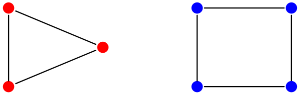
To distinguish vertices we need to color edges of Γ.

Algorithm. Set $\mathcal{C} = \{1_\Omega\} \cup \{E\} \cup \{(\Omega \times \Omega) \setminus E\}$.

- For all $(\alpha, \beta) \in \Omega \times \Omega$ and $R, S \in \mathcal{C}$ find the intersection number
$$c(\alpha, \beta; R, S) = |\alpha R \cap S\beta|.$$
- Build a new partition $\text{Ы}(\mathcal{C})$ by putting (α, β) and (α', β') to the same class of $\text{Ы}(\mathcal{C})$ if $|\alpha R \cap S\beta| = |\alpha' R \cap S\beta'|$ for all $R, S \in \mathcal{C}$.

The Weisfeiler-Leman algorithm, 1968.

No automorphism moves red points to blue ones.



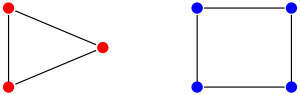
To distinguish vertices we need to color edges of Γ.

Algorithm. Set $\mathcal{C} = \{1_\Omega\} \cup \{E\} \cup \{(\Omega \times \Omega) \setminus E\}$.

- For all $(\alpha, \beta) \in \Omega \times \Omega$ and $R, S \in \mathcal{C}$ find the intersection number
$$c(\alpha, \beta; R, S) = |\alpha R \cap S\beta|.$$
- Build a new partition Ы$(\mathcal{C})$ by putting (α, β) and (α', β') to the same class of Ы$(\mathcal{C})$ if $|\alpha R \cap S\beta| = |\alpha' R \cap S\beta'|$ for all $R, S \in \mathcal{C}$.
- Repeat the procedure till $|\mathcal{C}|$ stops to increase.

The Weisfeiler-Leman algorithm, 1968.

No automorphism moves red points to blue ones.

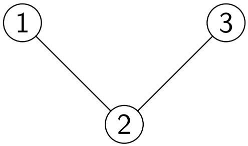


To distinguish vertices we need to color edges of Γ.

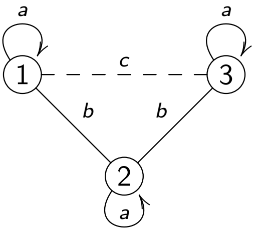
Algorithm. Set $\mathcal{C} = \{1_\Omega\} \cup \{E\} \cup \{(\Omega \times \Omega) \setminus E\}$.

- For all $(\alpha, \beta) \in \Omega \times \Omega$ and $R, S \in \mathcal{C}$ find the intersection number
$$c(\alpha, \beta; R, S) = |\alpha R \cap S\beta|.$$
- Build a new partition Ы$(\mathcal{C})$ by putting (α, β) and (α', β') to the same class of Ы$(\mathcal{C})$ if $|\alpha R \cap S\beta| = |\alpha' R \cap S\beta'|$ for all $R, S \in \mathcal{C}$.
- Repeat the procedure till $|\mathcal{C}|$ stops to increase.

The WL-algorithm. Very small example



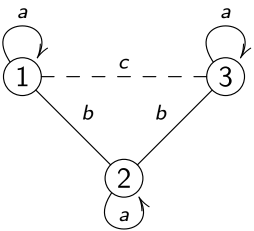
Initial coloring



Adjacency matrix

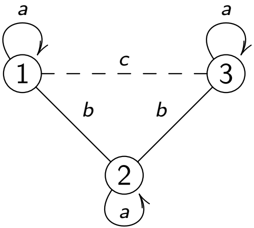
$$A = \begin{pmatrix} a & b & c \\ b & a & b \\ c & b & a \end{pmatrix}.$$

First iteration



$$A^2 = \begin{pmatrix} a^2+b^2+c^2 & ab+ba+cb & ac+b^2+ca \\ ba+ab+bc & 2b^2+a^2 & bc+ab+ba \\ ca+b^2+ac & cb+ba+ab & c^2+b^2+a^2 \end{pmatrix}.$$

First iteration



$$A^2 = \begin{pmatrix} a^2+b^2+c^2 & {\color{blue}ab+ba+cb} & {\color{green}ac+b^2+ca} \\ {\color{magenta}bc+ab+ba} & {\color{red}2b^2+a^2} & {\color{magenta}bc+ab+ba} \\ {\color{green}ac+b^2+ca} & {\color{blue}ab+ba+cb} & c^2+b^2+a^2 \end{pmatrix}.$$

Second iteration

New matrix A

$$A = \begin{pmatrix} d & e & f \\ g & h & g \\ f & e & d \end{pmatrix}$$

Second iteration

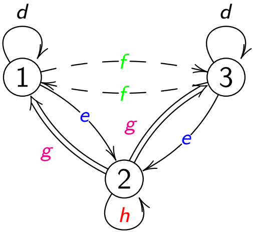
New matrix A

$$A = \begin{pmatrix} d & e & f \\ g & h & g \\ f & e & d \end{pmatrix}$$

Second iteration

New matrix A

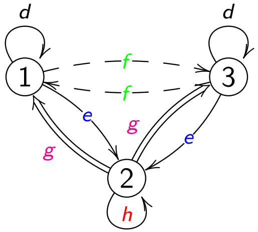
$$A = \begin{pmatrix} d & e & f \\ g & h & g \\ f & e & d \end{pmatrix}$$



Second iteration

New matrix A

$$A = \begin{pmatrix} d & e & f \\ g & h & g \\ f & e & d \end{pmatrix}$$



The matrix A is stable, that is A^2 produces the same coloring as A does.

WL-refinement (operation Ы)

Properties

WL-refinement (operation Ы)

Properties

- $\mathcal{C} \sqsubseteq \mathcal{S} \implies \text{Ы}(\mathcal{C}) \sqsubseteq \text{Ы}(\mathcal{S})$;

WL-refinement (operation Ы)

Properties

- $\mathcal{C} \sqsubseteq \mathcal{S} \implies \text{Ы}(\mathcal{C}) \sqsubseteq \text{Ы}(\mathcal{S})$;
- $\mathcal{C}^* = \mathcal{C} \implies \text{Ы}(\mathcal{C})^* = \text{Ы}(\mathcal{C})$;

WL-refinement (operation Ы)

Properties

- $\mathcal{C} \sqsubseteq \mathcal{S} \implies \text{Ы}(\mathcal{C}) \sqsubseteq \text{Ы}(\mathcal{S})$;
- $\mathcal{C}^* = \mathcal{C} \implies \text{Ы}(\mathcal{C})^* = \text{Ы}(\mathcal{C})$;
- $1_\Omega \in \mathcal{C}^\cup \implies \mathcal{C} \sqsubseteq \text{Ы}(\mathcal{C})$;

WL-refinement (operation Ы)

Properties

- $\mathcal{C} \sqsubseteq \mathcal{S} \implies$ Ы$(\mathcal{C}) \sqsubseteq$ Ы$(\mathcal{S})$;
- $\mathcal{C}^* = \mathcal{C} \implies$ Ы$(\mathcal{C})^* =$ Ы$(\mathcal{C})$;
- $1_\Omega \in \mathcal{C}^\cup \implies \mathcal{C} \sqsubseteq$ Ы$(\mathcal{C})$;

Proposition

Let $f : \Omega \to \Delta$ be a bijection that maps a partition $\mathcal{C}$ of Ω^2 onto a paritition $\mathcal{T}$ of Δ^2 (i.e. $\mathcal{C}^f = \mathcal{T}$). Then Ы$(\mathcal{C})^f =$ Ы$(\mathcal{T})$.

WL-refinement (operation Ы)

Properties

- $\mathcal{C} \sqsubseteq \mathcal{S} \implies \text{Ы}(\mathcal{C}) \sqsubseteq \text{Ы}(\mathcal{S})$;
- $\mathcal{C}^* = \mathcal{C} \implies \text{Ы}(\mathcal{C})^* = \text{Ы}(\mathcal{C})$;
- $1_\Omega \in \mathcal{C}^\cup \implies \mathcal{C} \sqsubseteq \text{Ы}(\mathcal{C})$;

Proposition

Let $f : \Omega \to \Delta$ be a bijection that maps a partition $\mathcal{C}$ of Ω^2 onto a paritition $\mathcal{T}$ of Δ^2 (i.e. $\mathcal{C}^f = \mathcal{T}$). Then $\text{Ы}(\mathcal{C})^f = \text{Ы}(\mathcal{T})$.

Given an ordered partition $\vec{\mathcal{C}} = (S_1, \ldots, S_m)$ of Ω^2 the WL-algorithm produces a unique (canonical) ordering of the refinement $\text{Ы}(\mathcal{C})$ (denoted as $\text{Ы}(\vec{\mathcal{C}})$) with the following property:

$$\vec{\mathcal{C}}^f = \vec{\mathcal{T}} \implies \text{Ы}(\vec{\mathcal{C}})^f = \text{Ы}(\vec{\mathcal{T}})$$

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,
- $\mathcal{C} = \text{Ы}(\mathcal{C})$,

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,
- $\mathcal{C} =$ ЫI$(\mathcal{C})$,that is for all $R, S, T \in \mathcal{C}$ the intersection number $c_{RS}^T = |\alpha R \cap S\beta|$ does not depend on the choice of $(\alpha, \beta) \in T$.

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,
- $\mathcal{C} = \text{Ы}(\mathcal{C})$,that is for all $R, S, T \in \mathcal{C}$ the intersection number $c_{RS}^T = |\alpha R \cap S\beta|$ does not depend on the choice of $(\alpha, \beta) \in T$.
- the degree and rank of $\mathcal{X}$ are the numbers $|\Omega|$ and $|\mathcal{C}|$,

Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,
- $\mathcal{C} = \text{Ы}(\mathcal{C})$,that is for all $R, S, T \in \mathcal{C}$ the intersection number $c_{RS}^T = |\alpha R \cap S\beta|$ does not depend on the choice of $(\alpha, \beta) \in T$.
- the degree and rank of $\mathcal{X}$ are the numbers $|\Omega|$ and $|\mathcal{C}|$,
- the basic relations and relations of $\mathcal{X}$ are the relations of $\mathcal{C}$ and of $\mathcal{C}^\cup$.

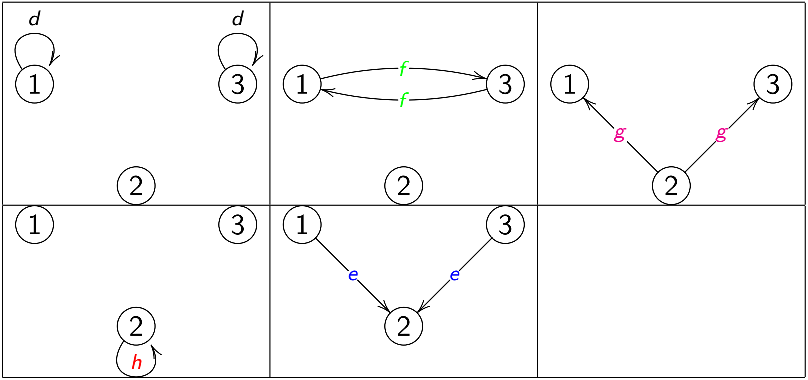
Coherent configurations (D. Higman, 1970).

The output partition of the Weisfeiler-Leman algorithm is a coherent configuration, i.e. a pair $\mathcal{X} = (\Omega, \mathcal{C})$ such that:

- $\mathcal{C}$ is a partition of $\Omega \times \Omega$,
- $1_\Omega \in \mathcal{C}^\cup$,
- $\mathcal{C}^* = \mathcal{C}$,
- $\mathcal{C} = $ Ы$(\mathcal{C})$,that is for all $R, S, T \in \mathcal{C}$ the intersection number $c_{RS}^T = |\alpha R \cap S\beta|$ does not depend on the choice of $(\alpha, \beta) \in T$.
- the degree and rank of $\mathcal{X}$ are the numbers $|\Omega|$ and $|\mathcal{C}|$,
- the basic relations and relations of $\mathcal{X}$ are the relations of $\mathcal{C}$ and of $\mathcal{C}^\cup$.

The configuration $\mathcal{X}$ is homogeneous (or association scheme, or scheme), if $1_\Omega \in \mathcal{C}$.

Coherent configurations: a concrete example.



Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$. Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Proposition. The following statements hold:

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Proposition. The following statements hold:

- $\Omega = \bigcup_{\Delta \in \Phi} \Delta$,

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Proposition. The following statements hold:

- $\Omega = \bigcup_{\Delta\in\Phi} \Delta$,
- for any $S \in \mathcal{C}$ the sets $D(S)$ and $R(S)$ are fibres of $\mathcal{X}$,

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Proposition. The following statements hold:

- $\Omega = \bigcup_{\Delta \in \Phi} \Delta$,
- for any $S \in \mathcal{C}$ the sets $D(S)$ and $R(S)$ are fibres of $\mathcal{X}$,
- for any $S \in \mathcal{C}$ and $\alpha \in D(S)$ we have $|\alpha S| = c_{SS^*}^T$ where $T = 1_{D(S)}$.

Coherent configurations. Fibers and relations.

A fiber of $\mathcal{X}$ is a set $\Delta \subset \Omega$ such that $1_\Delta \in \mathcal{C}$; the set of all fibers is denoted by $\Phi = \Phi(\mathcal{X})$.Thus $\mathcal{X}$ is a scheme iff $|\Phi| = 1$.

Proposition. The following statements hold:

- $\Omega = \bigcup_{\Delta \in \Phi} \Delta$,
- for any $S \in \mathcal{C}$ the sets $D(S)$ and $R(S)$ are fibres of $\mathcal{X}$,
- for any $S \in \mathcal{C}$ and $\alpha \in D(S)$ we have $|\alpha S| = c_{SS^*}^T$ where $T = 1_{D(S)}$.
- for any fiber $\Delta \in \Phi$ the set of relations $\mathcal{C}_\Delta := \{C \in \mathcal{C} \mid D(C) = \Delta, R(C) = \Delta\}$ form a homogeneous co.co. on Δ, called a homogeneous constituent of $\mathcal{C}$.

The number $n_S = c_{SS^*}^T$ is called the valency of S.

Properties of coherent configurations.

Proposition

Let $\mathcal{X} = (\Omega, \mathcal{C})$ be a co.co. Then

- the set $\mathcal{C}^{\cup}$ is closed w.r.t. boolean operations;
- $1_{\Omega}, \Omega^2 \in \mathcal{C}^{\cup}$;
- $(\mathcal{C}^{\cup})^* = \mathcal{C}^{\cup}$;
- $\mathcal{C}^{\cup}$ is closed w.r.t. relational product;

Isomorphisms between coherent configurations

Definition

Two coherent configuration $\mathcal{X} = (\Omega, \mathcal{C})$ and $\mathcal{X}' = (\Omega', \mathcal{C}')$ are called (combinatorially) isomorphic iff there exist bijections $f : \Omega \to \Omega', \phi : \mathcal{C} \to \mathcal{C}'$ such that

$$\forall_{\alpha,\beta\in\Omega} \;\; (\alpha, \beta) \in C \iff (\alpha^f, \beta^f) \in C^\phi.$$

The set of all isomorphisms between $\mathcal{X}$ and $\mathcal{X}'$ is denoted as $\mathrm{Iso}(\mathcal{X}, \mathcal{X}')$. Notice that ϕ is uniquely determined by f.

In what follows we set $\mathrm{Iso}(\mathcal{X}) := \mathrm{Iso}(\mathcal{X}, \mathcal{X})$. We call the elements of this group colored automorphisms of the configuration.

Coherent configurations generated by a graph.

The mapping $(f, \phi) \mapsto \phi$ is an group homomorphism from $\mathrm{Iso}(\mathcal{X})$ into $\mathrm{Sym}(\mathcal{C})$. The kernel of this homomorphism denoted as $\mathrm{Aut}(\mathcal{X})$ is called the the automorphism group of $\mathcal{X}$:

$$\mathrm{Aut}(\mathcal{X}) = \{f \in \mathrm{Sym}(\Omega) : \ S^f = S \text{ for all } S \in \mathcal{C}\}$$

Theorem

Let $\langle\langle \Gamma \rangle\rangle$ be the WL-closure of a graph $\Gamma = (\Omega, E)$ obtained by applying WL-algorithm to Γ. Then

- $E \in \langle\langle \Gamma \rangle\rangle^{\cup}$;
- $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\langle\langle \Gamma \rangle\rangle)$.

Coherent configurations generated by a graph.

The mapping $(f, \phi) \mapsto \phi$ is an group homomorphism from $\mathrm{Iso}(\mathcal{X})$ into $\mathrm{Sym}(\mathcal{C})$. The kernel of this homomorphism denoted as $\mathrm{Aut}(\mathcal{X})$ is called the the automorphism group of $\mathcal{X}$:

$$\mathrm{Aut}(\mathcal{X}) = \{f \in \mathrm{Sym}(\Omega) : \; S^f = S \text{ for all } S \in \mathcal{C}\}$$

Theorem

Let $\langle\langle \Gamma \rangle\rangle$ be the WL-closure of a graph $\Gamma = (\Omega, E)$ obtained by applying WL-algorithm to Γ. Then

- $E \in \langle\langle \Gamma \rangle\rangle^{\cup}$;
- $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\langle\langle \Gamma \rangle\rangle)$.

Examples. Strongly regular graphs.

Definition

A graph $\Gamma = (\Omega, E)$ is called strongly regular if its WL-closure has rank three. In other words, WL-algorithm stops at the first iteration and $\langle\langle \Gamma \rangle\rangle = \{1_\Omega, E, E^c\}$.

Examples. Strongly regular graphs.

Definition

A graph $\Gamma = (\Omega, E)$ is called strongly regular if its WL-closure has rank three. In other words, WL-algorithm stops at the first iteration and $\langle\langle \Gamma \rangle\rangle = \{1_\Omega, E, E^c\}$.

Proposition

A graph $\Gamma = (\Omega, E)$ is strongly regular if and only if there exists non-negative integers k, λ, μ such that

1. Γ is k-regular,
2. any pair of points connected by an edge have λ common neighbours,
3. any pair of points not connected by an edge have μ common neighbours

Examples. Permutation groups.

Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega, \ g \in G.$$

Examples. Permutation groups.

Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega,\ g \in G.$$

Set $\mathrm{Inv}(G) := (\Omega, \mathcal{C})$ where $\mathcal{C} := \mathrm{Orb}(G, \Omega \times \Omega)$. Then

Examples. Permutation groups.

Let $G \le \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega,\ g \in G.$$

Set $\mathrm{Inv}(G) := (\Omega, \mathcal{C})$ where $\mathcal{C} := \mathrm{Orb}(G, \Omega \times \Omega)$. Then

1. $\mathrm{Inv}(G)$ is a coherent configuration (of G),

Examples. Permutation groups.

Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega,\ g \in G.$$

Set $\mathrm{Inv}(G) := (\Omega, \mathcal{C})$ where $\mathcal{C} := \mathrm{Orb}(G, \Omega \times \Omega)$. Then

1. $\mathrm{Inv}(G)$ is a coherent configuration (of G),
2. the basic relations of $\mathcal{X}$ are the 2-orbits of G,

Examples. Permutation groups.

Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega,\ g \in G.$$

Set $\mathrm{Inv}(G) := (\Omega, \mathcal{C})$ where $\mathcal{C} := \mathrm{Orb}(G, \Omega \times \Omega)$. Then

1. $\mathrm{Inv}(G)$ is a coherent configuration (of G),
2. the basic relations of $\mathcal{X}$ are the 2-orbits of G,
3. $\Phi(\mathcal{X}) = \mathrm{Orb}(G, \Omega)$, in particular $\mathcal{X}$ is a scheme iff G is transitive;

Examples. Permutation groups.

Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. It acts on $\Omega \times \Omega$:

$$(\alpha, \beta)^g := (\alpha^g, \beta^g), \qquad \alpha, \beta \in \Omega,\ g \in G.$$

Set $\mathrm{Inv}(G) := (\Omega, \mathcal{C})$ where $\mathcal{C} := \mathrm{Orb}(G, \Omega \times \Omega)$. Then

1. $\mathrm{Inv}(G)$ is a coherent configuration (of G),
2. the basic relations of $\mathcal{X}$ are the 2-orbits of G,
3. $\Phi(\mathcal{X}) = \mathrm{Orb}(G, \Omega)$, in particular $\mathcal{X}$ is a scheme iff G is transitive;

Definition.

A coherent configuration $\mathcal{X}$ is called schurian if $\mathcal{X} = \mathrm{Inv}(G)$ for some group G.

Schurity problem

Given a coherent configuration $\mathcal{X}$, find whether it is schurian.

Galois correspondence.

Definition

Let $\mathcal{X} = (\Omega, \mathcal{C}), \mathcal{X}' = (\Omega, \mathcal{C}')$ be two coherent configuratios. We say that $\mathcal{X}$ is a fusion of $\mathcal{X}'$ (equivalently $\mathcal{X}'$ is a fission of $\mathcal{X}$), notation $\mathcal{X} \sqsubseteq \mathcal{X}'$ if $\mathcal{C} \sqsubseteq \mathcal{C}'$.

Galois correspondence.

Definition

Let $\mathcal{X} = (\Omega, \mathcal{C}), \mathcal{X}' = (\Omega, \mathcal{C}')$ be two coherent configuratios. We say that $\mathcal{X}$ is a fusion of $\mathcal{X}'$ (equivalently $\mathcal{X}'$ is a fission of $\mathcal{X}$), notation $\mathcal{X} \sqsubseteq \mathcal{X}'$ if $\mathcal{C} \sqsubseteq \mathcal{C}'$.

Proposition

Let $\mathcal{X}, \mathcal{X}'$ be two coherent configurations defined on Ω and $G, H \leq \mathrm{Sym}(\Omega)$ arbitrary subgroups. Then

- $\mathcal{X} \sqsubseteq \mathcal{X}' \implies \mathrm{Aut}(\mathcal{X}) \geq \mathrm{Aut}(\mathcal{X}')$;
- $H \leq G \implies \mathrm{Inv}(H) \sqsupseteq \mathrm{Inv}(G)$;
- $G \leq \mathrm{Aut}(\mathrm{Inv}(G)$;
- $\mathcal{X} \sqsubseteq \mathrm{Inv}(\mathrm{Aut}(\mathcal{X}))$

Galois closed objects.

Definition

The group $G^{(2)} := \mathsf{Aut}(\mathsf{Inv}(G))$ is called a 2-closure of $G \leq \mathsf{Sym}(\Omega)$. A group is called 2-closed if $G = G^{(2)}$.

Definition

Given a coherent configuration $\mathcal{X} = (\Omega, \mathcal{C})$, the configuration $\mathrm{Sch}(\mathcal{X}) := \mathsf{Inv}(\mathsf{Aut}(\mathcal{X}))$ is called a Schurian closure of $\mathcal{X}$. A configuration $\mathcal{X}$ is schurian iff $\mathrm{Sch}(\mathcal{X}) = \mathcal{X}$.

Theorem

The mappings $(\mathsf{Aut}, \mathsf{Inv})$ are bijections between 2-closed subgroups of $\mathsf{Sym}(\Omega)$ and schurian coherent configurations defined on Ω.

Theorem.

The GI is polynomially equivalent to the problem of finding the schurian closure of a coherent configuration.

Galois closed objects.

Definition

The group $G^{(2)} := \mathsf{Aut}(\mathsf{Inv}(G))$ is called a 2-closure of $G \leq \mathsf{Sym}(\Omega)$. A group is called 2-closed if $G = G^{(2)}$.

Definition

Given a coherent configuration $\mathcal{X} = (\Omega, \mathcal{C})$, the configuration $\mathrm{Sch}(\mathcal{X}) := \mathsf{Inv}(\mathsf{Aut}(\mathcal{X}))$ is called a Schurian closure of $\mathcal{X}$. A configuration $\mathcal{X}$ is schurian iff $\mathrm{Sch}(\mathcal{X}) = \mathcal{X}$.

Theorem

The mappings $(\mathsf{Aut}, \mathsf{Inv})$ are bijections between 2-closed subgroups of $\mathsf{Sym}(\Omega)$ and schurian coherent configurations defined on Ω.

Theorem.

The GI is polynomially equivalent to the problem of finding the schurian closure of a coherent configuration.