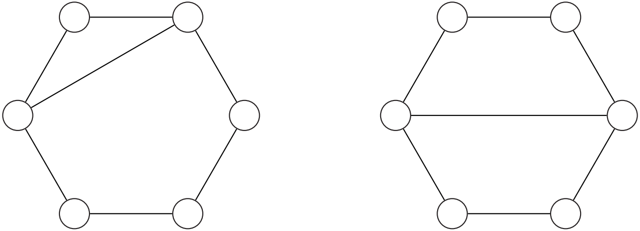
The Weisfeiler-Leman Dimension and the Speed of Color Stabilization — Logical and Algorithmic Aspects

Oleg Verbitsky

Humboldt University of Berlin

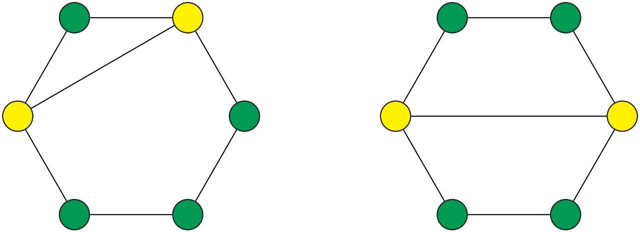
Symmetry vs Regularity, Pilsen, 3 July, 2018

Color refinement algorithm, a.k.a. WL[1]



The initial coloring C^0 is monochromatic.

Color refinement algorithm, a.k.a. WL[1]



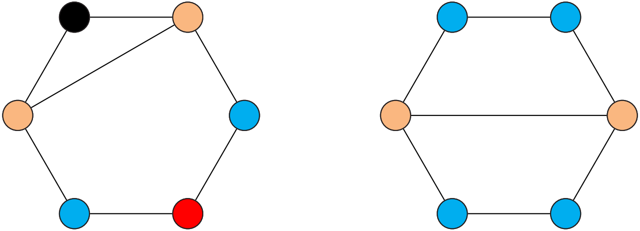
A color refinement step: $C^{i+1}(v) = C^i(v) \mid \{\!\{ \, C^i(u) \, \}\!\}_{u \in N(v)}$

$\bigcirc = \bigcirc \mid \{\bigcirc, \bigcirc\}$

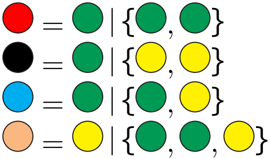
$\bigcirc = \bigcirc \mid \{\bigcirc, \bigcirc, \bigcirc\}$

Note that $C^1(v) \equiv \deg v$

Color refinement algorithm, a.k.a. WL[1]



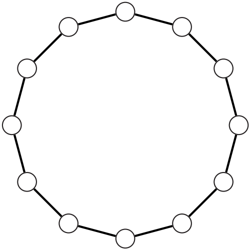
The next color refinement: $C^2(v) = C^1(v) \mid \{\!\{\, C^1(u) \,\}\!\}_{u \in N(v)}$

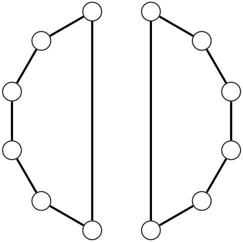


The color partition is refined in each step.

Once it stabilizes, we can terminate.

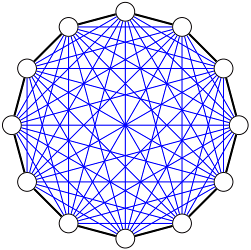
The Weisfeiler-Leman algorithm, a.k.a. WL[2]

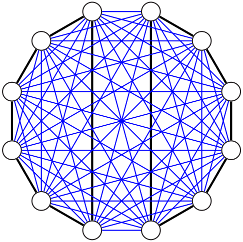




Input graphs: C_{12} and $2\,C_6$

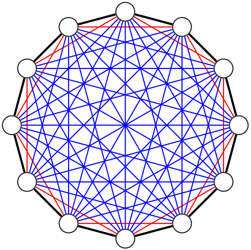
The Weisfeiler-Leman algorithm, a.k.a. WL[2]

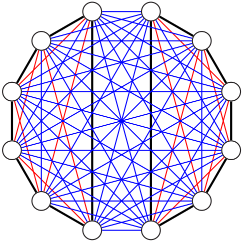




The initial coloring C^0 of V^2: "edges" for (u,v) s.t. $u \sim v$, "non-edges" for (u,v) s.t. $u \not\sim v$, "vertices" for (u,v) s.t. $u = v$.

The Weisfeiler-Leman algorithm, a.k.a. WL[2]

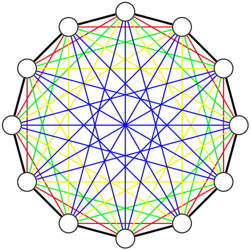


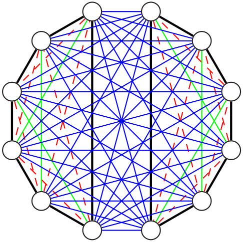


A color refinement step:

$$C^{i+1}(u,v) = C^{i}(u,v) \mid \{\!\{\, C^{i}(u,w) \mid C^{i}(w,v) \,\}\!\}_{w \in V}$$

The Weisfeiler-Leman algorithm, a.k.a. WL[2]

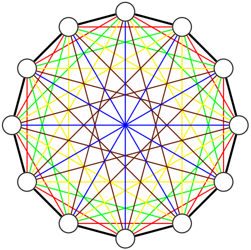


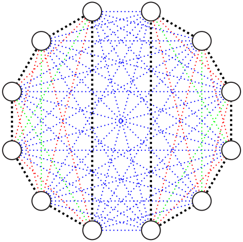


The next color refinement step:
$C^2(u,v) = C^1(u,v) \mid \{\!\{\, C^1(u,w) \mid C^1(w,v) \,\}\!\}_{w \in V}$

The non-isomorphism recognized.

The Weisfeiler-Leman algorithm, a.k.a. WL[2]

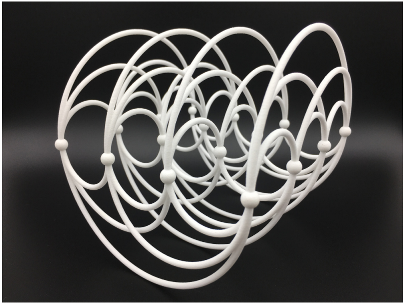


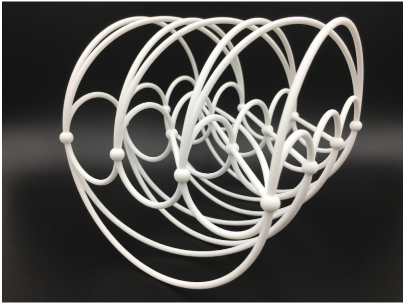


The next color refinement step:
$C^3(u,v) = C^2(u,v) \mid \{\!\{ C^2(u,w) \mid C^2(w,v) \}\!\}_{w \in V}$

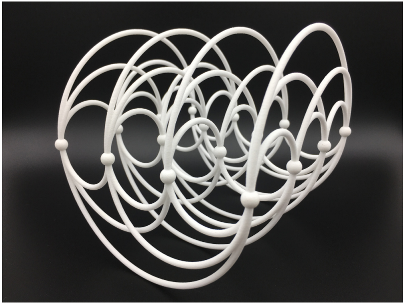
The color partition stabilized.

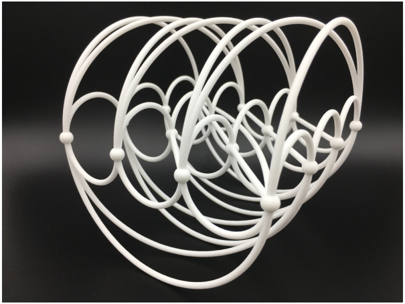
The 4×4 rook's graph and the Shrikhande graph
(3D printing by Jonathan Gerhard)





The 4×4 rook's graph and the Shrikhande graph
(3D printing by Jonathan Gerhard)





WL[3] distinguishes between these two SRGs(16,6,2,2).

The initial coloring C^0 of V^3: according to the equality type of (x_1, x_2, x_3) and the isomorphism type of $G[x_1, x_2, x_3]$.

Color refinement step: $C^{i+1}(x_1, x_2, x_3) = C^i(x_1, x_2, x_3) \mid \{\!\!\{ C^1(w, x_2, x_3) \mid C^1(x_1, w, x_3) \mid C^1(x_1, x_2, w) \}\!\!\}_{w \in V}$

The Weisfeiler-Leman dimension of a graph

Definition

$\dim_{WL}(G)$ is equal to the minimum k such that WL[k] distinguishes G from every non-isomorphic graph H.

A basic observation

If $\dim_{WL}(G) = O(1)$ for all G in a class of graphs $\mathcal{C}$, then the isomorphism problem for $\mathcal{C}$ is solvable in polynomial time.

Examples:

- $\dim_{WL}(G) = 1$ if G is a tree [Edmonds 65]
- $\dim_{WL}(G) \leq 3$ if G planar [Kiefer, Ponomarenko, Schweitzer 2017]
- $\dim_{WL}(G) \leq 2$ if G is an interval graph [Evdokimov, Ponomarenko, Tinhofer 2000]
- $\dim_{WL}(G) \leq k+1$ if $\mathrm{tw}(G) \leq k$ [Grohe, Mariño 1999]

The Weisfeiler-Leman dimension of a graph

Theorem (Arvind, Köbler, Rattan, V. 2015; Kiefer, Schweitzer, Selman 2015)

Given a graph G, one can efficiently recognize whether or not $\dim_{WL}(G) = 1$.

Remark: If the same would be possible for dimension 2, then for a given strongly regular graph we could efficiently decide whether or not it is uniquely determined by its parameters.

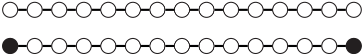
The speed of color stabilization

WL[1]

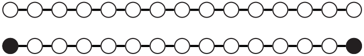
WL[2] (only the diagonal of V^2 is shown)

The speed of color stabilization

WL[1]

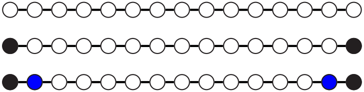


WL[2] (only the diagonal of V^2 is shown)

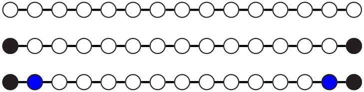


The speed of color stabilization

WL[1]

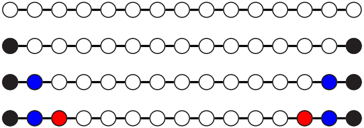


WL[2] (only the diagonal of V^2 is shown)

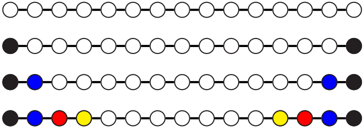


The speed of color stabilization

WL[1]

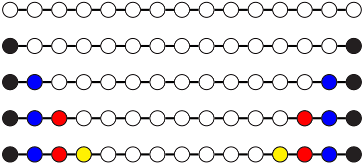


WL[2] (only the diagonal of V^2 is shown)

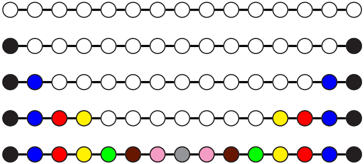


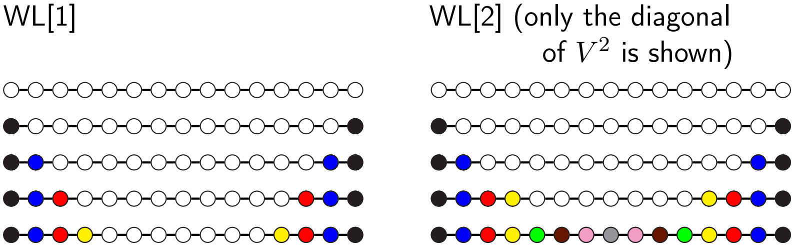
The speed of color stabilization

WL[1]



WL[2] (only the diagonal of V^2 is shown)





Definition

Let $\mathrm{Stab}_k(G)$ denote the number of refinement rounds that WL[k] performs on input G until stabilization.

Example:

- $\mathrm{Stab}_1(P_n) = \lfloor (n-1)/2 \rfloor$ while $\mathrm{Stab}_2(P_n) \leq \lceil \log_2 n \rceil$.

Importance of fast stabilization I

Theorem (Babai, Erdős, Selkow 1980)

$\mathrm{Stab}_1(G) \leq 2$ *for almost all* G *and, moreover, the stable partition consists of singletons.*

Corollary

The average case of Graph Isomorphism is solvable in linear time.

Theorem (Bollobás 1982)

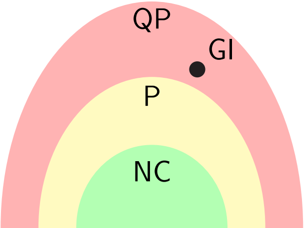
$\mathrm{Stab}_2(G) = O(\log \log n)$ *for almost all regular* G *of a fixed degree.*

Here and below n denotes the number of vertices in G.

Importance of fast stabilization II

An observation

Suppose that
$$\dim_{WL}(G) \le k$$
for all G in a class of graphs $\mathcal{C}$ and, moreover,
$$\mathrm{Stab}_k(G) = O(\log n).$$
Then the isomorphism problem for $\mathcal{C}$ is solvable in polylogarithmic time using polynomially many parallel processors.



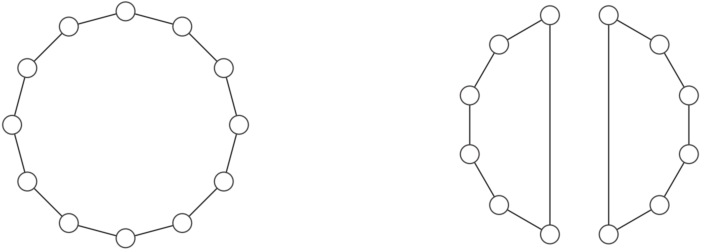
QP – quasi-polynomial time

P – polynomial time

NC – parallel polylogarithmic time

Logic behind WL[k]

Suppose we are given two non-isomorphic graphs.



We want to *succinctly* express the reasons *why* these graphs are non-isomorphic.

Logic behind WL[k]

Vocabulary:

$=$ equality of vertices

$\sim$ adjacency of vertices

Syntax:

$\wedge, \vee, \neg$ etc. Boolean connectives

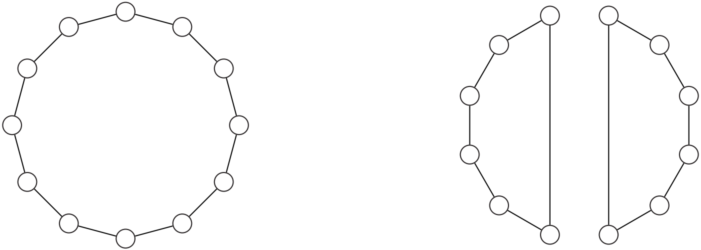
$\exists, \forall$ quantification over vertices

(no quantification over sets).

Moreover:

$\exists^{\geq m}, \exists^{=m}$ etc. counting quantifiers

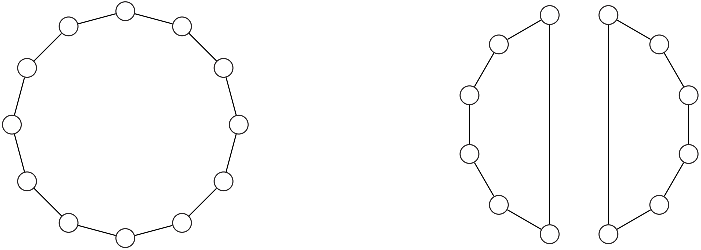
Logic behind WL[k]



Example: True for both graphs

$$\exists^{=12} x\,(x = x) \land \forall x\, \exists^{=2} y\,(y \sim x).$$

Logic behind WL[k]



True for C_{12} but false for $2C_6$:

$$\forall x \, \forall y \; \mathrm{dist}_{\leq 6}(x, y),$$

where $\mathrm{dist}_{\leq n}(x, y)$ is an expression for

"x and y are at distance at most n from each other"

Succinctness of logical expressions

1st version: $n+1$ variables, quantifier depth $n-1$

$$\mathrm{dist}_{\leq 1}(x,y) \overset{\mathsf{def}}{=} x \sim y \vee x = y$$

$$\mathrm{dist}_{\leq n}(x,y) \overset{\mathsf{def}}{=} \exists z_1 \ldots \exists z_{n-1} \Big(\mathrm{dist}_{\leq 1}(x,z_1) \wedge \mathrm{dist}_{\leq 1}(z_1,z_2) \wedge \ldots$$
$$\ldots \wedge \mathrm{dist}_{\leq 1}(z_{n-2},z_{n-1}) \wedge \mathrm{dist}_{\leq 1}(z_{n-1},y) \Big)$$

- $\forall x(\forall y(\exists z(\ldots)))$ – depth 3; $(\forall x \ldots) \wedge (\forall y \ldots) \wedge (\exists z \ldots)$ – depth 1

Succinctness of logical expressions

1st version: $n+1$ variables, quantifier depth $n-1$

$$\mathrm{dist}_{\leq 1}(x,y) \overset{\text{def}}{=} x \sim y \lor x = y$$

$$\mathrm{dist}_{\leq n}(x,y) \overset{\text{def}}{=} \exists z_1 \ldots \exists z_{n-1} \Big(\mathrm{dist}_{\leq 1}(x,z_1) \land \mathrm{dist}_{\leq 1}(z_1,z_2) \land \ldots$$
$$\ldots \land \mathrm{dist}_{\leq 1}(z_{n-2},z_{n-1}) \land \mathrm{dist}_{\leq 1}(z_{n-1},y) \Big)$$

- $\forall x(\forall y(\exists z(\ldots)))$ – depth 3; $(\forall x \ldots) \land (\forall y \ldots) \land (\exists z \ldots)$ – depth 1

2nd version: 3 variables, quantifier depth $\lceil \log_2 n \rceil$

$$\mathrm{dist}_{\leq n}(x,y) \overset{\text{def}}{=} \exists z \left(\mathrm{dist}_{\leq \lfloor n/2 \rfloor}(x,z) \land \mathrm{dist}_{\leq \lceil n/2 \rceil}(z,y) \right).$$

A logical characterization of WL[k]

Theorem (Cai, Fürer, Immerman 1992)

1. *WL[k] distinguishes G and H iff G and H are distinguishable by a $(k+1)$-variable sentence with counting quantifiers.*
2. *Moreover,*

   G and H are distinguishable by a sentence with $k+1$ variables of quantifier depth $r+1$

   $$\Downarrow$$

   WL[k] distinguishes G and H after r rounds

   $$\Downarrow$$

   G and H are distinguishable by a sentence with $k+1$ variables of quantifier depth $r+k$

Three reasons why the logical characterization is useful

1. A sentence *defines* a graph G if it is true on G and false on any non-isomorphic graph. If we prove that each graph G in a class $\mathcal{C}$ is definable by a sentence with $k+1$ variables, we know that the isomorphism problem for G is solvable by WL[k]. A prominent example:
    - any class of graphs excluding a fixed minor [Grohe 2012]

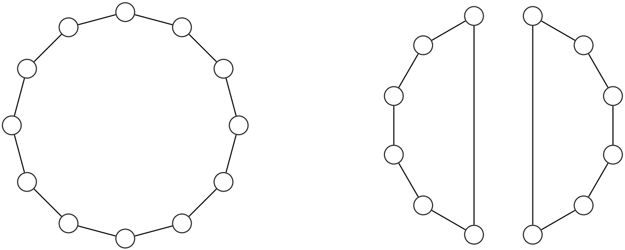
Three reasons why the logical characterization is useful

1. A sentence *defines* a graph G if it is true on G and false on any non-isomorphic graph. If we prove that each graph G in a class $\mathcal{C}$ is definable by a sentence with $k+1$ variables, we know that the isomorphism problem for G is solvable by WL[k]. A prominent example:
    - any class of graphs excluding a fixed minor [Grohe 2012]
2. If we show that, moreover, each $G \in \mathcal{C}$ is definable with quantifier depth $O(\log n)$, then the isomorphism problem for G is solved even in NC.

Three reasons why the logical characterization is useful

1. A sentence *defines* a graph G if it is true on G and false on any non-isomorphic graph. If we prove that each graph G in a class $\mathcal{C}$ is definable by a sentence with $k+1$ variables, we know that the isomorphism problem for G is solvable by WL[k]. A prominent example:
   - any class of graphs excluding a fixed minor [Grohe 2012]
2. If we show that, moreover, each $G \in \mathcal{C}$ is definable with quantifier depth $O(\log n)$, then the isomorphism problem for G is solved even in NC.
3. Logic gives us a useful working tool — the Ehrenfeucht game.

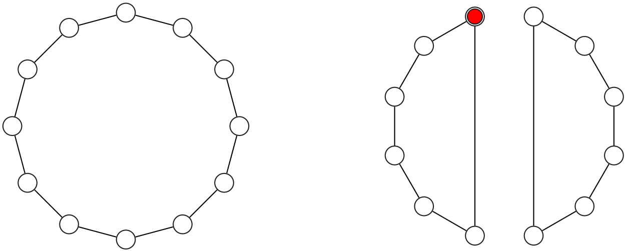
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

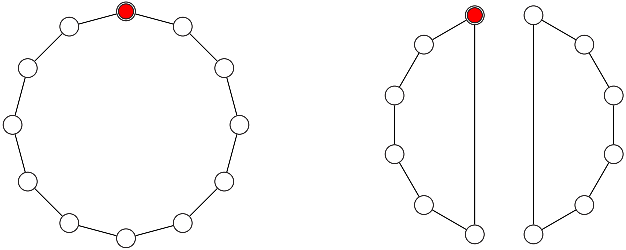
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

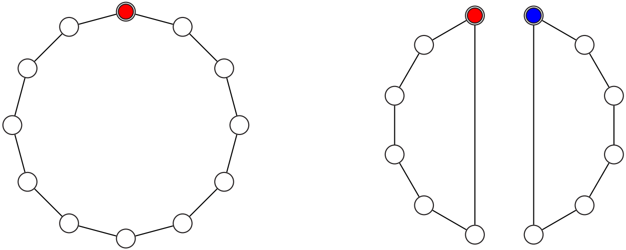
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

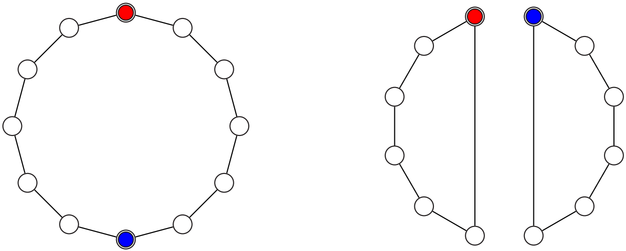
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

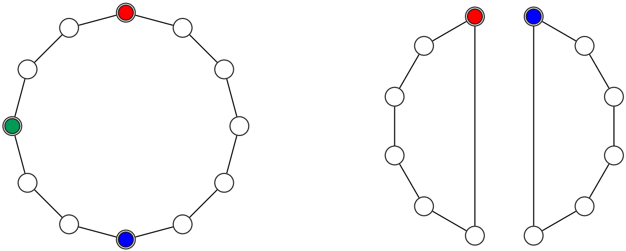
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

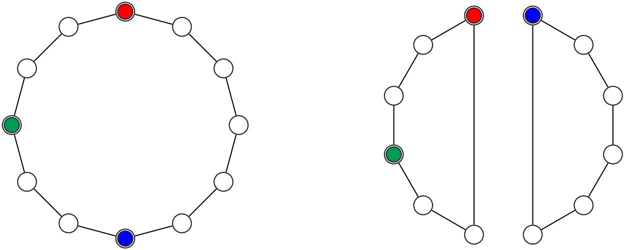
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

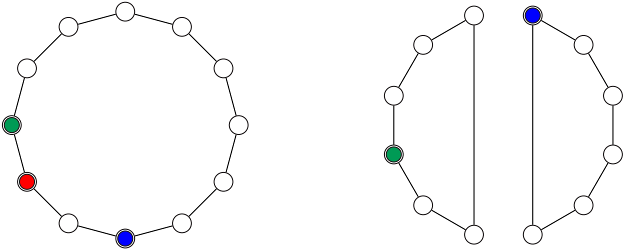
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

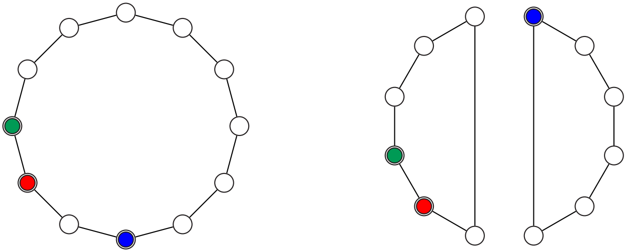
Ehrenfeucht game



Two players: Spoiler and Duplicator

Duplicator's objective: to keep a partial isomorphism

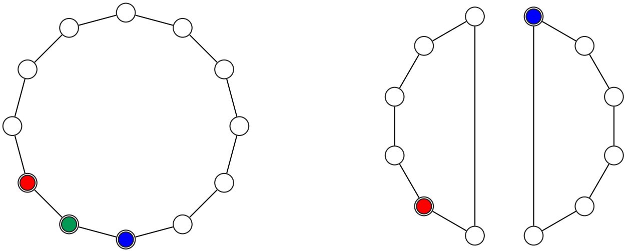
Ehrenfeucht game



Two players: Spoiler and Duplicator

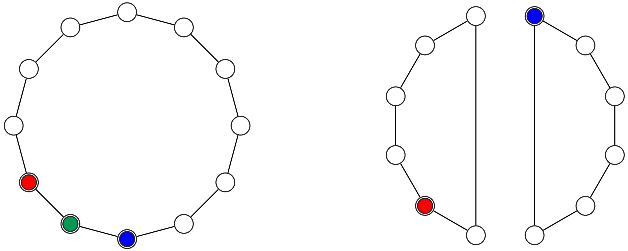
Duplicator's objective: to keep a partial isomorphism

Ehrenfeucht game



Two players: Spoiler and Duplicator

Ehrenfeucht game



Two players: Spoiler and Duplicator

Conclusion: If G is connected and H is disconnected, then Spoiler wins with 3 pebbles in $O(\log n)$ moves $\implies$ G and H are distinguishable with 3 variables and quantifier depth $O(\log n)$ $\implies$ WL[2] distinguishes G and H making $O(\log n)$ refinement rounds

A lower bound

Comments:

1. We have just seen a simplified version of the game, which corresponds to logic without counting.
2. The game is useful in proving both positive and negative results.

Theorem (Cai, Fürer, Immerman 1992)

There are infinitely many regular graphs G of degree 3 with $\dim_{WL}(G) > 0.004\, n$.

A positive example

Recall that every tree T has $\dim_{WL}(T) = 1$ or, equivalently, is definable using 2 variables. For some trees we need linear quantifier depth. This can be improved:

- one extra variable $\implies$ logarithmic depth !

Theorem

Every n-vertex tree T is definable with 3 variables and quantifier depth $\leq 3 \log n$.

A positive example

Recall that every tree T has $\dim_{WL}(T) = 1$ or, equivalently, is definable using 2 variables. For some trees we need linear quantifier depth. This can be improved:

- one extra variable $\implies$ logarithmic depth !

Theorem

Every n-vertex tree T is definable with 3 variables and quantifier depth $\leq 3 \log n$.

Proof idea:

- Consider non-isomorphic trees T and T'.
- We can suppose that the diameters of T and T' are equal for else Spoiler wins with 3 pebbles in logarithmically many moves.
- For simplicity, assume that the diameter is even, hence both T and T' have a unique central vertex.
- If Spoiler pebbles the central vertex in one of the trees, Duplicator is forced to respond with the central vertex in the other tree (for else he loses in a log number of rounds).

cont'd

- In this way, Spoiler can play as if T and T' were directed trees rooted at their central vertices.
- Since $T \not\cong T'$, T and T' differ by the number of branches of some isomorphism type from the root.
- Spoiler can force further play on non-isomorphic branches. He repeats this, making the branches each time smaller.
- To speed up, Spoiler marks a branch by pebbling its *separator*. A vertex v is a separator of T if every connected component of $T - v$ has size at most $n/2$. The most non-obvious part of the argument is that 3 pebbles are still enough.

History of the tree isomorphism problem

Theorem

Every n-vertex tree T is definable with 3 variables and quantifier depth $\leq 3\,\log n$.

Testing isomorphism of trees is

- in Lin-Time by CR [Edmonds 1965]
- in NC if $\Delta = O(\log n)$ [Ruzzo 1981]
- in NC [Miller-Reif 1991]
- in Log-Space [Lindell 1992]

Miller and Reif [SIAM J. Comput. 1991]: *"No polylogarithmic parallel algorithm was previously known for isomorphism of unbounded-degree trees."*

However, the $3\log n$-round WL[2] solves it in NC and is known since 1968 !

50 years of WL[2]

УДК 519.1

ПРИВЕДЕНИЕ ГРАФА К КАНОНИЧЕСКОМУ ВИДУ И ВОЗНИКАЮЩАЯ ПРИ ЭТОМ АЛГЕБРА

Б. Ю. ВЕЙСФЕЙЛЕР, А. А. ЛЕМАН

Рассматривается алгоритм приведения заданного конечного мультиграфа Г к канsocialическому виду. В процессе такого приведения возникает новый инвариант графа— алгебра $\mathfrak{A}$ (Г). Изучение свойств алгебры $\mathfrak{A}$ (Г) оказывается полезным при решении некоторых задач теории графов.

Выдвигаются и обсуждаются некоторые предположения относительно связи между свойствами алгебры $\mathfrak{A}$ (Г) и группой автоморфизмов графа Aut (Г). Построен пример неориентированного графа Г, алгебра $\mathfrak{A}$ (Г) которого совпадает с групповой алгеброй некоторой некоммутативной группы.

An algorithm is considered, reducing the specified finite multigraph Г to canonical form. In the course of this reduction, a new invariant of the graph is generated — algebra $\mathfrak{A}$ (Г). Study of the properties of the algebra $\mathfrak{A}$ (Г) proves helpful in solving a nu mber of graph-theoretic problems. Some propositions concerning the relationships between the properties of the algebra $\mathfrak{A}$ (Г) and the graph's automorphism group *Aut* (Г) are discussed. An example of non—oriented graph Г is constructed whose algebra $\mathfrak{A}$ (Г) coincides with the group algebra of a non—commutative group.

1. Рассмотрим произвольный конечный граф Г и его матрицу смежности $A(\Gamma)=\{a_{ij}\}$; здесь a_{ij}—число ребер, ведущих из i-й вершины графа в j-ую; $i, j=1, 2, \ldots, n$. В случае неориентированного графа полагаем $a_{ij}=a_{ji}$. Каноническим видом графа мы будем называть его матрицу смежности при канонической нумерации вершин, т. е. при таком частичном упорядочении множества вершин, при котором из того, что a и b несравнимы следует, что существует автоморфизм графа, переводящий вершину a в b и сохраняющий отношение смежности.

В п. 6, 7 описан процесс приведения графа к каноническому виду, состоящий в поэтапном переупорядочении строк и столбцов матрицы $A(\Gamma)$, который, грубо говоря, сводится к следующему.

Рассмотрим для простоты неориентированный граф без кратных связей. Сначала каждой вершине графа сопоставим характеристический вектор, единственная компонента которого равна числу соседей данной вершины. Затем разобьем вершины на классы, так чтобы вершины с одинаковым характеристическим вектором принадлежали одному классу; классы при этом упорядочим в соответствии с естественным порядком в множестве характеристических векторов. Далее, каждой вершине сопоставим характеристический вектор $v_i=\{l, v_{i1}, v_{i2}, \ldots\}$, где v_{ik}—число соседей k-го класса у i-ой вершины, l—номер класса, к которому принадлежит i-ая вершина. Теперь снова разобьем вершины на классы в соответствии с новыми характеристическими векторами, упорядоченными лексикографически, и т. д. Заметим, что если вершины a и b на некотором шаге принадлежали разным классам и было выполнено условие $a<b$, то и в дальнейшем это условие всегда будет выполнено. Отсюда следует, что описанный процесс обрывается не позже, чем через n шагов—и либо все вершины относятся к разным классам (т. е. построена каноническая нумерация), либо дальнейшего разбиения на классы не происходит.

В случае, если Г—ориентированный мультиграф, возьмем в качестве характеристического вектора v_i упорядоченную i-ую строку матрицы $A(\Gamma)$ (считая при этом, что диагональный элемент предшествует всем остальным). Вместо различных элементов a_{ij} введем различные независимые переменные $x_1, x_2, \ldots$, упорядочив их в соответствии с порядком среди a_{ij}. Полученную таким способом матрицу обозначим $X=X(\Gamma)$. При очередном разбиении вершин на классы к одному классу, как и прежде, отнесем вершины с одинаковым характеристическим вектором; при этом k-ая компонента вектора v_i есть по определению сумма элементов i-ой строки матрицы $X(\Gamma)$, соответствующих вершинам k-го класса (предыдущего разбиения). Матрица $X(\Gamma)$ разбивается, таким образом, на блоки, в каждом из которых мы можем ввести свои независимые переменные и т. д. (точное определение этих операций см. п. 6, операции α_1, β_1).

Заметим, что описанная до сих пор процедура аналогична методам, изложенным в [1] и [2].

Для дальнейшего разбиения вершин на классы рассмотрим элемент u_{ij} матрицы $U=X\cdot X'$, где X'—матрица, полученная из X заменой переменных $x_1, x_2, \ldots$ переменными $x'_1, x'_2, \ldots$, причем все переменные $x_1, x_2, \ldots, x'_1, x'_2, \ldots$ независимы. Элемент u_{ij} является многочленом второй степени от $x_1, x_2, \ldots, x'_1, x'_2, \ldots$ Если теперь обозначить различные многочлены различными новыми независимыми переменными, то к полученной матрице снова можно будет применять все описанные выше операции и т. д., до тех пор, пока и этот процесс не оборвется (см. п. 6, операции α_2, β_2, β_3).

2. Геометрически введение независимых переменных и матрицы $U=X\cdot X'$ означает следующее. Если наряду с ребрами данного графа Г рассматривать ребра дополнительного графа $\overline{\Gamma}$, то на первом же шаге нашей процедуры этим ребрам будут соответствовать наверняка разные переменные— т. е. произойдет раскраска ребер данного и дополнительного графов в разные цвета. В дальнейшем каждое новое введение дополнительных переменных вводит новую раскраску ребер, причем 1) ребра, окрашенные на каком-то шаге по разному, в дальнейшем также будут окрашены по-разному; 2) разбиение вершин на классы производится в соответствии с количеством ребер каждого цвета, исходящих из вершины.

Известно, далее, что элемент $a_{ij}\in A^2$, $A=A(\Gamma)$, где Г— неориентированный граф без кратных связей, равен числу путей длины 2, ведущих из i-й вершины в j-ую. Аналогично, коэффициент при $x_k x'_l$ в многочлене $u_{ij}\in U=X\cdot X'$ равен числу путей, ведущих из i-й вершины в j-ую по ребрам сначала k-го, а затем l-го цвета.

3). Дальнейшая процедура приведения графа к каноническому виду использует применение описанных выше операций к матрице, полученной из $X(\Gamma)$ вычеркиванием некоторого ее столбца и соответствующей строки. Если для матриц порядка $k\leqslant n-1$ приведение к каноническому виду определено, то возникает новая возможность для разбиения вершин на классы: старшей считается та вершина, после вычеркивания которой получается лексикографически старший канонический вид оставшегося графа (см. операции α_3, α_4, β_4, Ы п. 7). Очевидно, что и такое разбиение на некотором шаге оборвется. Доказано, что любые две вершины a и b, отнесенные к одному классу при последнем разбиении, эквивалентны, т. е. существует автоморфизм графа Г, переводящий a в b и сохраняющий отношения смежности.

Рассмотрим снова такую матрицу $X=X(\Gamma)$, что в матрице $U=X\cdot X'$ на месте одинаковых переменных матрицы X стоят одинаковые многочлены. Матрица $X(\Gamma)$ является тогда общей точкой некоторой матричной алгебры $\mathfrak{A}(\Gamma)$, т. е. если задано кольцо K (например, кольцо целых чисел Z или поле рациональных чисел Q), то множество матриц, полученных подстановкой в X вместо ее переменных элементов кольца K образует алгебру $\mathfrak{A}_K(\Gamma)=\mathfrak{A}(\Gamma)\otimes K$. Алгебра $\mathfrak{A}(\Gamma)$ является, очевидно, инвариантом графа. Некоторые соотношения между

Further results for particular graph classes

Every graph G of tree-width k is definable with $4k+4$ variables and quantifier depth $O(k \log n)$ [Grohe, V. 2006].

Every 3-connected planar graph G is definable with 15 variables and quantifier depth $O(\log n)$ [V. 2007].

Every interval graph G is definable with 15 variables and quantifier depth $O(\log n)$ cf. [Köbler, Kuhnert, Laubner, V. 2011].

Open problem

Is it true that, for each F, if G excludes F as a minor, then G is definable not only with finitely many variables but also with logarithmic quantifier depth?

The speed of stabilization — a worst-case analysis

We have seen that $\mathrm{Stab}_1(P_n) \geq \frac{1}{2}\,n - 1$.

Theorem (Krebs, V. 2015)

There are n-vertex graphs G and H that are distinguishable with 2 variables but only with quantifier depth more than $n - 8\sqrt{n}$.
Consequently, there are n-vertex graphs G with

$$\mathrm{Stab}_1(G) = (1 - o(1))\,n.$$

This has some applications...

Universal covers and distributed computing

Theorem (Leighton 82, Immerman and Lander 1990)

The following three conditions are equivalent:

- G *and* H *are indistinguishable by any 2-variable sentence with counting quantifiers;*
- G *and* H *are indistinguishable by Color Refinement.*
- G *and* H *have isomorphic universal covers.*

- Our result provides n-vertex G and H with non-isomorphic universal covers that are isomorphic when truncated at depth $(2 - o(1))n$.
- This solves Norris's problem (1995) and implies that the standard upper bound of $2n$ for the communication round complexity of a certain class of distributed algorithms is essentially tight.

The speed of stabilization — a worst-case analysis

Theorem (Kiefer, Schweitzer 2016)

For all G,

$$\mathrm{Stab}_2(G) = o(n^2).$$

Specifically, $\mathrm{Stab}_2(G) = O(n^2/\log n)$.

Theorem (Fürer 2001)

For each $k \geq 1$, there are G such that

$$\mathrm{Stab}_k(G) = \Omega(n).$$

Theorem (Berkholz, Nordström 2016)

If k is sufficiently large, then there are k-ary relational structures S such that

$$\mathrm{Stab}_k(S) = n^{\Omega(k/\log k)}.$$

Thank you for your attention!