A logical approach to Isomorphism Testing and Constraint Satisfaction

Oleg Verbitsky

Humboldt University of Berlin, Germany

ESSLLI 2016, 15–19 August

Part 5: $\mathrm{FO}^2_{\#}$ and Distributed Computing.

Outline

Outline

Three pillars

1980

- L. Babai, P. Erdős, and S.M. Selkow. Random graph isomorphism. *SIAM J. Comput.*
- D. Angluin. Local and global properties in networks of processors. *STOC'80.*

1990

- N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity Theory Retrospective*, Springer.

Outline

Color refinement algorithm (formal definition)

$$\begin{aligned} C^1(v) &= \deg v \\ C^{i+1}(v) &= \{\!\{\, C^i(u) : u \in N(v) \,\}\!\} \end{aligned}$$

Exercise

If ϕ is an isomorphism from G to H, then $C^i(v) = C^i(\phi(v))$.

Therefore,

$$G \cong H \quad \Longrightarrow \quad \{\!\{\, C^i(u) \,\}\!\}_{u \in V(G)} = \{\!\{\, C^i(v) \,\}\!\}_{v \in V(H)}$$

Color Refinement accepts G and H as isomorphic

iff

the equality is true for all i.

- The output "non-isomorphic" is always true.
- The output "isomorphic" can be wrong.

Color refinement and 2-variable counting logic

Immerman and Lander:

The following three conditions are equivalent:

- Color refinement distinguishes G and H;
- G and H are distinguishable in two-variable first-order logic with counting quantifiers.
- Spoiler has a winning strategy in the *2-pebble counting game* on G and H.

In particular, if color refinement distinguishes G and H in less than s rounds, then G and H are distiguishable with quantifier depth s.

Question

Suppose that color refinement distinguishes n-vertex G and H. How many refinement rounds does it need?

- Just 2 for almost all G (Babai, Erdős, Selkow).
- What about the worst case?

A related question

How large can $D^2_{\#}(G)$ be for G definable in $\mathrm{FO}^2_{\#}$?

We already know that n rounds always suffice.

At least $n/2 - 2$ rounds are sometimes needed:
e.g., on P_n and $P_{n-3} + C_3$.

- Thus, the optimum is between $n/2$ and n. Where?

Theorem (Krebs, V. 2015)

There are n-vertex G and H distinguishable in 2-variable counting logic but only with quantifier depth $(1 - o(1))n$.

Corollary

There are n-vertex G and H such that color refinement needs $(1 - o(1))n$ refinement rounds to distinguish them.

Moreover,

color refinement stabilizes on the disjoint union $G + H$ in $(2 - o(1))n$ rounds

(despite the stabilization on each of G and H is reached in less than n rounds).

Outline

Basic concepts

- A network = a graph G
- A processor (a finite automaton) = a node in G
- The initial states are identical for nodes of the same degree
- In a unit of time — a message exchange along each edge

Examples of problems.

Leader election: Exactly one processor has to come in a distinguished state "elected".

Network topology recognition: One of the processors (or all of them) has to come in a special state iff G has a specified property (for example, G is bipartite, planar, ...).

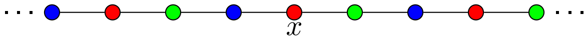
Covering maps — basic definitions

Let G and H be connected.
α is a covering map from H to G if α is

- a homomorphism from H onto G,
- a bijection from $N(v)$ onto $N(\alpha(v))$ for each $v \in V(H)$.

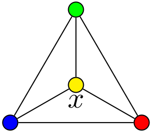
We say that H is a covering graph of G or that H covers G.

$U_x(G)$ is the "unfolding" of G from x into an (infinite) tree.

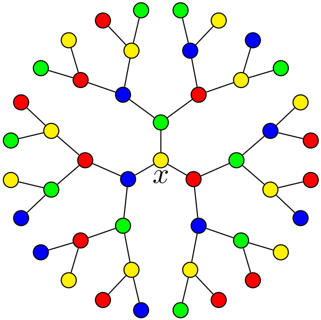


$U_x(G)$ covers any covering graph of G and is called a universal cover of G.

Another example of a universal cover



G



$U_x^4(G)$

Applications in distributed computing: an example

Lemma

Let $\alpha : H \to G$ be a covering map for the networks G and H. Then the processors v and $\alpha(v)$ will be always in the same state.

Lemma

Planar graphs are not closed under covering maps.

Corollary

Planarity is not recognizable by local computations.

Color refinement in distributed computing

Angluin:

The following conditions are equivalent.

- G and H have a common covering graph.
- $U(G) \cong U(H)$
- $\{C^i(u) : u \in V(G)\} = \{C^i(v) : v \in V(H)\}$, for all i.

Relation to $\mathrm{FO}^2_\#$

Angluin + Immerman & Lander + Ramana et al.

If G and H have equally many vertices, then the following conditions are equivalent.

- G and H are indistinguishable in $\mathrm{FO}^2_\#$.
- G and H are fractionally isomorphic.
- G and H are indistinguishable by Color Refinement.
- G and H have isomorphic universal covers.

Truncated universal covers

$U_x(G) \cong U_y(H) \implies$ the processors x and y are all the time in equal states (i.e., indistinguishable by local computations).

Let $U_x^t(G)$ denote the rooted tree $U_x(G)$ truncated at depth t.

$U_x^t(G) \cong U_y^t(H) \implies x$ and y are in equal states up to time t.

Truncated universal covers

$U_x(G) \cong U_y(H) \implies$ the processors x and y are all the time in equal states (i.e., indistinguishable by local computations).

Let $U_x^t(G)$ denote the rooted tree $U_x(G)$ truncated at depth t.

$U_x^t(G) \cong U_y^t(H) \implies x$ and y are in equal states up to time t.

Lemma

$U_x^t(G) \cong U_y^t(H)$ iff $C^t(x) = C^t(y)$.

Lemma

$U_x^t(G) \cong U_y^t(H)$ *iff* $C^t(x) = C^t(y)$.

Exercise

Prove it by induction on t.

Hint

Prove first the following Tree Reconstruction Lemma:

Let T and S be trees, $x \in V(T)$, $y \in V(S)$, $N(x) = \{x_1, \ldots, x_k\}$, and $N(y) = \{y_1, \ldots, y_k\}$. Then

$$T_x^r \cong S_y^r \text{ and } T_{x_i}^r \cong S_{y_i}^r \text{ for all } i \le k \implies T_x^{r+1} \cong S_y^{r+1}.$$

Exercise

Apply it for another proof that every tree is definable in $\mathrm{FO}_{\#}^2$.

Outline

What truncation depth is enough?

Lemma

$U_x^t(G) \cong U_y^t(H)$ *iff* $C^t(x) = C^t(y)$.

By the color stabilization argument:
if G and H are two graphs with at most n vertices each, then

$$U_x^{2n-1}(G) \cong U_y^{2n-1}(H) \implies U_x(G) \cong U_y(H).$$

Norris's question (1995)

Can $2n-1$ be improved to n in this implication?
(Yes if $G = H$)

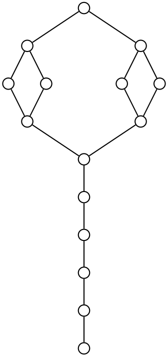
Theorem (Krebs, V. 2015)

There are n-vertex graphs G and H with vertices $x \in V(G)$ and $y \in V(H)$ such that

1. $U_x^{2n-16\sqrt{n}}(G) \cong U_y^{2n-16\sqrt{n}}(H)$ *while* $U_x(G) \ncong U_y(H)$;
2. $D^2_{\#}(G,H) > n - 8\sqrt{n}$.

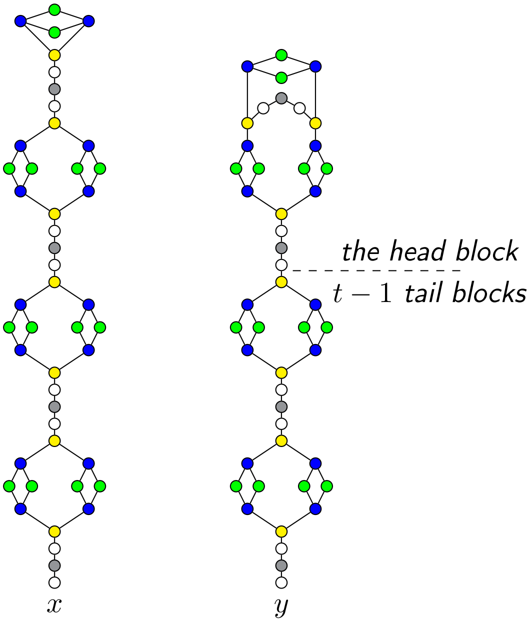
Construction of $G = G_{s,t}$ and $H = H_{s,t}$

- Each graph is a chain of t blocks:
  - one head block,
  - $t - 1$ tail blocks
- All tail blocks are identical and have $s + 10$ vertices.



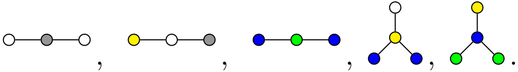
The tail block for $s = 5$

.



The graphs $G_{s,t}$ and $H_{s,t}$ for $s = 3$, $t = 3$.

- We distinguish $\lceil s/2 \rceil + 3$ types of vertices, presented by auxiliary colors.
- This auxiliary coloring is almost stable: All vertex neighborhoods (excepting for x and y) are



- $C^i(x) = C^i(y)$ iff Duplicator has a winning strategy in the i-round bisimulation version of the Immerman-Lander game on (G, x, H, y). This is so for $i = 2t(s+5) - 2$, while Spoiler has a winning strategy for larger i.
- The graphs have $n = (t+1)(s+10) - 5$ vertices. Take $s = 2t + 1$.

Conclusion

- The universal cover $U_x(G)$ contains all knowledge about the network G available to a particular party x.
- A large bunch of distributed algorithms is based on computing the isomorphism type of $U_x(G)$ by the party x.
- The bound of $2n$ is a standard upper bound for the communication round complexity of such algorithms.
- Our solution of Norris's problem implies that this bound is tight up to a term of $o(n)$.
- This seems to be the first application of FMT in the field.

Conclusion

- The universal cover $U_x(G)$ contains all knowledge about the network G available to a particular party x.
- A large bunch of distributed algorithms is based on computing the isomorphism type of $U_x(G)$ by the party x.
- The bound of $2n$ is a standard upper bound for the communication round complexity of such algorithms.
- Our solution of Norris's problem implies that this bound is tight up to a term of $o(n)$.
- This seems to be the first application of FMT in the field.

Open problem

Can the lower bound of $2n - O(\sqrt{n})$ be improved to $2n - O(1)$?

Outline

References

- Andreas Krebs and Oleg Verbitsky. Universal covers, color refinement, and two-variable counting logic: Lower bounds for the depth. LICS 2015.