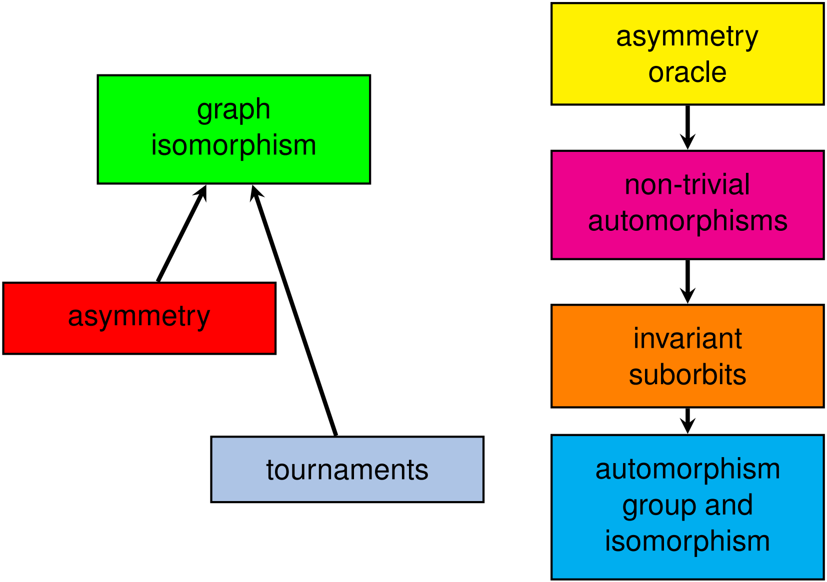
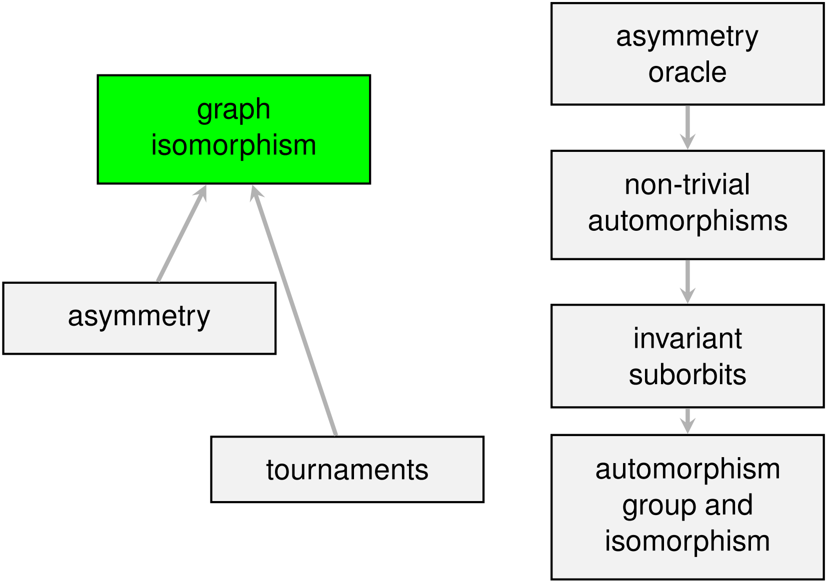
Graph isomorphism and asymmetric graphs

Pascal Schweitzer

Ghent Graph Theory Workshop 2017
August 18th, Ghent

RWTH AACHEN UNIVERSITY

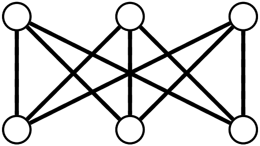

graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

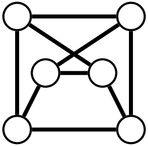

graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

The graph isomorphism problem

Two graphs are isomorphic if there is a bijection of vertices that preserves adjacency.

Isomorphic graphs





The graph isomorphism problem

Two graphs are isomorphic if there is a bijection of vertices that preserves adjacency.

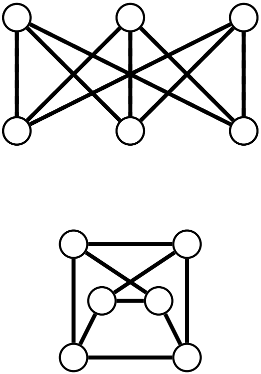
Isomorphic graphs

The graph isomorphism problem

Two graphs are isomorphic if there is a bijection of vertices that preserves adjacency.

Graph isomorphism (GI): Algorithmic task to decide whether two graphs are isomorphic.

Isomorphic graphs



Unknown complexity

Is there an efficient algorithm for graph isomorphism?

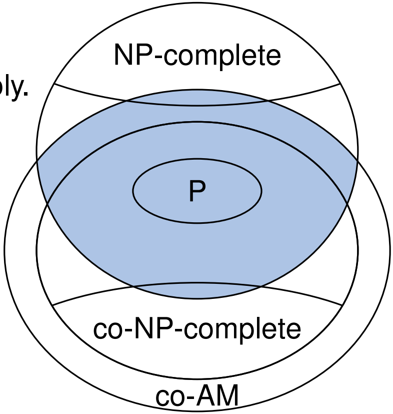
Is there an efficient algorithm for graph isomorphism?

known

- *GI* $\in$ NP
- *GI* NP-hard $\Rightarrow$ SAT quasi-poly. ($\Rightarrow$ ETH false)
- *GI* NP-hard $\Rightarrow$ PH collapses (*GI* $\in$ co-AM)

unknown

- *GI* $\in$ P?
- Is *GI* NP-complete?
- *GI* $\in$ co-NP?



Current status

$$2^{O(\sqrt{(n \log n)})} \quad \Rightarrow \quad 2^{(\log(n)^c)}$$

[Babai using Luks,Zemlyachenko] (1981) [Babai] (2015)

Two major **open subcases**:

- group isomorphism (given by multiplication table)
- tournament isomorphism

Both subcases have $2^{O(\log(n)^2)}$-time algorithms.

Problems equivalent to isomorphism

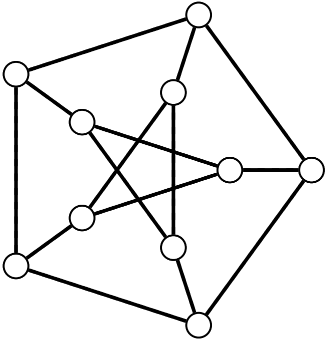
These following problems are **polynomially equivalent**:

- GI: the graph isomorphism problem
- col-GI: isomorphism problem of colored graphs
- ISO: isomorphism of general combinatorial objects
- $\mathrm{Aut}(G)$: compute generating set for automorphism group
- $|\mathrm{Aut}(G)|$: determine the size of $\mathrm{Aut}(G)$.

The graph isomorphism problem is actually the problem of detecting symmetries of combinatorial objects.

Automorphisms

An automorphism is an isomorphism from a graph to itself.



The automorphism group captures the intrinsic symmetries of the graph.

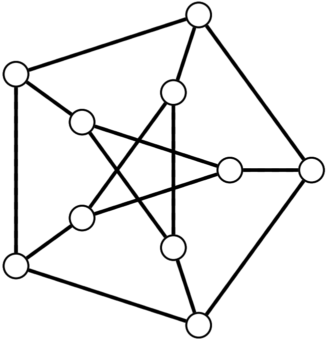
Automorphisms

An automorphism is an isomorphism from a graph to itself.

The automorphism group captures the intrinsic symmetries of the graph.

Automorphisms

An automorphism is an isomorphism from a graph to itself.



The automorphism group captures the intrinsic symmetries of the graph.

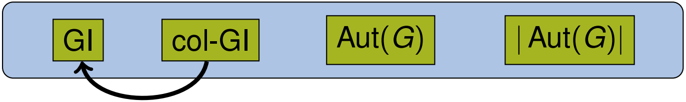
An automorphism is an isomorphism from a graph to itself.

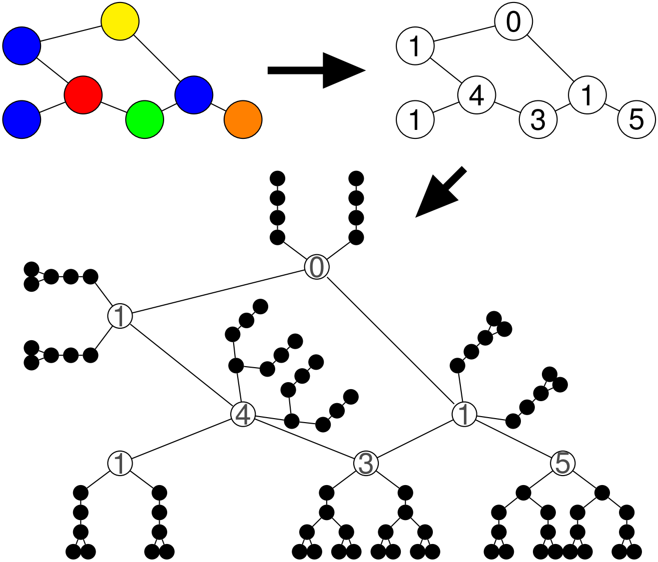
The automorphism group captures the intrinsic symmetries of the graph.

Some reductions

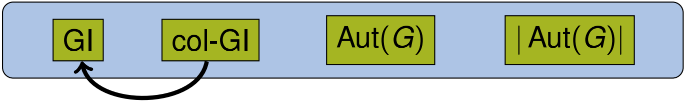
GI   col-GI $\mathrm{Aut}(G)$ $|\mathrm{Aut}(G)|$

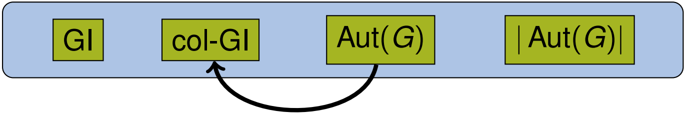
Some reductions



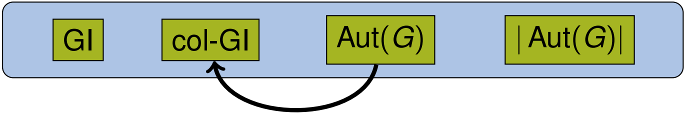

0
1
4
1
3
1
5
0
1
4
1
3
1
5

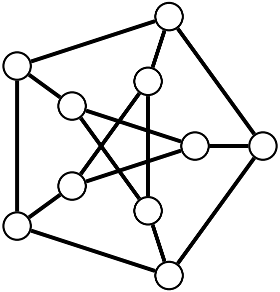
Some reductions

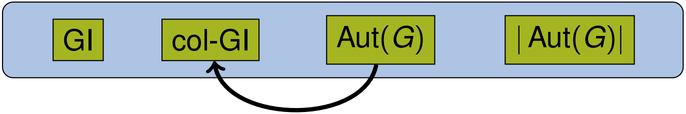


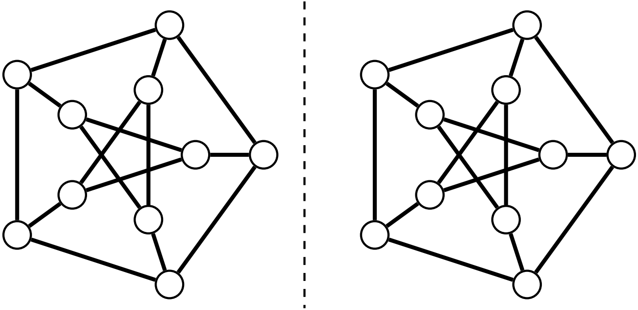

GI
col-GI
Aut(G)
$|\text{Aut}(G)|$

Some reductions

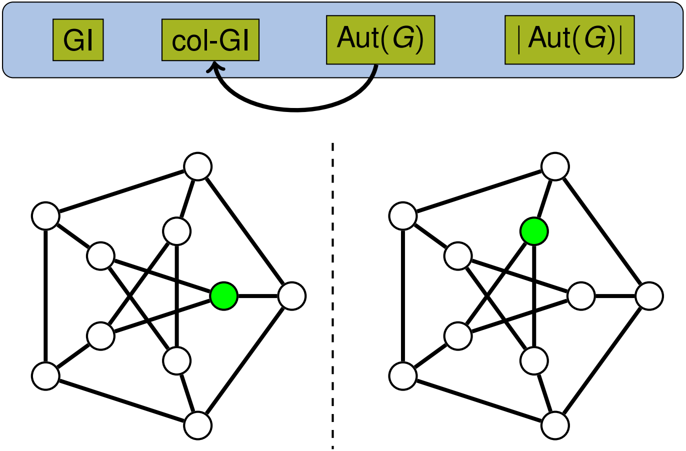




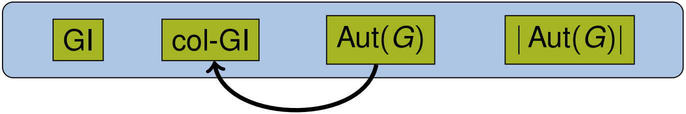

GI
col-GI
Aut(G)
|Aut(G)|

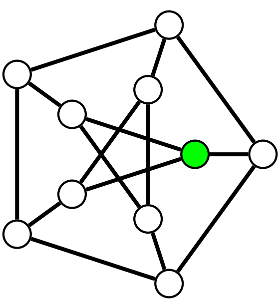


Some reductions



Some reductions

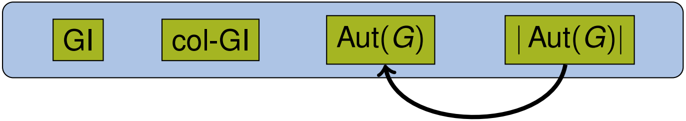




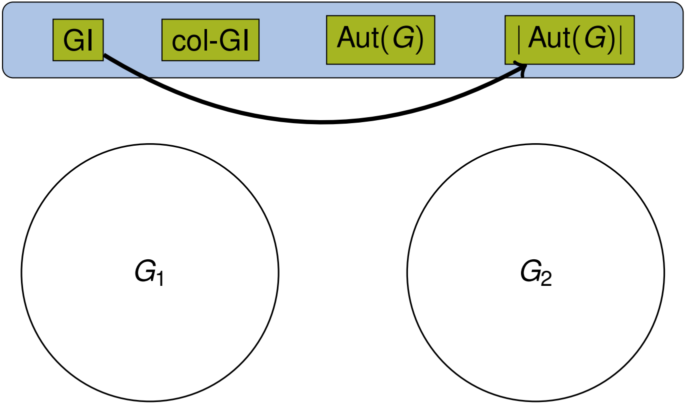
Some reductions

GI col-GI $\mathrm{Aut}(G)$ $|\mathrm{Aut}(G)|$

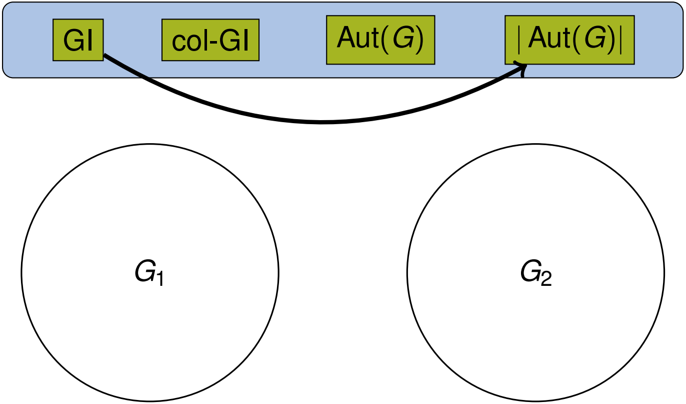
Some reductions



Some reductions

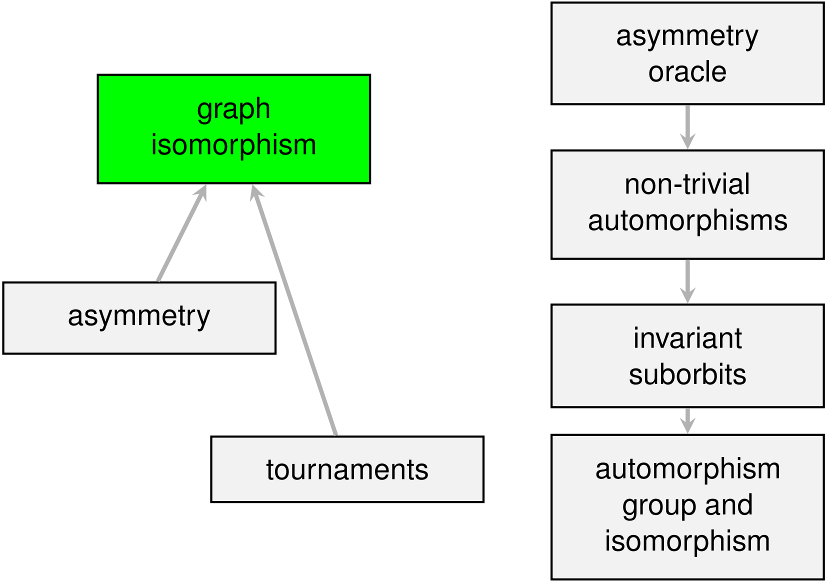


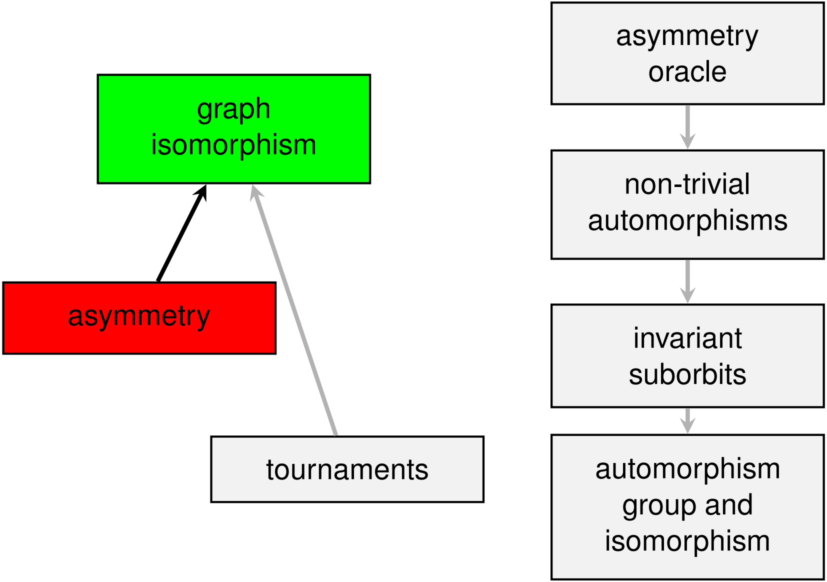
Some reductions



W.l.o.g. G_1, G_2 connected.

$$|\mathrm{Aut}(G_1 \cup G_2)| = \begin{cases} 2 \cdot |\mathrm{Aut}(G_1)| \cdot |\mathrm{Aut}(G_2)| & \text{if } G_1 \cong G_2 \\ |\mathrm{Aut}(G_1)| \cdot |\mathrm{Aut}(G_2)| & \text{otherwise.} \end{cases}$$


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Worst Case instances for IR algorithms

What is the running time of IR algorithms (such as nauty, or traces, bliss, saucy, conauto)?

What is the running time of IR algorithms (such as nauty, or traces, bliss, saucy, conauto)?

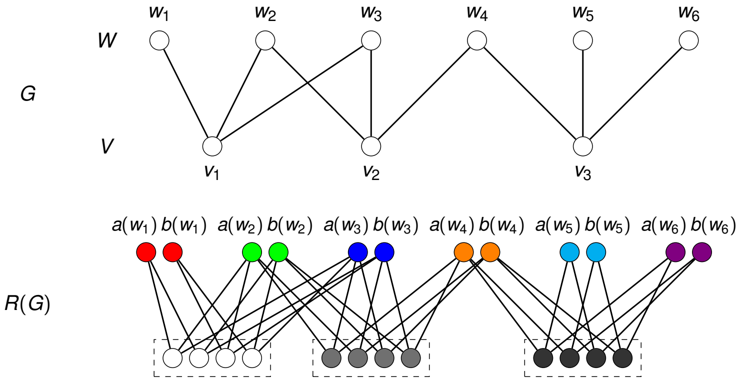
- In the worst case IR algorithms have exponential running time.
[Neuen, S.] (2017+)

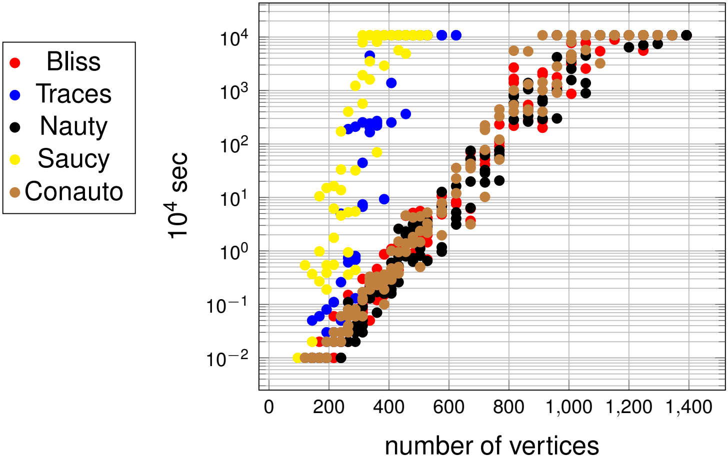
Worst Case instances for IR algorithms

What is the running time of IR algorithms (such as nauty, or traces, bliss, saucy, conauto)?

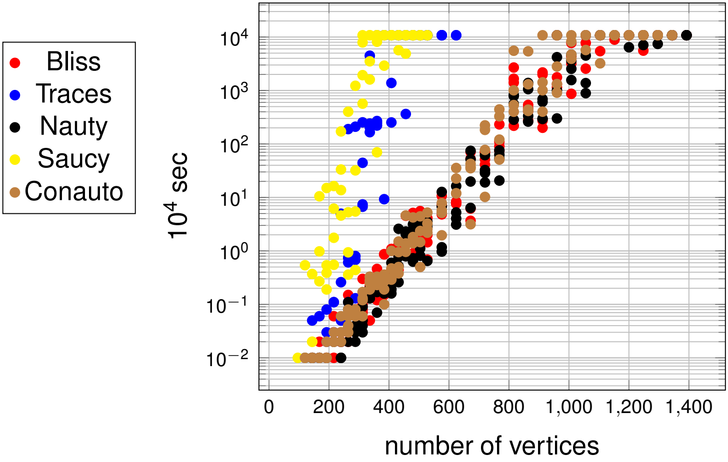
- In the worst case IR algorithms have exponential running time.

[Neuen, S.] (2017+)




Bliss
Traces
Nauty
Saucy
Conauto
10^4 sec
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
0
200
400
600
800
1,000
1,200
1,400
number of vertices

Benchmark graphs



These benchmarks are asymmetric graphs (rigid).

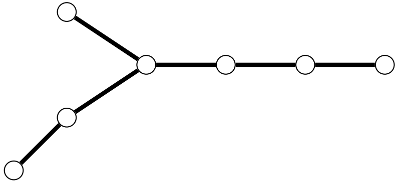
Graph asymmetry

A graph G is called asymmetric (or *rigid*) if it does not have a non-trivial automorphism (i.e., $|\operatorname{Aut}(G)| = 1$).

Graph asymmetry

A graph G is called asymmetric (or *rigid*) if it does not have a non-trivial automorphism (i.e., $|\mathrm{Aut}(G)| = 1$).

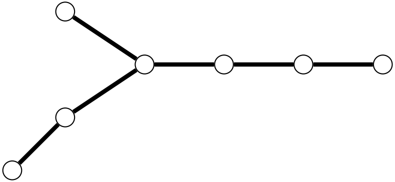
Example:



Graph asymmetry

A graph G is called asymmetric (or *rigid*) if it does not have a non-trivial automorphism (i.e., $|\mathrm{Aut}(G)| = 1$).

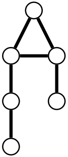
Example:



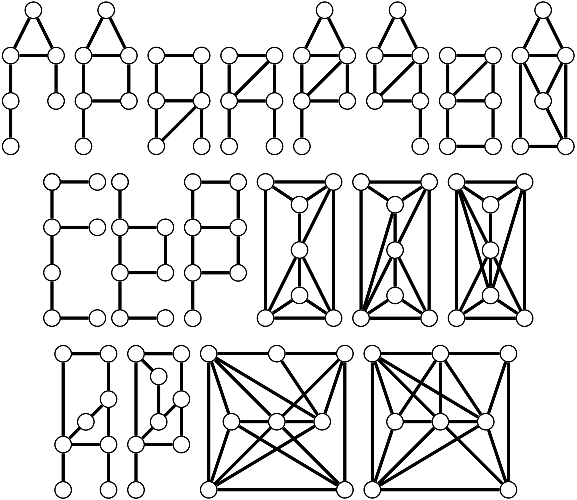
Graph asymmetry denoted GA is the algorithmic task to decide whether a given graph is asymmetric.

(Many authors call this the *graph automorphism problem*.)

Absence of symmetry



Absence of symmetry



Thm. exactly 18 minimal asymmetric graphs

Nešetřil Conjecture [S., Schweitzer] (2017+)

Asymmetry vs isomorphism

GI col-GI $\mathrm{Aut}(G)$ $|\mathrm{Aut}(G)|$

Asymmetry vs isomorphism

GI

col-GI

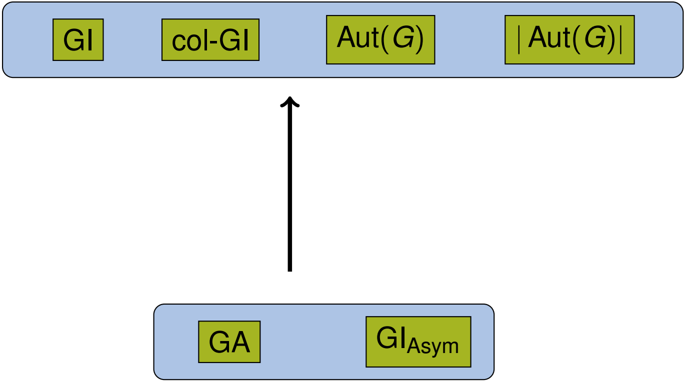
Aut(G)

$|\mathrm{Aut}(G)|$

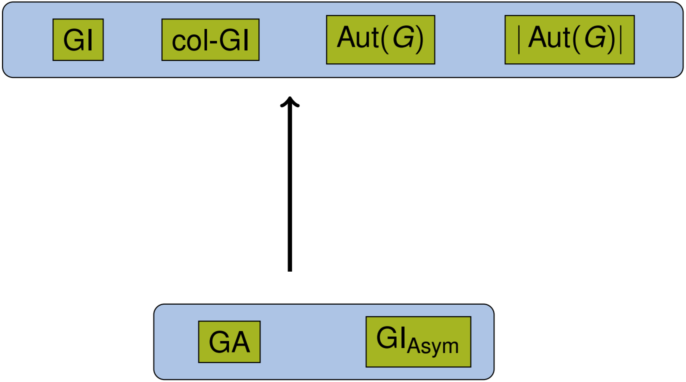
GA

$\mathrm{GI}_{\mathrm{Asym}}$

Asymmetry vs isomorphism



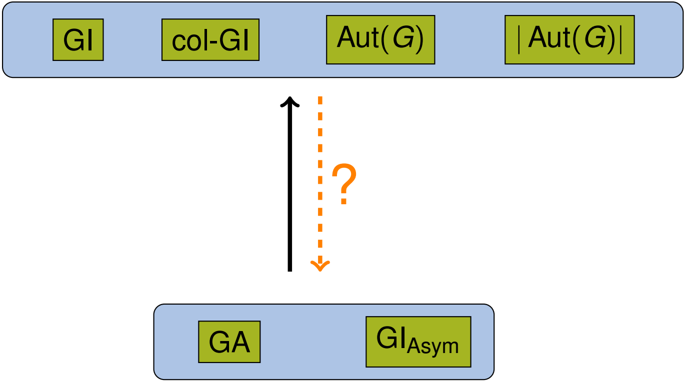
Asymmetry vs isomorphism



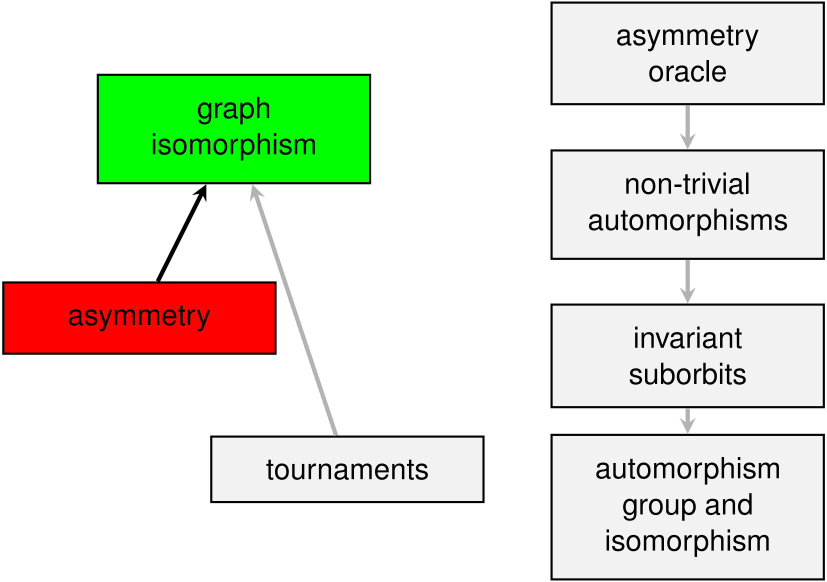
Open question:

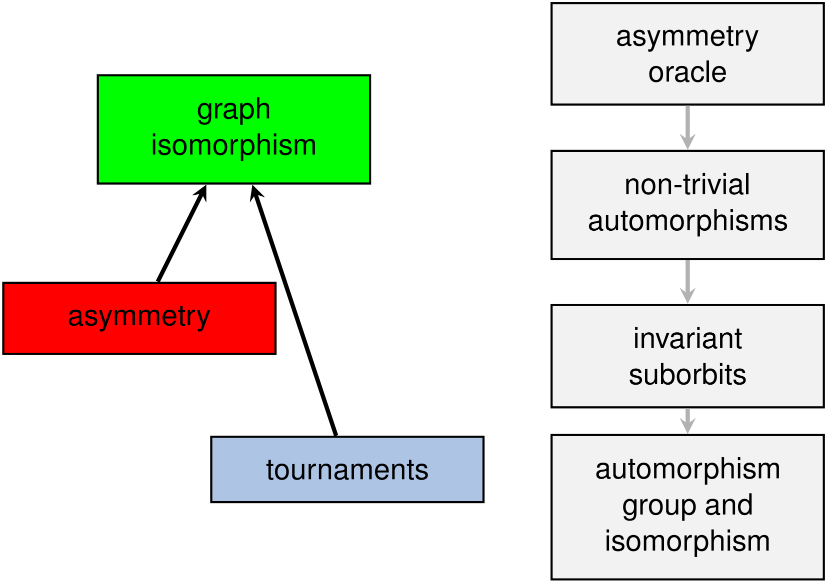
Is it harder to find all symmetries than to detect asymmetry?

Asymmetry vs isomorphism



Open question:
Is it harder to find all symmetries than to detect asymmetry?


asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism
graph isomorphism
asymmetry
tournaments

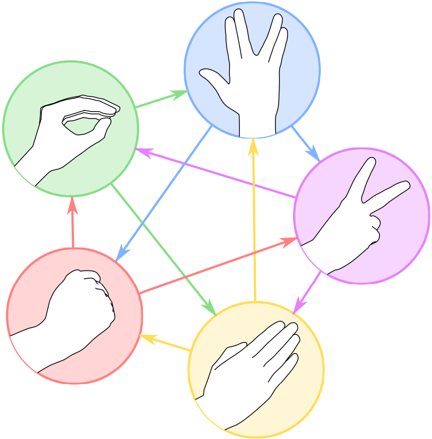

graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Tournaments

A tournament is an oriented complete graph.
(exactly one directed edge between every pair of vertices)

Tournaments

A tournament is an oriented complete graph.
(exactly one directed edge between every pair of vertices)



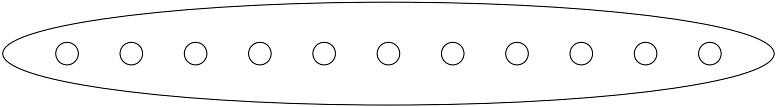
Symmetry problems for tournaments

$\mathrm{GI}_{\mathrm{Tour}}$ | $\text{col-GI}_{\mathrm{Tour}}$ | $\mathrm{Aut}(T)$ | $|\mathrm{Aut}(T)|$

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

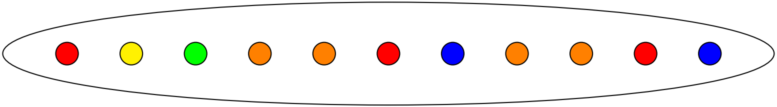
$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$



[Arvind, Das, Mukhopadhyay] (2010)

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

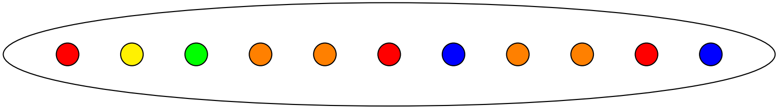


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

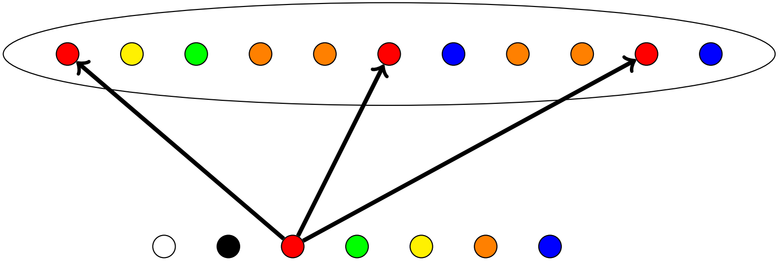


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

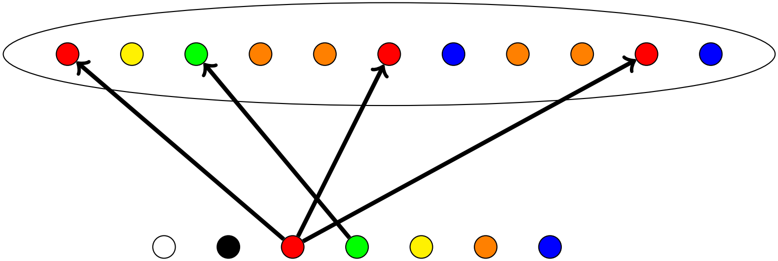


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

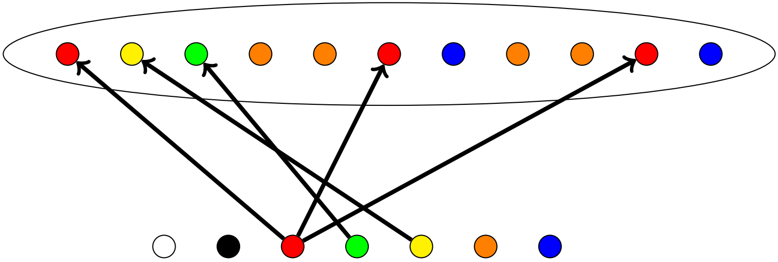


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

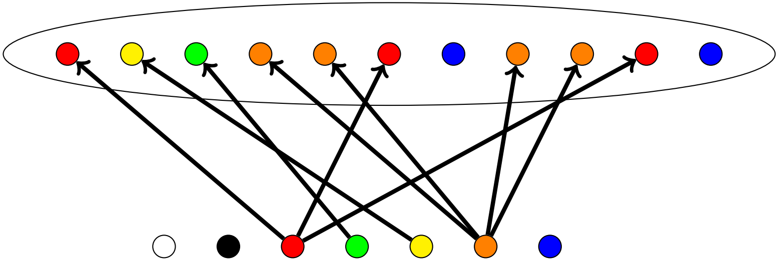


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

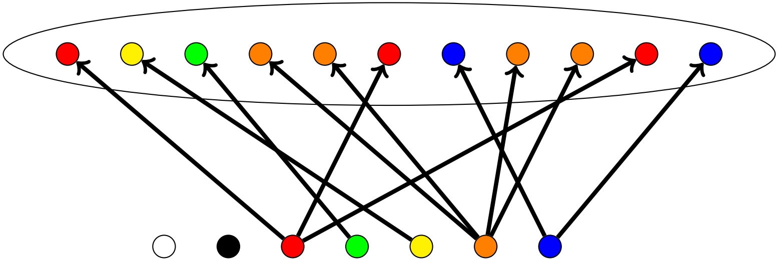
$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$



[Arvind, Das, Mukhopadhyay] (2010)

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

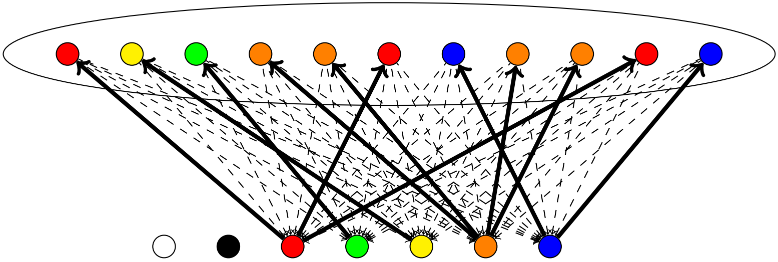


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

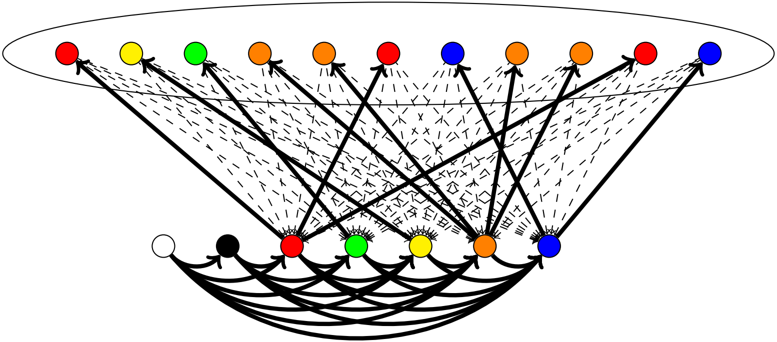


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

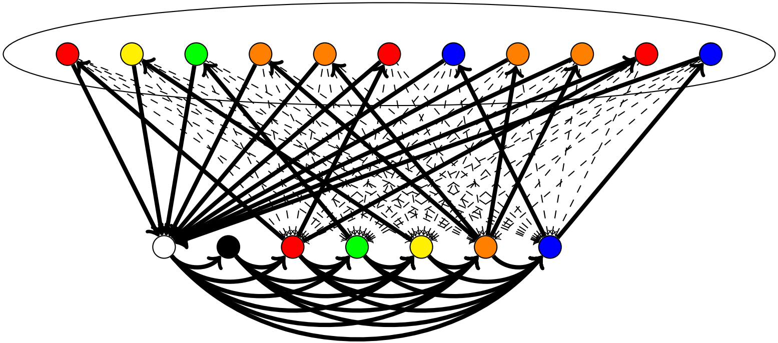


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

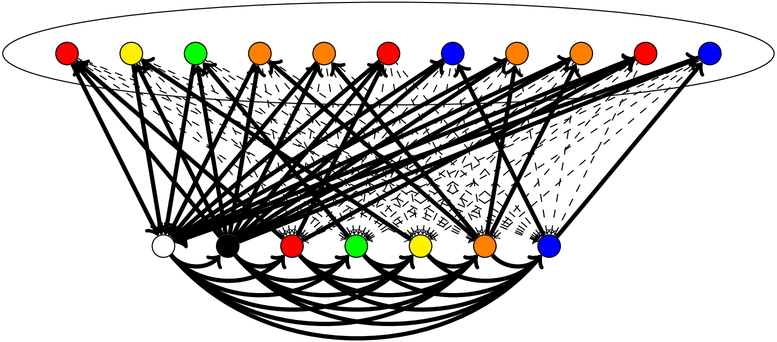


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

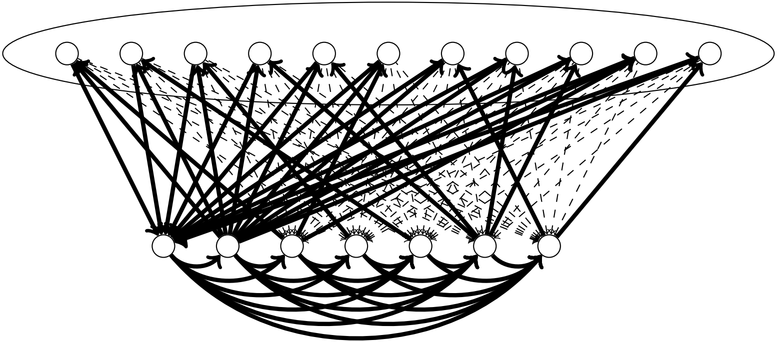


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$

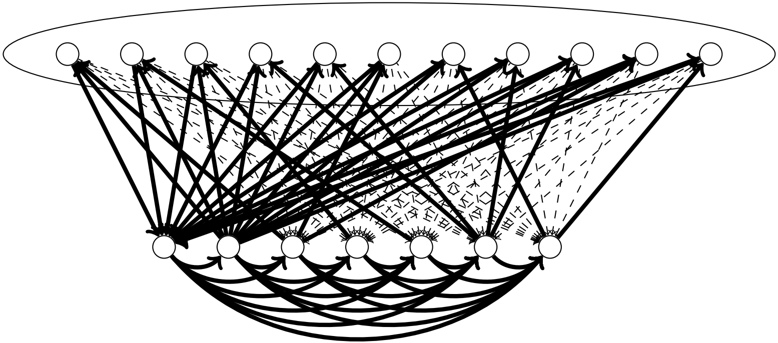


[Arvind, Das, Mukhopadhyay] (2010)

Removing colors for tournaments

- colored tournament isomorphism $\rightsquigarrow$ tournament isomorphism

$$\text{col-GI}_{\text{Tour}} \leq_m^p \text{GI}_{\text{Tour}}$$



[Arvind, Das, Mukhopadhyay] (2010)

- colored tournament asymmetry $\rightsquigarrow$ tournament asymmetry

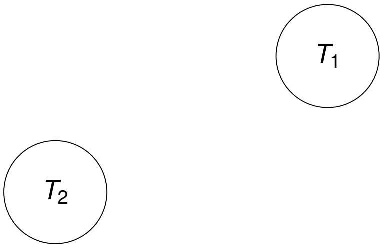
$$\text{col-GA}_{\text{Tour}} \leq_m^p \text{GA}_{\text{Tour}}$$

Alternative to disjoint union for tournaments

For tournaments we cannot form the **disjoint union.**
Instead we form the **triangle tournament** $\mathrm{Tri}(T_1, T_2)$.

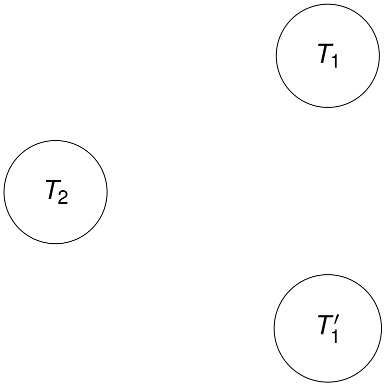
Alternative to disjoint union for tournaments

For tournaments we cannot form the **disjoint union.**
Instead we form the **triangle tournament** $\mathrm{Tri}(T_1, T_2)$.



Alternative to disjoint union for tournaments

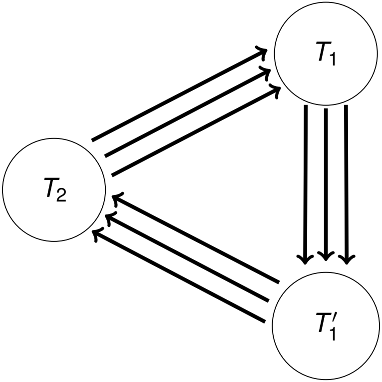
For tournaments we cannot form the **disjoint union.**
Instead we form the **triangle tournament** $\mathrm{Tri}(T_1, T_2)$.



$T_1 \cong T_1'$

Alternative to disjoint union for tournaments

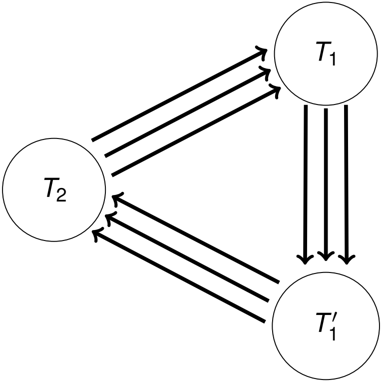
For tournaments we cannot form the **disjoint union.**
Instead we form the **triangle tournament** $\mathrm{Tri}(T_1, T_2)$.



$T_1 \cong T_1'$

Alternative to disjoint union for tournaments

For tournaments we cannot form the **disjoint union.**
Instead we form the **triangle tournament** $\mathrm{Tri}(T_1, T_2)$.



$T_1 \cong T_1'$

$$|\,\mathrm{Aut}(\mathrm{Tri}(T_1, T_2))| = \begin{cases} 3 \cdot |\,\mathrm{Aut}(T_1)|^2 \cdot |\,\mathrm{Aut}(T_2)| & \text{if } T_1 \cong T_2 \\ |\,\mathrm{Aut}(T_1)|^2 \cdot |\,\mathrm{Aut}(T_2)| & \text{otherwise.} \end{cases}$$

Asymmetry vs isomorphism for tournaments

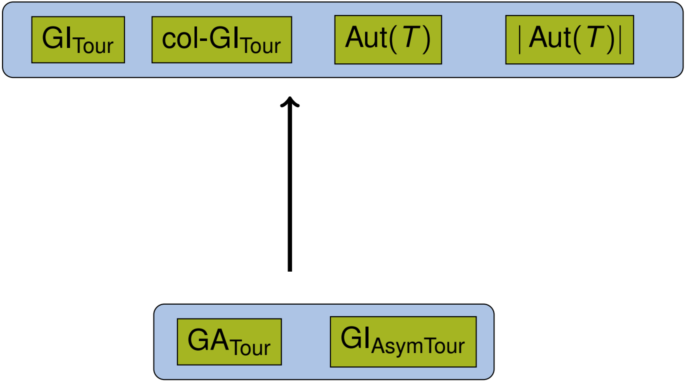
$\mathrm{GI}_{\mathrm{Tour}}$ $\quad$ $\mathrm{col\text{-}GI}_{\mathrm{Tour}}$ $\quad$ $\mathrm{Aut}(T)$ $\quad$ $|\mathrm{Aut}(T)|$

Asymmetry vs isomorphism for tournaments

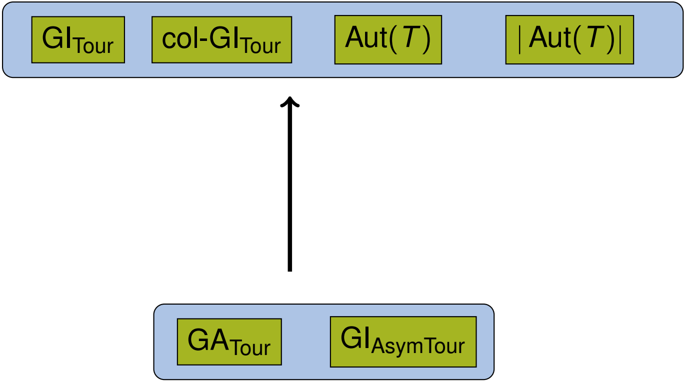
$\mathrm{GI}_{\mathrm{Tour}}$ $\text{col-GI}_{\mathrm{Tour}}$ $\mathrm{Aut}(T)$ $|\mathrm{Aut}(T)|$

$\mathrm{GA}_{\mathrm{Tour}}$ $\mathrm{GI}_{\mathrm{AsymTour}}$

Asymmetry vs isomorphism for tournaments



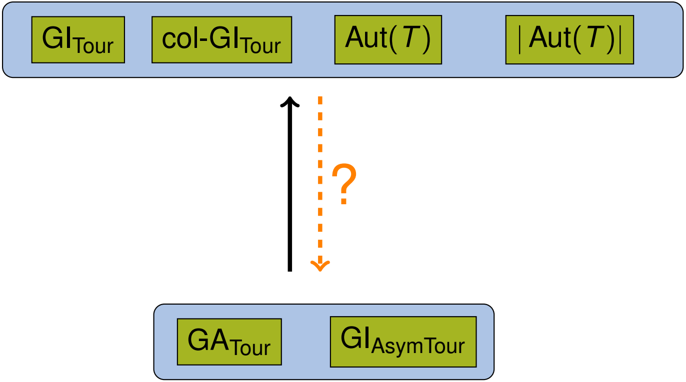
Asymmetry vs isomorphism for tournaments



Open question:

Is it harder to find all symmetries than to detect asymmetry?

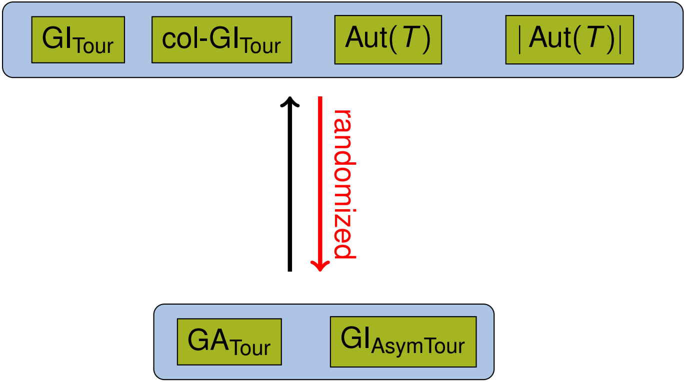
Asymmetry vs isomorphism for tournaments



Open question:

Is it harder to find all symmetries than to detect asymmetry?

Asymmetry vs isomorphism for tournaments



Open question:

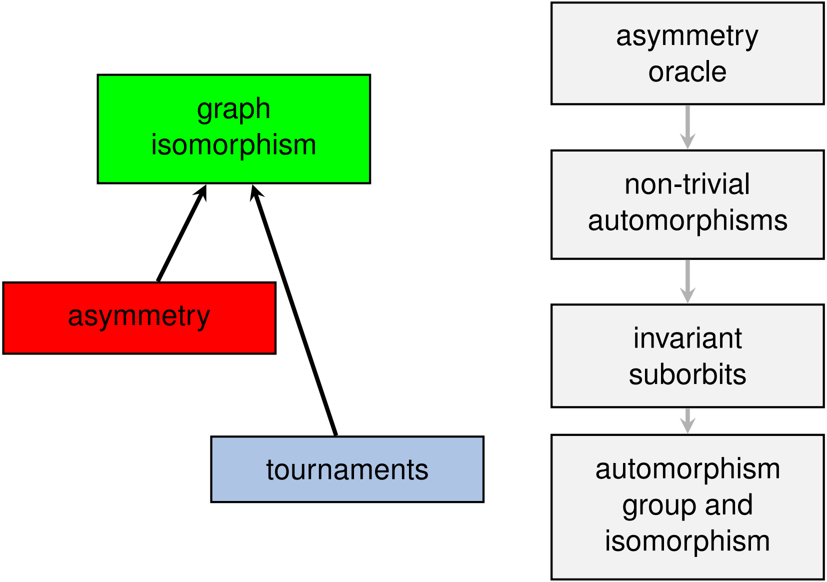
Is it harder to find all symmetries than to detect asymmetry?

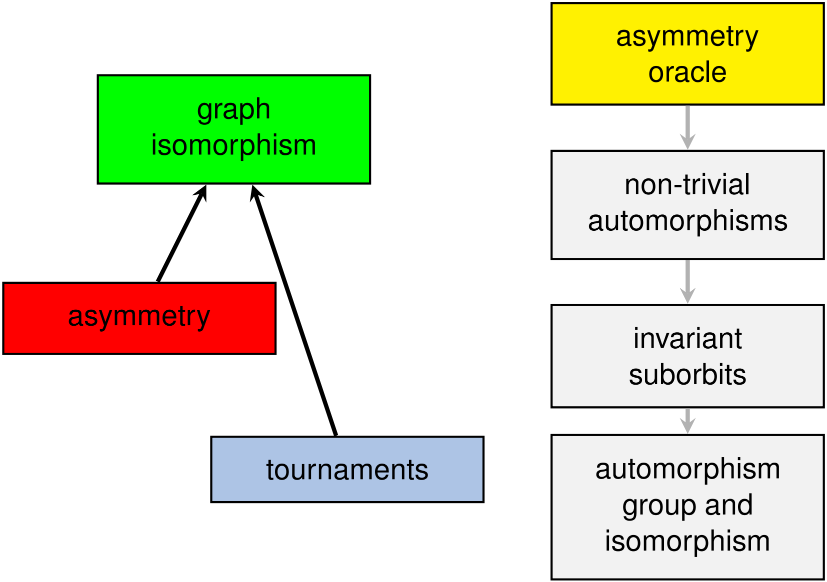
Main Result

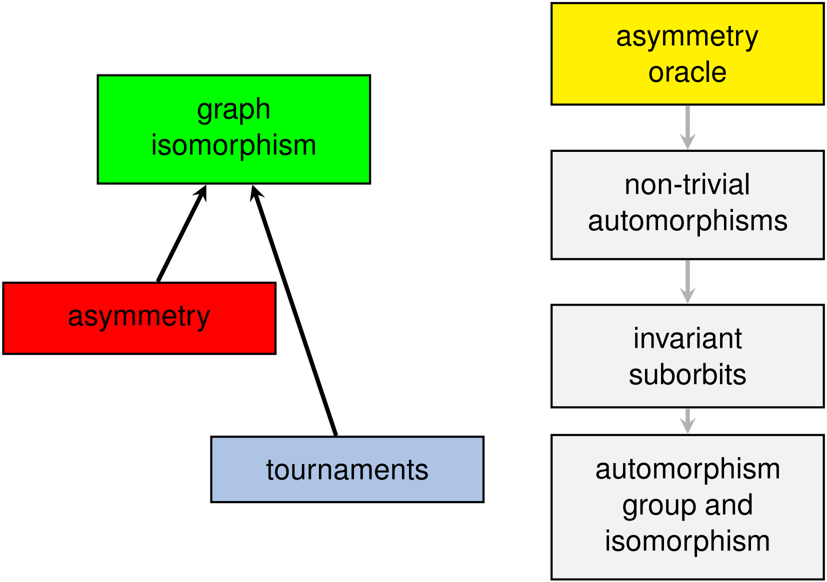
Theorem

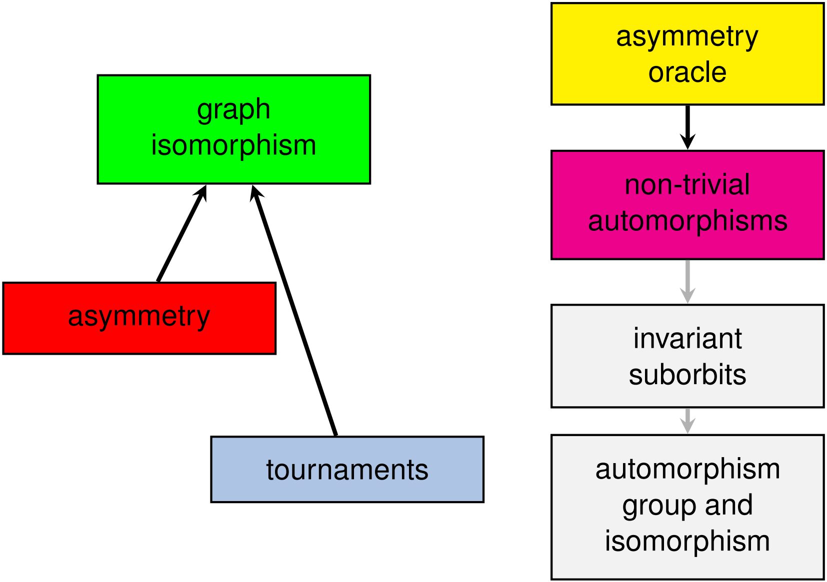
There is a polynomial-time randomized reduction from tournament isomorphism to tournament asymmetry.

Thus:
For tournaments finding all symmetries and detecting asymmetry are polynomially equivalent.


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Sampling automorphisms

Technique 1:

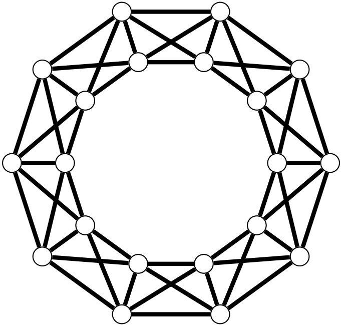
asymmetry test $\rightsquigarrow$ non-trivial automorphism sampler

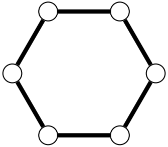
Technique 1:

asymmetry test $\rightsquigarrow$ non-trivial automorphism sampler

Strategy
- fix more and more vertices until graph is asymmetric
- make a copy of the graph
- undo last fixing in copy
- find alternative vertex to the vertex fixed last
- find isomorphism from original to copy

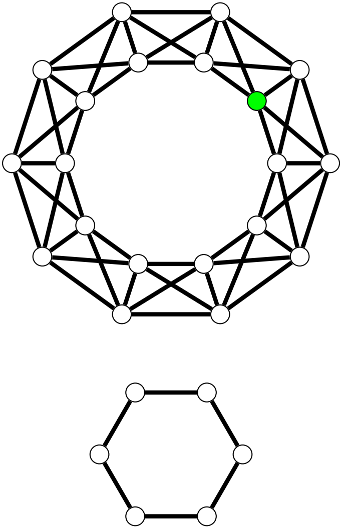
How to get automorphisms — Illustration





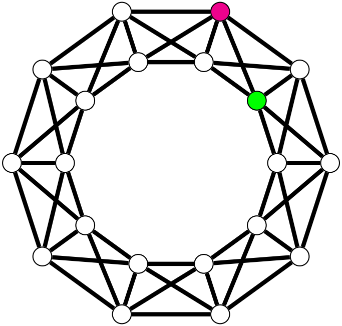
Automorphisms:

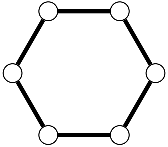
How to get automorphisms — Illustration



Automorphisms:

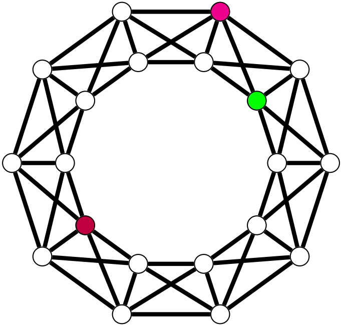
How to get automorphisms — Illustration

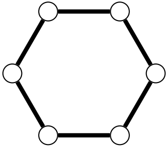




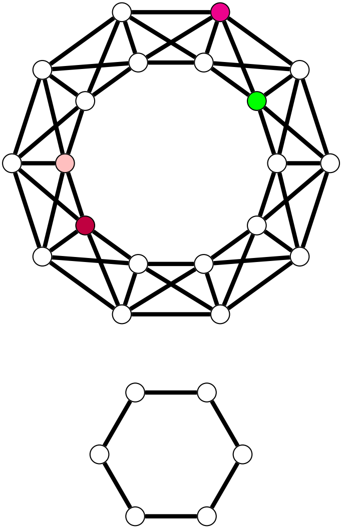
Automorphisms:

How to get automorphisms — Illustration



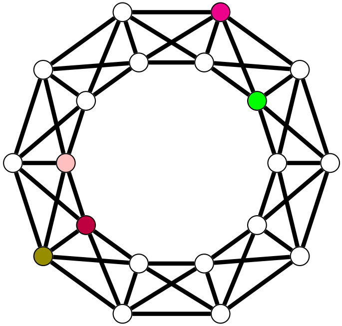


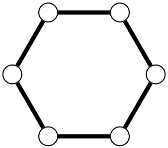
Automorphisms:



Automorphisms:

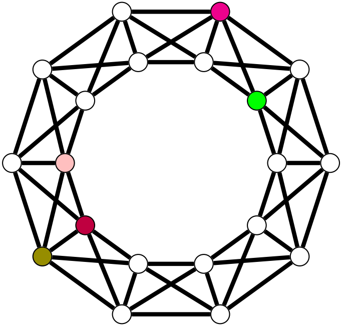
How to get automorphisms — Illustration

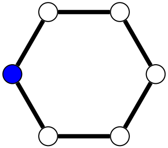




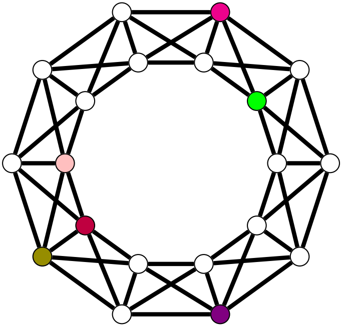
Automorphisms:

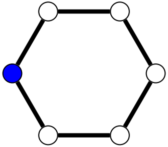
How to get automorphisms — Illustration





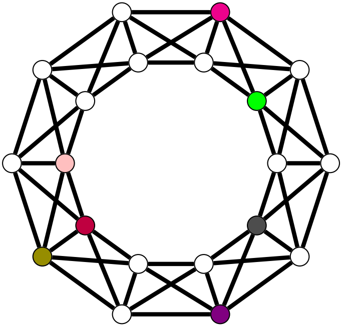
Automorphisms:

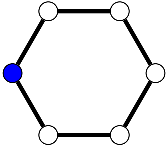




Automorphisms:

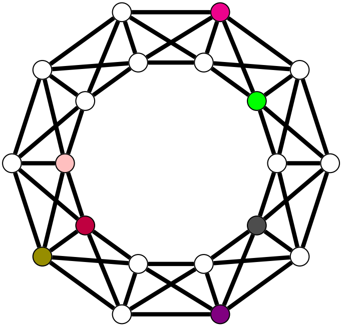
How to get automorphisms — Illustration

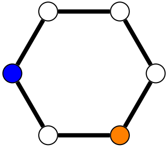




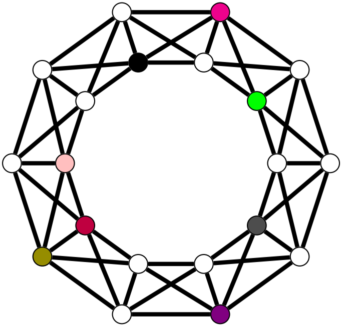
Automorphisms:

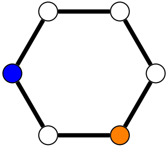
How to get automorphisms — Illustration





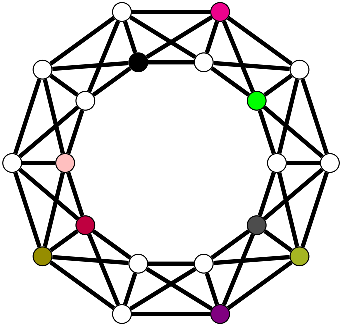
Automorphisms:

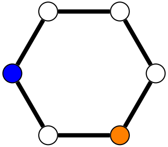




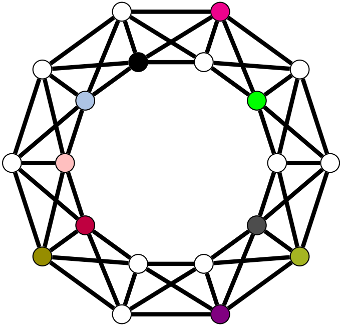
Automorphisms:

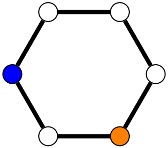
How to get automorphisms — Illustration





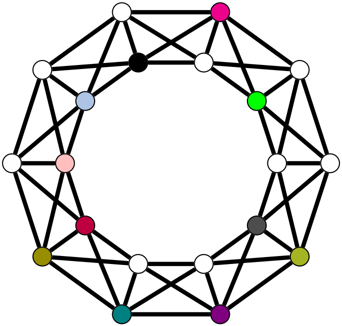
Automorphisms:

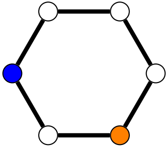




Automorphisms:

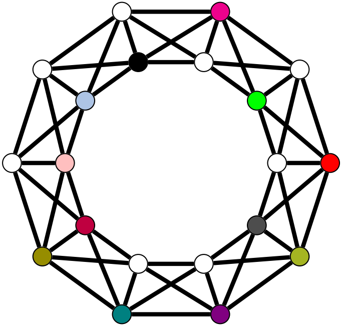
How to get automorphisms — Illustration

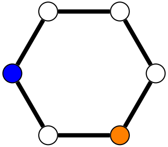




Automorphisms:

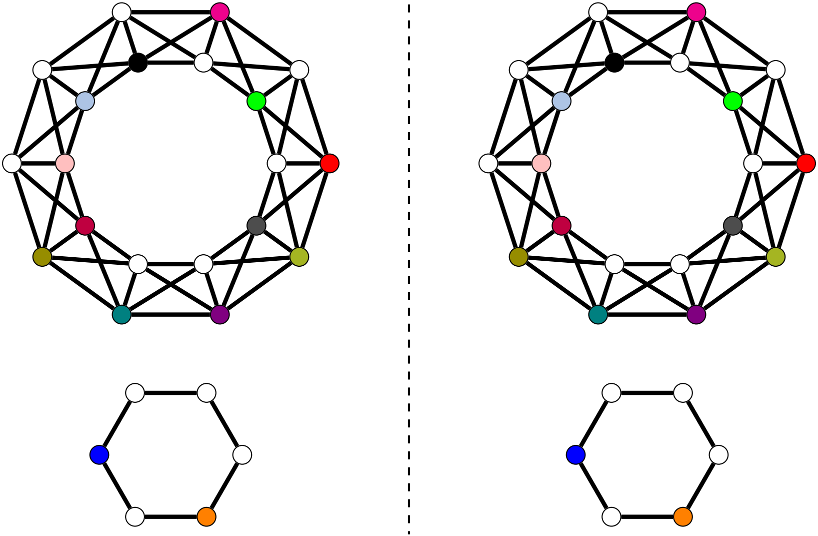
How to get automorphisms — Illustration





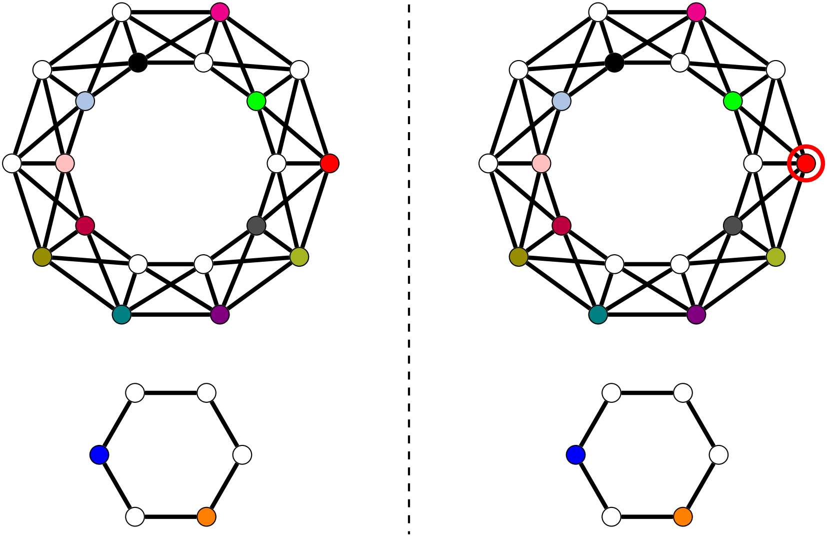
Automorphisms:

How to get automorphisms — Illustration



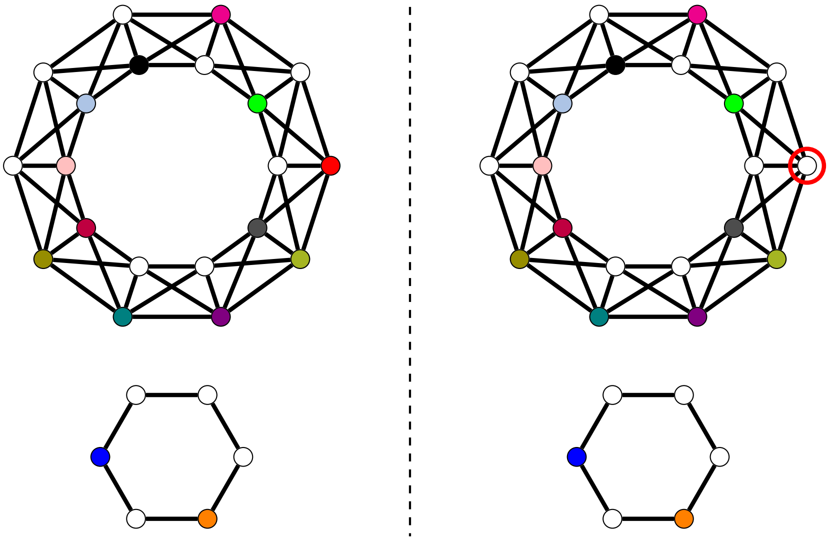
Automorphisms:

How to get automorphisms — Illustration



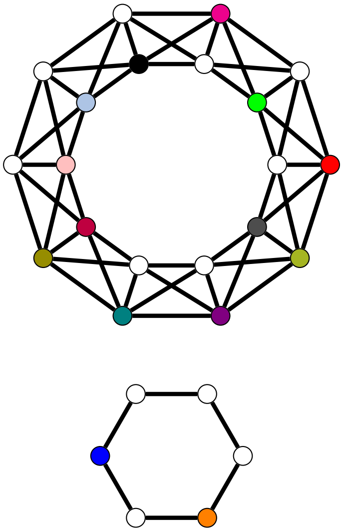
Automorphisms:

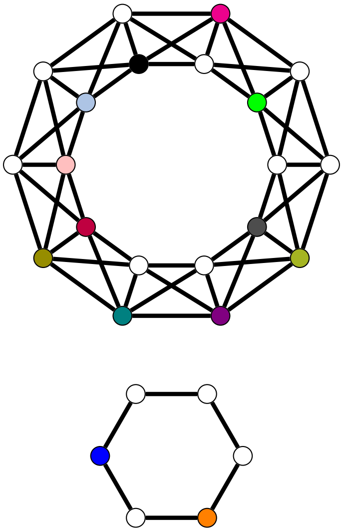
How to get automorphisms — Illustration



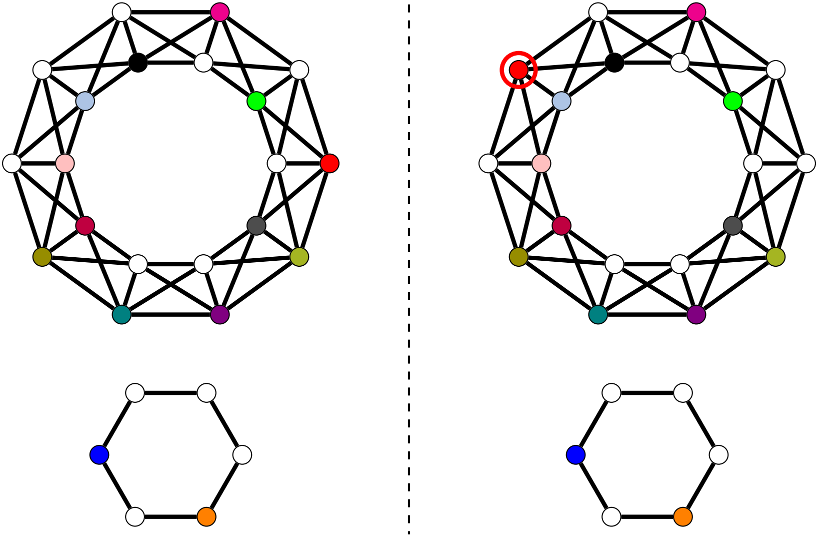
Automorphisms:

How to get automorphisms — Illustration

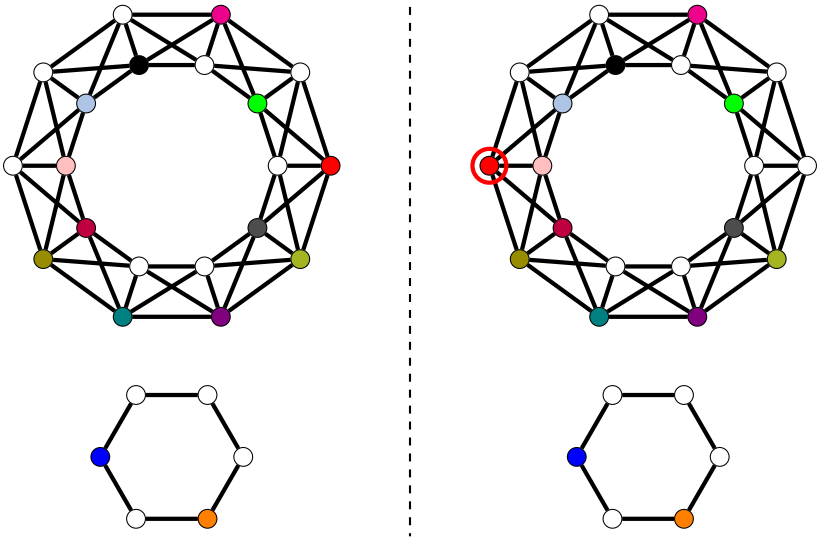




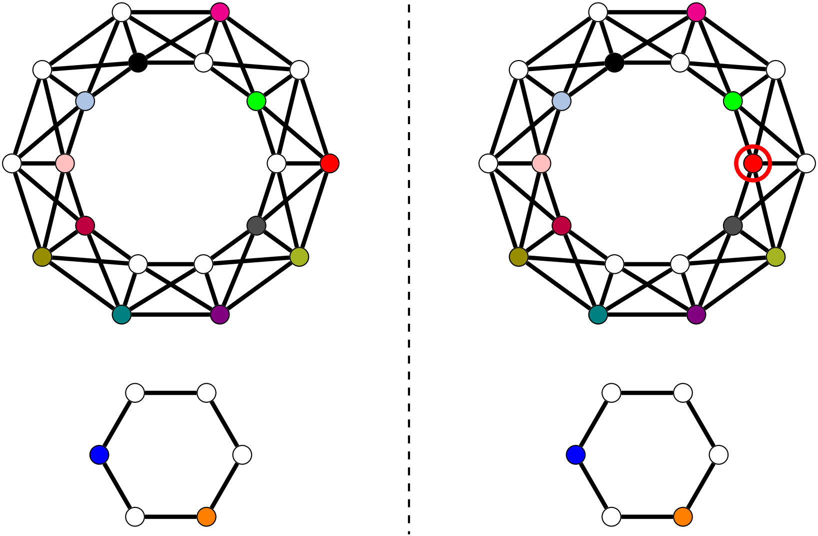
Automorphisms:



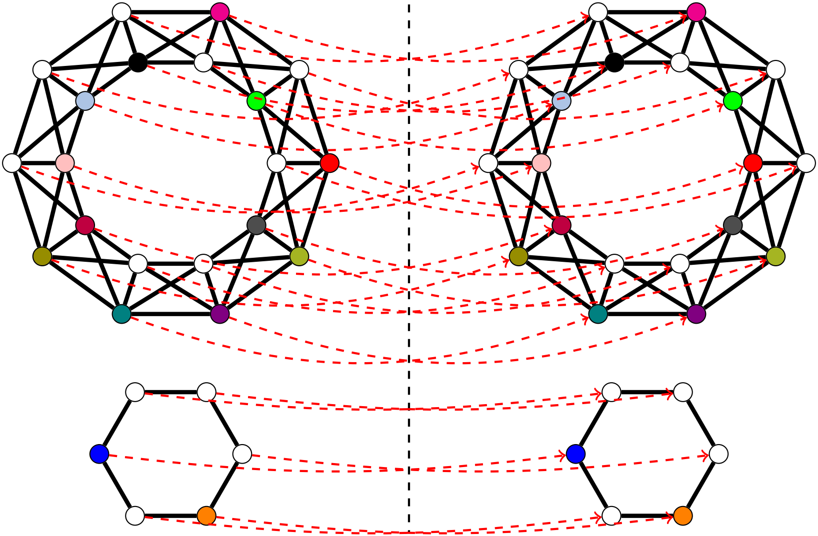
Automorphisms:



Automorphisms:

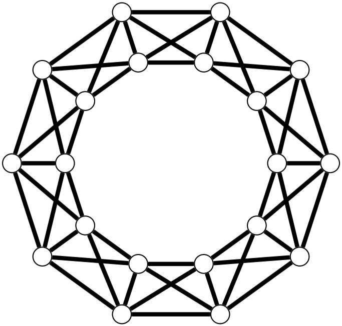


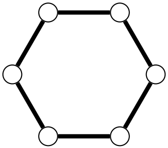
Automorphisms:



Automorphisms:

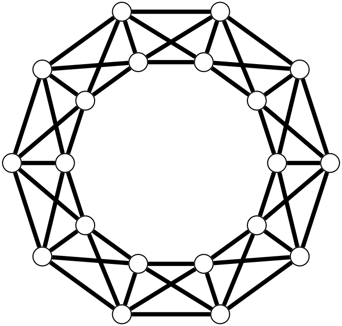
How to get automorphisms — Illustration

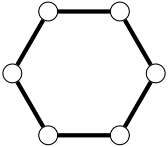




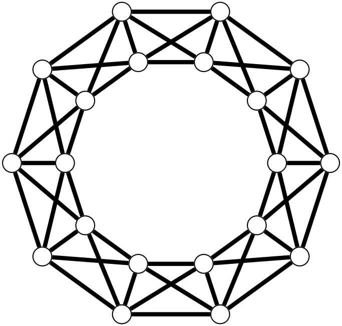
Automorphisms: φ_1

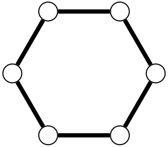
How to get automorphisms — Illustration





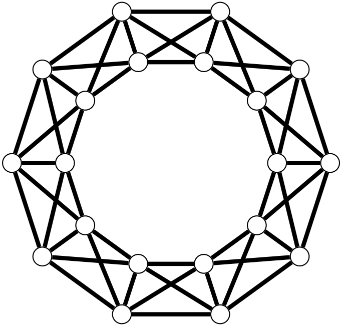
Automorphisms: φ_1

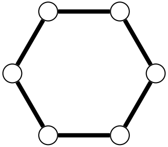




Automorphisms: φ_1, φ_2

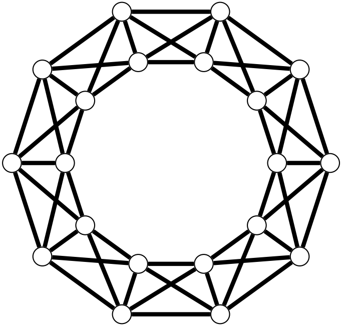
How to get automorphisms — Illustration

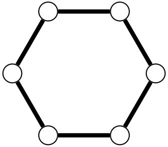




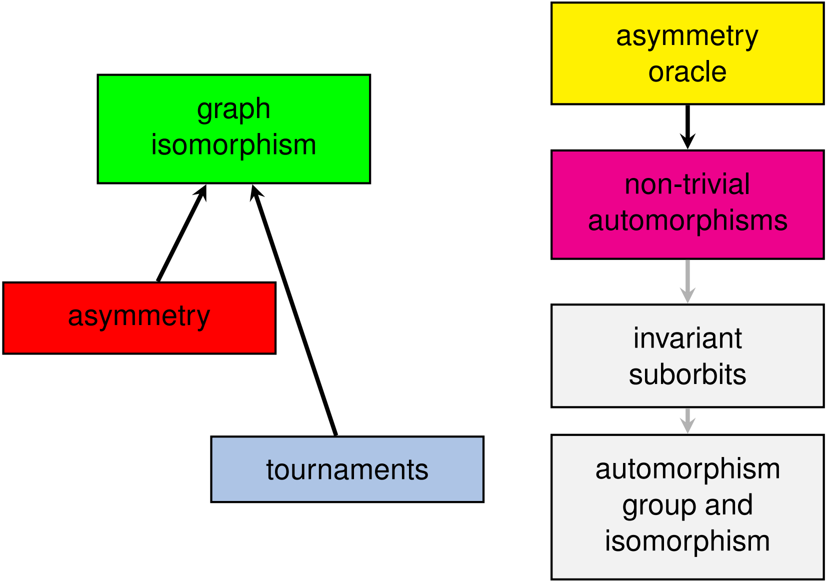
Automorphisms: $\varphi_1, \varphi_2, \varphi_3$

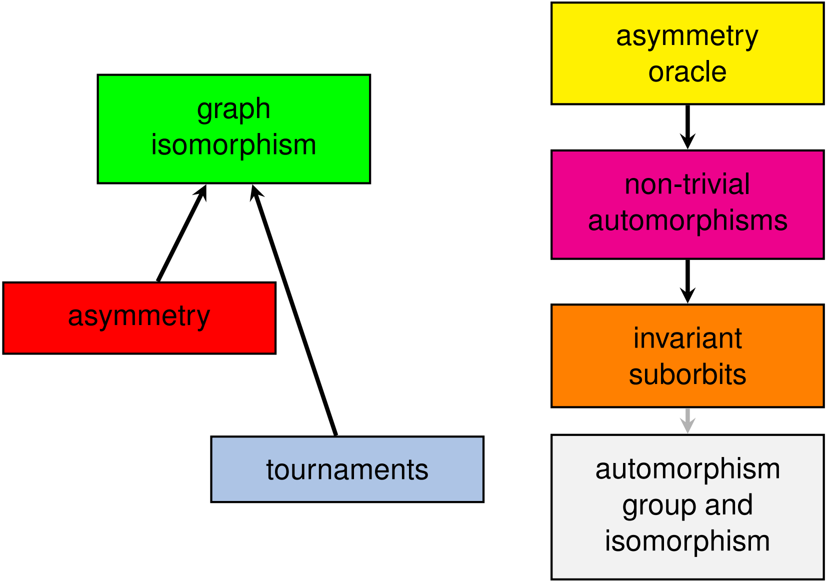
How to get automorphisms — Illustration





Automorphisms: $\varphi_1, \varphi_2, \varphi_3, \ldots$


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Sampling sets

A set of automorphisms $M' \subseteq \mathrm{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \mathrm{Aut}(G)$.

Sampling sets

A set of automorphisms $M' \subseteq \mathrm{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \mathrm{Aut}(G)$. Only invariant sets of automorphisms are useful.

Sampling sets

A set of automorphisms $M' \subseteq \mathrm{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \mathrm{Aut}(G)$. Only invariant sets of automorphisms are useful.

Technique 2: sampling invariant subsets

Sampling sets

A set of automorphisms $M' \subseteq \mathrm{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \mathrm{Aut}(G)$. Only invariant sets of automorphisms are useful.

Technique 2: sampling invariant subsets
There is a technique to extract invariant subsets with high probability. (Sample often and apply Chernoff bounds.)

Sampling sets

A set of automorphisms $M' \subseteq \text{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \text{Aut}(G)$. Only invariant sets of automorphisms are useful.

Technique 2: sampling invariant subsets
There is a technique to extract invariant subsets with high probability. (Sample often and apply Chernoff bounds.)

But: The number of samples required is polynomial in $|\text{Aut}(G)|$, which may be exponential in $|G|$.

Sampling sets

A set of automorphisms $M' \subseteq \text{Aut}(G)$ is invariant if $M'^{\varphi} = M'$ for all $\varphi \in \text{Aut}(G)$. Only invariant sets of automorphisms are useful.

Technique 2: sampling invariant subsets
There is a technique to extract invariant subsets with high probability. (Sample often and apply Chernoff bounds.)

But: The number of samples required is polynomial in $|\text{Aut}(G)|$, which may be exponential in $|G|$.

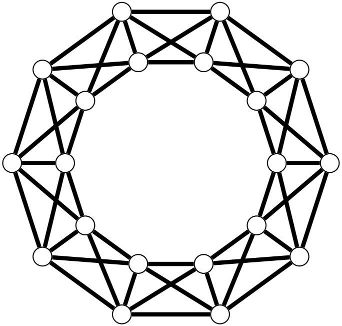
However, we can sample pairs of vertices lying in a common orbit. There are less than n^2 such pairs.

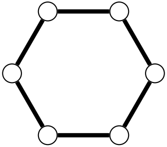
We call a partition $\pi = \{C_1, \ldots, C_t\}$ of the vertices a partition into invariant suborbits if

- every C_i is contained in an orbit
- π is invariant under $\mathrm{Aut}(G)$ (i.e., $\pi^\varphi = \pi$ for all $\varphi \in \mathrm{Aut}(G)$)

Invariant suborbits — Illustration

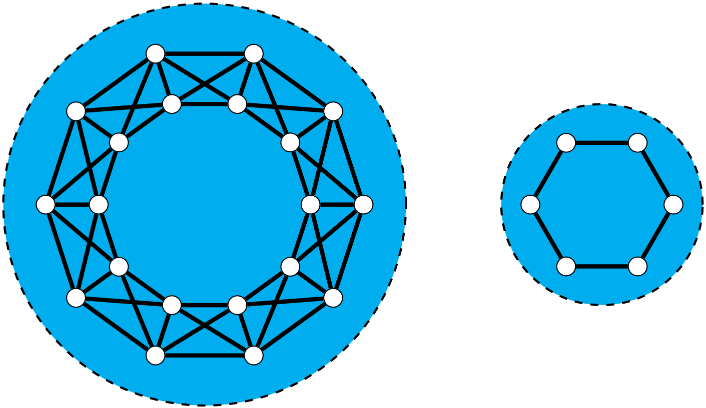
Examples of invariant suborbits





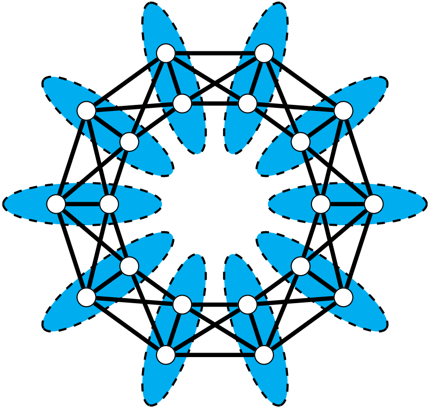
Invariant suborbits — Illustration

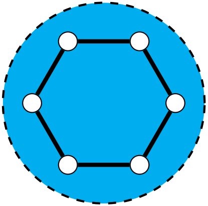
Examples of invariant suborbits



Invariant suborbits — Illustration

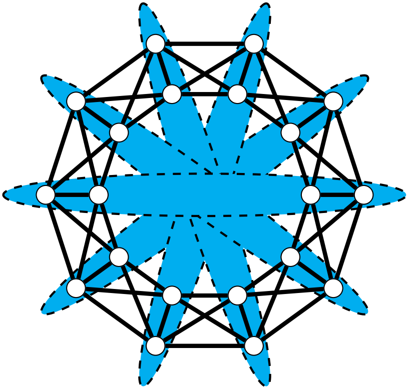
Examples of invariant suborbits

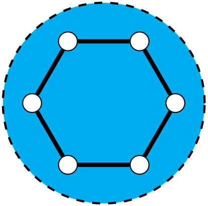




Invariant suborbits — Illustration

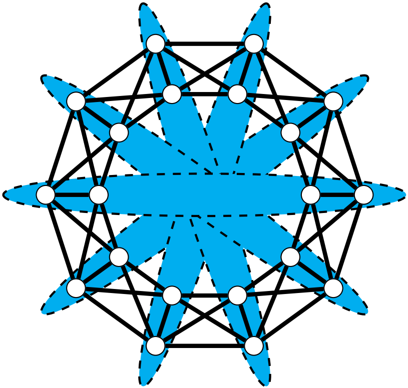
Examples of invariant suborbits

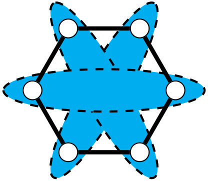




Invariant suborbits — Illustration

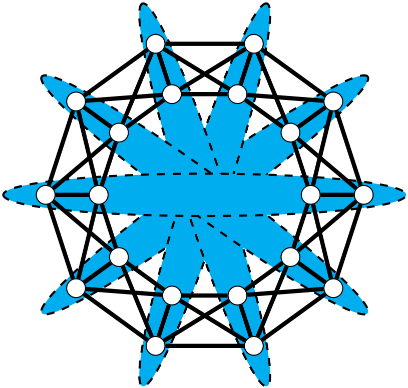
Examples of invariant suborbits

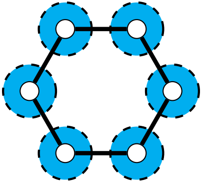




Invariant suborbits — Illustration

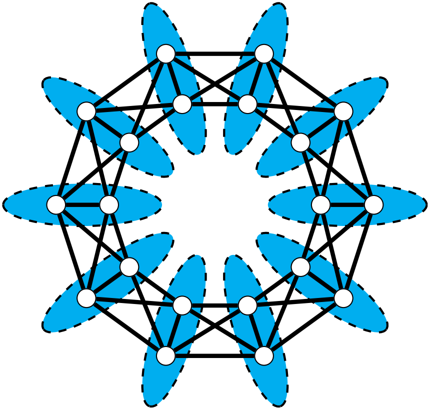
Examples of invariant suborbits

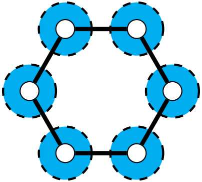




Invariant suborbits — Illustration

Examples of invariant suborbits





Finding invariant suborbits

Lemma

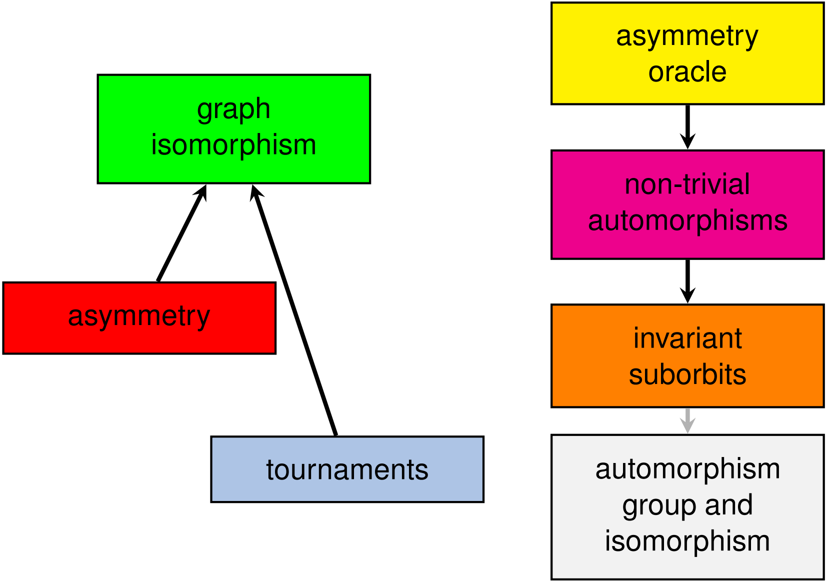
Given an invariant sampler we can compute in polynomial time invariant suborbits (with high probability).

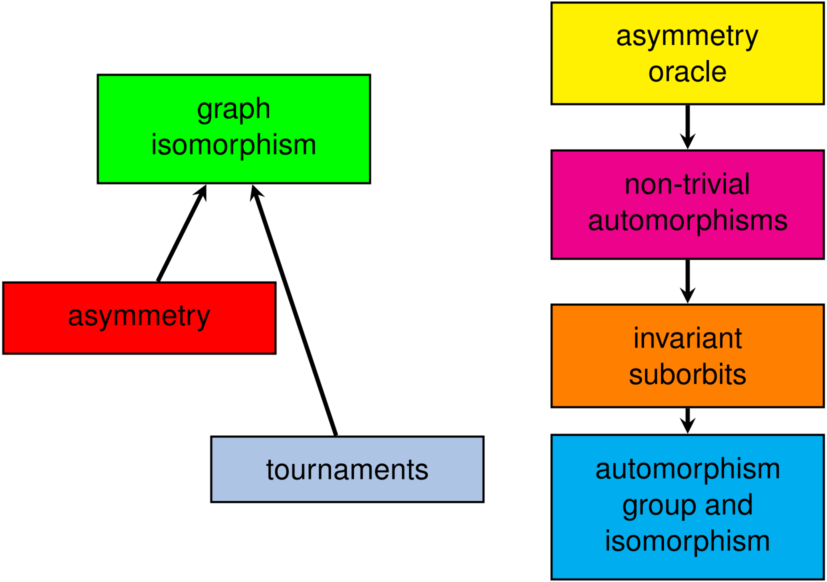
Proof technique:

- repeatedly sample $\varphi \in \mathrm{Aut}(G)$ and randomly pick pair

$$(x,\varphi(x)) \text{ with } x \in V(G)$$

- extract characteristic set of pairs
- compute the transitive closure of the relation induced by pairs


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Computing with solvable groups

Theorem (Luks (1982))

For a solvable permutation group Γ on V and a graph G on vertex set V we can compute $\Gamma \cap \mathrm{Aut}(G)$ in polynomial time.

Facts:

- Tournaments have solvable automorphism group.
- Wreath products of solvable groups are solvable.

The quotient tournament

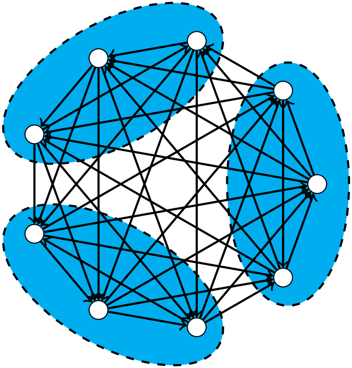
T: a tournament; $\pi = \{C_1, \ldots, C_t\}$: a partition of the vertices

The quotient tournament

T: a tournament; $\pi = \{C_1, \ldots, C_t\}$: a partition of the vertices
The quotient tournament T/π has the vertex set $\{C_1, \ldots, C_t\}$.
The direction of the edge between C_i and C_k is the **majority direction** between C_i and C_k in T.

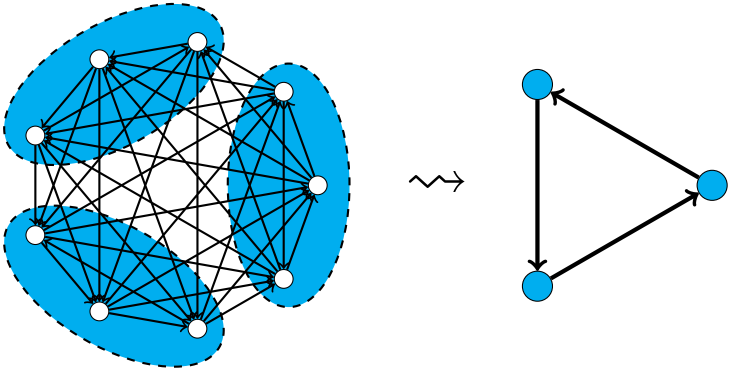
The quotient tournament

T: a tournament; $\pi = \{C_1, \ldots, C_t\}$: a partition of the vertices
The quotient tournament T/π has the vertex set $\{C_1, \ldots, C_t\}$.
The direction of the edge between C_i and C_k is the **majority direction** between C_i and C_k in T.



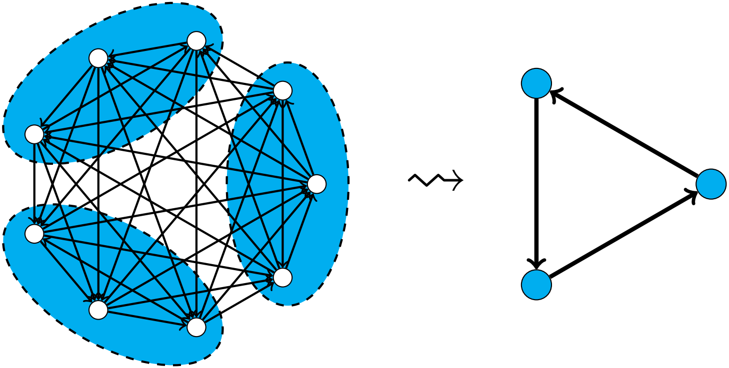
The quotient tournament

T: a tournament; $\pi = \{C_1, \ldots, C_t\}$: a partition of the vertices
The quotient tournament T/π has the vertex set $\{C_1, \ldots, C_t\}$.
The direction of the edge between C_i and C_k is the **majority direction** between C_i and C_k in T.



The quotient tournament

T: a tournament; $\pi = \{C_1, \ldots, C_t\}$: a partition of the vertices
The quotient tournament T/π has the vertex set $\{C_1, \ldots, C_t\}$.
The direction of the edge between C_i and C_k is the **majority direction** between C_i and C_k in T.



Note: if all C_i have odd size this operation is well defined.

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \ldots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \ldots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \ldots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \ldots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$
- compute $\Theta := \mathrm{Aut}(T[C_1])$

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \dots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$
- compute $\Theta := \mathrm{Aut}(T[C_1])$
- compute $\Gamma := \Theta \wr \Delta$ (wreath product)

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \dots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$
- compute $\Theta := \mathrm{Aut}(T[C_1])$
- compute $\Gamma := \Theta \wr \Delta$ (wreath product)
- compute $\Gamma \cap \mathrm{Aut}(T)$

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

- compute a partition $\pi = \{C_1, \dots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$
- compute $\Theta := \mathrm{Aut}(T[C_1])$
- compute $\Gamma := \Theta \wr \Delta$ (wreath product)
- compute $\Gamma \cap \mathrm{Aut}(T)$

Output: $\mathrm{Aut}(T) = \Gamma \cap \mathrm{Aut}(T)$

Basic Strategy

Technique 3: invariant suborbits $\rightsquigarrow$ automorphism group

Input: A tournament T; invariant suborbit oracle

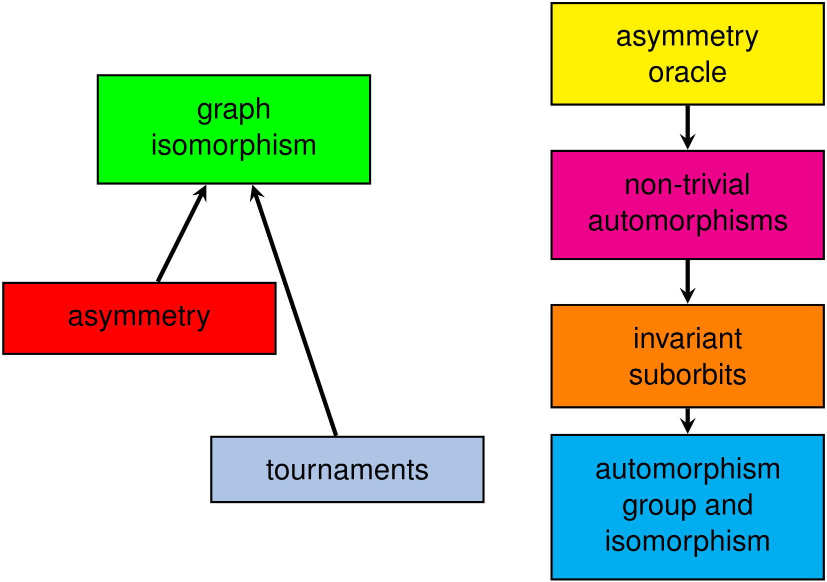
- compute a partition $\pi = \{C_1, \ldots, C_t\}$ into invariant suborbits

For simplicity assume all induced subtournaments $T[C_i]$ are isomorphic.

- compute T/π (quotient tournament)
- compute $\Delta := \mathrm{Aut}(T/\pi)$
- compute $\Theta := \mathrm{Aut}(T[C_1])$
- compute $\Gamma := \Theta \wr \Delta$ (wreath product)
- compute $\Gamma \cap \mathrm{Aut}(T)$

Output: $\mathrm{Aut}(T) = \Gamma \cap \mathrm{Aut}(T)$

A more careful case distinction and some running time analysis show that the overall process runs in polynomial time.


graph isomorphism
asymmetry
tournaments
asymmetry oracle
non-trivial automorphisms
invariant suborbits
automorphism group and isomorphism

Summary

Theorem

There is a polynomial-time randomized reduction from tournament isomorphism to tournament asymmetry.

Summary

Theorem

There is a polynomial-time randomized reduction from tournament isomorphism to tournament asymmetry.

Open: How about for graphs? How about for groups?

Summary

Theorem

There is a polynomial-time randomized reduction from tournament isomorphism to tournament asymmetry.

Open: How about for graphs? How about for groups?

Related results:

Canonization: With [Arvind, Das, Mukhopadhyay] (2010) we get an analogous result for canonization.

Hardness: tournament asymmetry is hard for NL, $C_=L$, PL, DET, and MOD_kL under AC^0 reductions. [Wager] (2007)

Cumulative Prize Money

Cumulative Prize Money

Prize for a proof that $GI \in \mathrm{P}$ or $GI \notin \mathrm{P}$!

100 Euro

Prize for a proof that $GI \in \mathrm{P}$ or $GI \notin \mathrm{P}$!

+ 

100 Euro

Cumulative Prize Money

Prize for a proof that $GI \in \mathrm{P}$ or $GI \notin \mathrm{P}$!

105 Euro