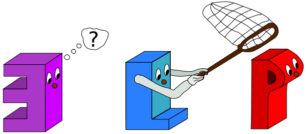
Symmetric Choice and the Quest for a Logic Capturing Polynomial Time

Moritz Lichter

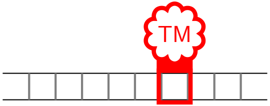
LoGAlg 2023
Nov 16, 2023

European Research Council
Established by the European Commission

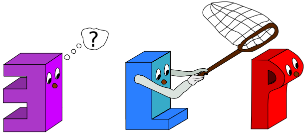
The Quest for a Logic Capturing Polynomial Time



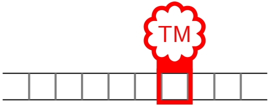
The Quest for a Logic Capturing Polynomial Time



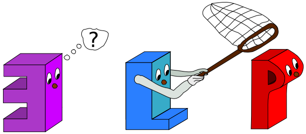
polynomial-time
Turing machine



The Quest for a Logic Capturing Polynomial Time



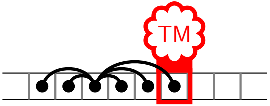
polynomial-time
Turing machine

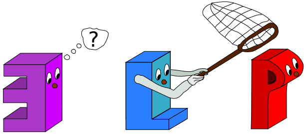


$$\exists x \exists y \exists z.\ E(x,y) \land E(y,z) \land E(x,z)$$

formula of a logic

The Quest for a Logic Capturing Polynomial Time



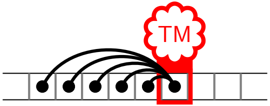


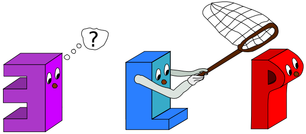
$$\exists x \exists y \exists z.\ E(x,y) \wedge E(y,z) \wedge E(x,z)$$

polynomial-time
Turing machine

formula of a logic

The Quest for a Logic Capturing Polynomial Time



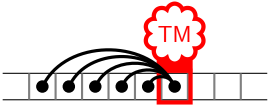


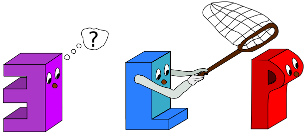
$$\exists x \exists y \exists z.\ E(x,y) \land E(y,z) \land E(x,z)$$

polynomial-time
Turing machine

formula of a logic

The Quest for a Logic Capturing Polynomial Time





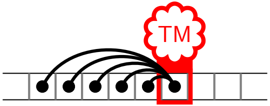
$$\exists x \exists y \exists z.\ E(x,y) \wedge E(y,z) \wedge E(x,z)$$

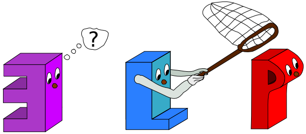
polynomial-time
Turing machine

formula of a logic

fixed-point logic with counting (IFPC)

The Quest for a Logic Capturing Polynomial Time





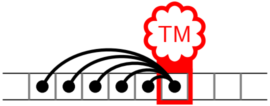
$$\exists x \exists y \exists z.\ E(x,y) \wedge E(y,z) \wedge E(x,z)$$

polynomial-time Turing machine

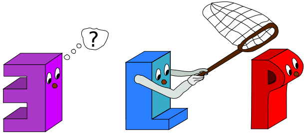
formula of a logic

fixed-point logic with counting (IFPC) + ?

The Quest for a Logic Capturing Polynomial Time



polynomial-time
Turing machine



$\exists x \exists y \exists z.\ E(x, y) \wedge E(y, z) \wedge E(x, z)$

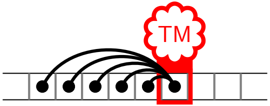
formula of a logic

fixed-point logic with counting (IFPC) + ?

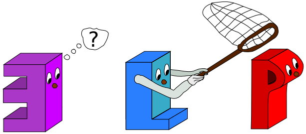
hereditarily finite sets

Choiceless
Polynomial Time

The Quest for a Logic Capturing Polynomial Time



polynomial-time
Turing machine



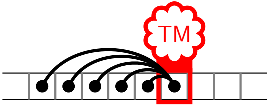
$$\exists x \exists y \exists z.\ E(x,y) \wedge E(y,z) \wedge E(x,z)$$

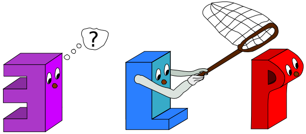
formula of a logic

fixed-point logic with counting (IFPC) + ?

hereditarily finite sets	algebraic operators
Choiceless Polynomial Time	rank logic

The Quest for a Logic Capturing Polynomial Time





$$\exists x \exists y \exists z.\ E(x,y) \wedge E(y,z) \wedge E(x,z)$$

polynomial-time Turing machine

formula of a logic

fixed-point logic with counting (IFPC) + ?

hereditarily finite sets	algebraic operators	choice operators
Choiceless Polynomial Time	rank logic	witnessed symmetric choice

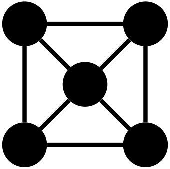
Witnessed Symmetric Choice (Gire and Hoang, '98)

Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits

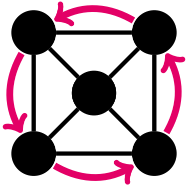
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



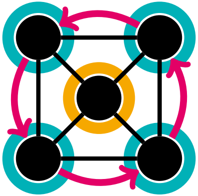
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



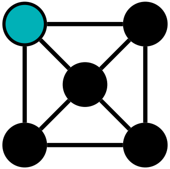
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



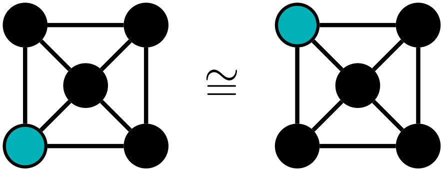
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



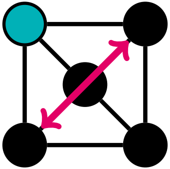
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



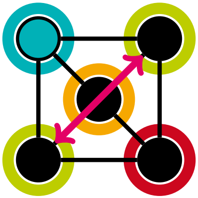
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



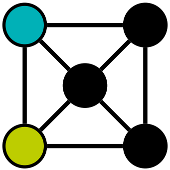
Witnessed Symmetric Choice (Gire and Hoang, ’98)

- **symmetric choice (SC)**: choices from orbits



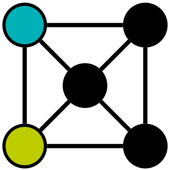
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits



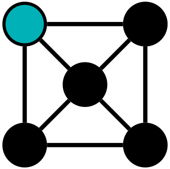
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits
- unknown whether symmetric choice can be evaluated in PTIME



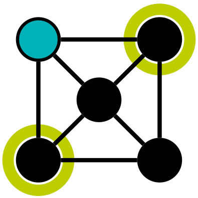
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits
- unknown whether symmetric choice can be evaluated in PTIME
- **witnessed symmetric choice (WSC)**: witness orbits by automorphisms



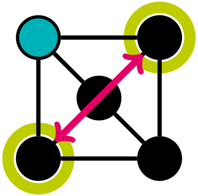
Witnessed Symmetric Choice (Gire and Hoang, ’98)

- **symmetric choice (SC)**: choices from orbits
- unknown whether symmetric choice can be evaluated in PTIME
- **witnessed symmetric choice (WSC)**: witness orbits by automorphisms



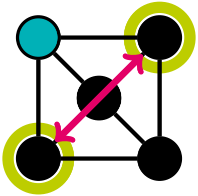
Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits
- unknown whether symmetric choice can be evaluated in PTIME
- **witnessed symmetric choice (WSC)**: witness orbits by automorphisms



Witnessed Symmetric Choice (Gire and Hoang, '98)

- **symmetric choice (SC)**: choices from orbits
- unknown whether symmetric choice can be evaluated in PTIME
- **witnessed symmetric choice (WSC)**: witness orbits by automorphisms



fixed-point operators with (W)SC

$$\textbf{fp-wsc}\Big(\Phi_{\textbf{step}}(x, y), \Phi_{\textbf{choice}}(x), \Phi_{\textbf{wit}}(x, z), \Phi_{\textbf{out}}(x)\Big)$$

Capturing PTIME and Canonization in Choiceless Polynomial Time with Witnessed Symmetric Choice

The Role of Canonization

Capturing PTIME via definable canonization

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

$\downarrow$ canonization

ordered graph class G

canonization:

for all $A, B \in \mathbf{G}$

- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B)$ $\Leftrightarrow$ $A \cong B$

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

↓ canonization

ordered graph class G

↓ Immerman-Vardi

capturing PTIME on G

canonization:
for all $A, B \in \mathbf{G}$
- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B)$
  $\Leftrightarrow \quad A \cong B$

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

$\downarrow$ canonization

ordered graph class G

$\downarrow$ Immerman-Vardi

capturing PTIME on G

canonization:
for all $A, B \in \mathbf{G}$
- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B)$
  $\Leftrightarrow \quad A \cong B$

Defining canonization vs. defining isomorphism

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

↓ canonization

ordered graph class G

↓ Immerman-Vardi

capturing PTIME on G

canonization:
for all $A, B \in \mathbf{G}$
- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B)$
  $\Leftrightarrow \quad A \cong B$

Defining canonization vs. defining isomorphism

- Defining canonization is more difficult.

The Role of Canonization

Capturing PTIME via definable canonization

graph class $\mathbf{G}$

$\downarrow$ canonization

ordered graph class $\mathbf{G}$

$\downarrow$ Immerman-Vardi

capturing PTIME on $\mathbf{G}$

canonization:
for all $A, B \in \mathbf{G}$
- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B)$
  $\Leftrightarrow \quad A \cong B$

Defining canonization vs. defining isomorphism

- Defining canonization is more difficult.
- Is canonization necessarily definable?

The Role of Canonization

Capturing PTIME via definable canonization

graph class G

↓ canonization

ordered graph class G

↓ Immerman-Vardi

capturing PTIME on G

canonization:
for all $A, B \in \mathbf{G}$
- $\text{can}(A) \cong A$
- $\text{can}(A) = \text{can}(B) \Leftrightarrow A \cong B$

Defining canonization vs. defining isomorphism

- Defining canonization is more difficult.
- Is canonization necessarily definable?
- Does isomorphism testing imply canonization?

CPT and Isomorphism Testing

CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**

CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$

- $\mathrm{inv}(A) = \mathrm{inv}(B)$
  $\Leftrightarrow A \cong B$

CPT and Isomorphism Testing

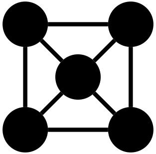
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\mathrm{inv}(A) = \mathrm{inv}(B)$
  $\Leftrightarrow\ A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

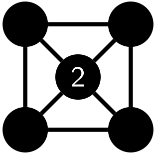
CPT-definable isomorphism test for **G** implies

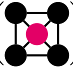
- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$

- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 2

CPT and Isomorphism Testing

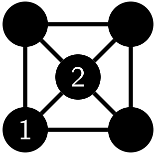
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

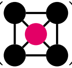
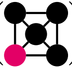
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B) \Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 2, inv () = 1

CPT and Isomorphism Testing

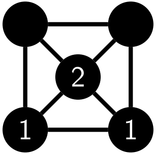
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

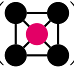
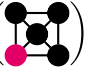
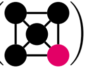
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow \; A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 2, inv () = 1, inv () = 1

CPT and Isomorphism Testing

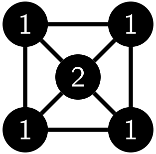
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

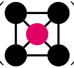
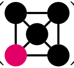
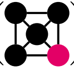
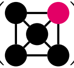
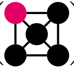
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow\ A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 2, inv () = 1, inv () = 1, inv () = 1, inv () = 1

CPT and Isomorphism Testing

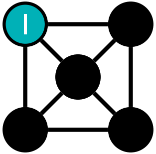
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\mathrm{inv}(A) = \mathrm{inv}(B) \Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

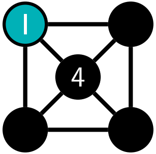
CPT-definable isomorphism test for **G** implies

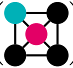
- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$

- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 4

CPT and Isomorphism Testing

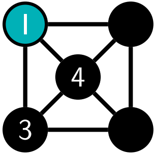
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

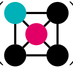
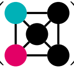
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow\ A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 4, inv () = 3

CPT and Isomorphism Testing

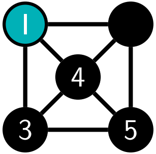
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

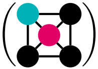
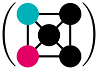
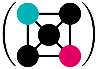
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B) \Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv  = 4, inv  = 3, inv  = 5

CPT and Isomorphism Testing

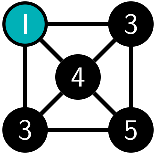
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

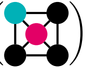
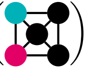
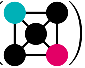
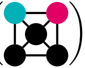
CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\mathrm{inv}(A) = \mathrm{inv}(B) \Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



inv () = 4, inv () = 3, inv () = 5, inv () = 3

CPT and Isomorphism Testing

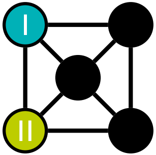
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

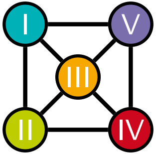
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

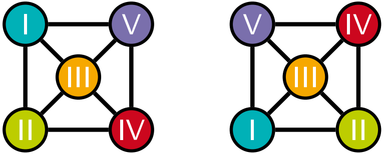
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\text{inv}(A) = \text{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

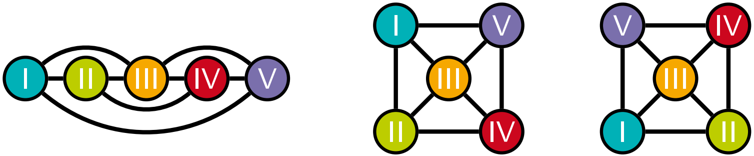
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\mathrm{inv}(A) = \mathrm{inv}(B) \Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



CPT and Isomorphism Testing

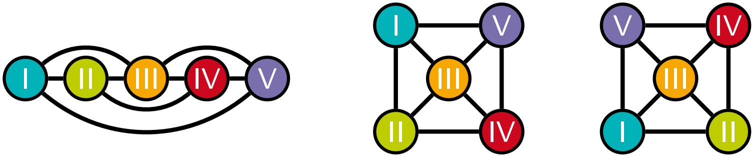
Deep Weisfeiler Leman (Grohe, Schweitzer, Wiebking, '19)

CPT-definable isomorphism test for **G** implies

- a CPT-definable complete invariant for **G**
- a canonization algorithm (if **G** is individualization-closed)

complete invariant:
for all $A, B \in \mathbf{G}$
- $\mathrm{inv}(A) = \mathrm{inv}(B)$
  $\Leftrightarrow A \cong B$

Gurevich's canonization algorithm (Gurevich, '01)



Not definable in CPT but definable in CPT+WSC!

Defining Isomorphisms and Canonization in CPT+WSC

Theorem. (L., Schweitzer, '21)

For every individualization-closed graph class **G**, the following are equivalent:

Defining Isomorphisms and Canonization in CPT+WSC

Theorem. (L., Schweitzer, '21)
For every individualization-closed graph class **G**, the following are equivalent:

1. CPT+WSC defines a **complete invariant** for **G**

Defining Isomorphisms and Canonization in CPT+WSC

Theorem. (L., Schweitzer, '21)
For every individualization-closed graph class **G**, the following are equivalent:

1. CPT+WSC defines a **complete invariant** for **G**
2. CPT+WSC defines a **canonization** for **G**

Defining Isomorphisms and Canonization in CPT+WSC

Theorem. (L., Schweitzer, '21)
For every individualization-closed graph class **G**, the following are equivalent:

1. CPT+WSC defines a **complete invariant** for **G**
2. CPT+WSC defines a **canonization** for **G**
3. CPT+WSC defines **isomorphism** of **G**

Defining Isomorphisms and Canonization in CPT+WSC

Theorem. (L., Schweitzer, '21)
For every individualization-closed graph class **G**, the following are equivalent:

1. CPT+WSC defines a **complete invariant** for **G**
2. CPT+WSC defines a **canonization** for **G**
3. CPT+WSC defines **isomorphism** of **G**

Corollary.
For every individualization-closed graph class **G** with a PTIME isomorphism test
CPT+WSC defines **isomorphism** of **G** $\Longleftrightarrow$ CPT+WSC **captures PTIME** on **G**.

Expressiveness of Symmetric Choice in Fixed-Point Logic with Counting

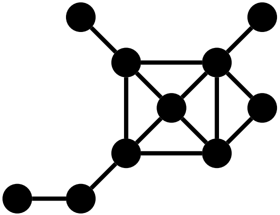
Symmetric Choice, Asymmetric Structures, and Interpretations

Symmetric Choice, Asymmetric Structures, and Interpretations

symmetric choice on **asymmetric structures** is useless

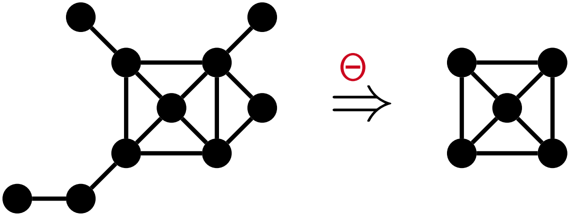
Symmetric Choice, Asymmetric Structures, and Interpretations

symmetric choice on **asymmetric structures** is useless



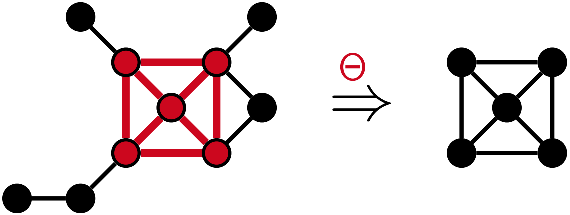
Symmetric Choice, Asymmetric Structures, and Interpretations

symmetric choice on **asymmetric structures** is useless



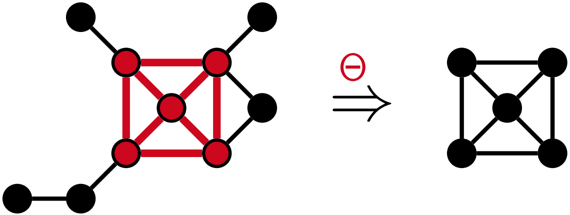
Symmetric Choice, Asymmetric Structures, and Interpretations

symmetric choice on **asymmetric structures** is useless



Symmetric Choice, Asymmetric Structures, and Interpretations

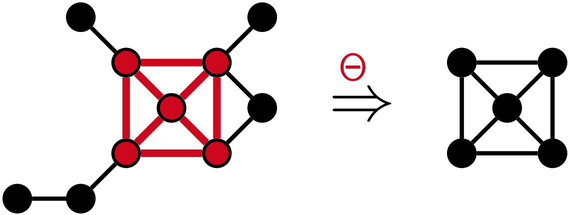
symmetric choice on **asymmetric structures** is useless



Interpretation operator $\mathrm{I}(\Theta, \Phi)$ (Gire and Hoang, '98)

Symmetric Choice, Asymmetric Structures, and Interpretations

symmetric choice on **asymmetric structures** is useless

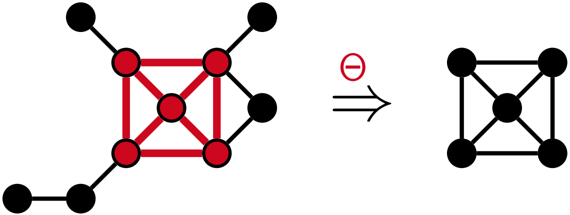


Interpretation operator $I(\Theta, \Phi)$ (Gire and Hoang, '98)

- Is IFPC+WSC+I more expressive than IFPC+WSC?

Symmetric Choice, Asymmetric Structures, and Interpretations

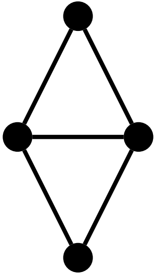
symmetric choice on **asymmetric structures** is useless



Interpretation operator $I(\Theta, \Phi)$ (Gire and Hoang, '98)

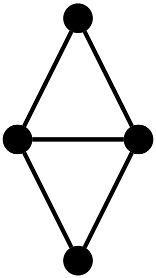
- Is IFPC+WSC+I more expressive than IFPC+WSC?
- Is IFP+SC+I more expressive than IFP+SC? (Dawar, Richerby, '03)

The CFI Query

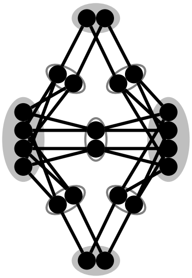


base graph

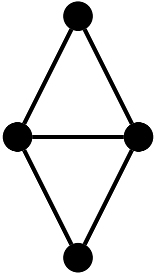
The CFI Query



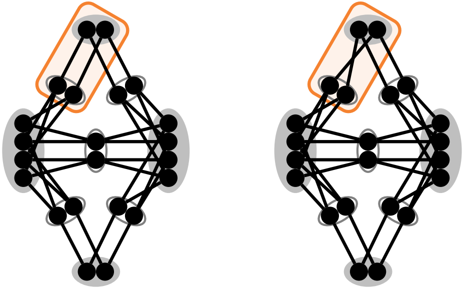
base graph



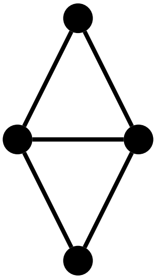
The CFI Query



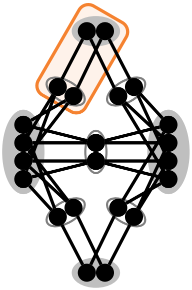
base graph



The CFI Query

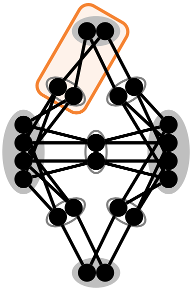


base graph



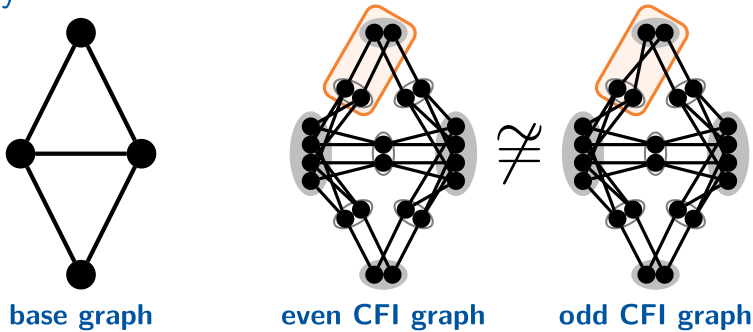
even CFI graph

$\ncong$



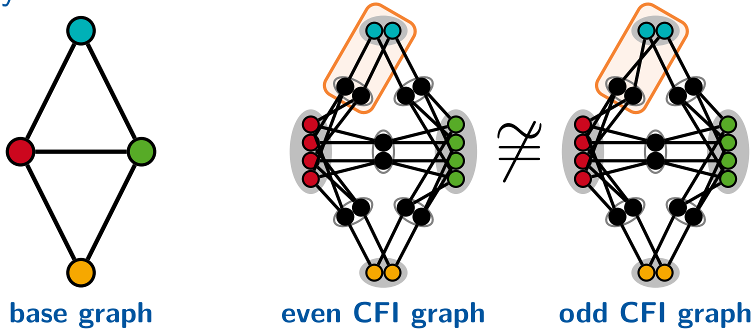
odd CFI graph

The CFI Query



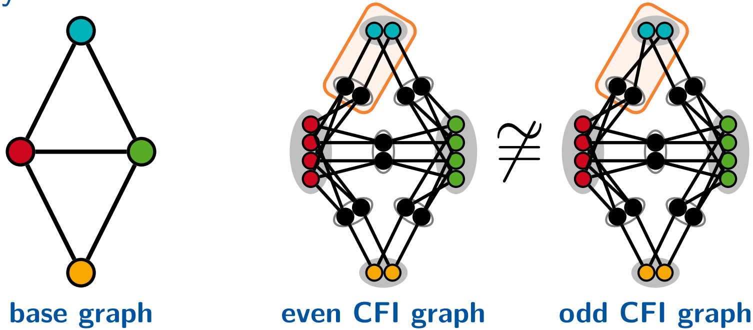
CFI query: define whether a CFI graph is even

The CFI Query



CFI query: define whether a CFI graph is even
ordered CFI query: ordered base graphs

The CFI Query



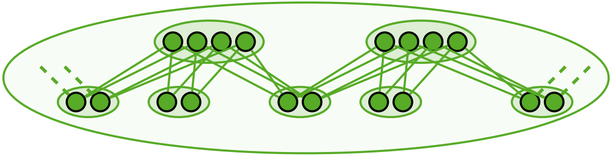
CFI query: define whether a CFI graph is even

ordered CFI query: ordered base graphs

Theorem (Cai, Fürer, Immerman, '92). The (ordered) CFI query is not IFPC-definable.

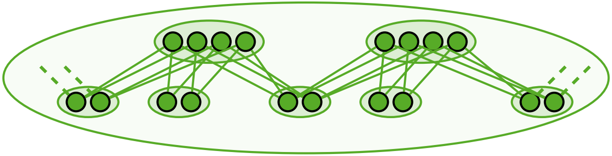
Theorem (Gire, Hoang, '98). The ordered CFI query is IFP+WSC-definable.

Separating IFPC+WSC from IFPC+WSC+I

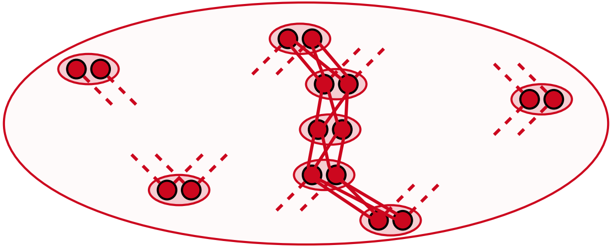


ordered CFI graph

Separating IFPC+WSC from IFPC+WSC+I



ordered CFI graph



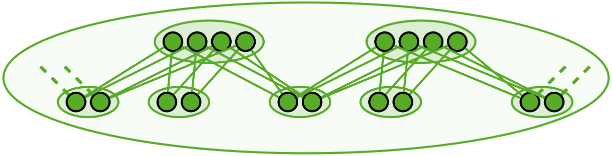
multipede

(Gurevich, Shelah, '96)

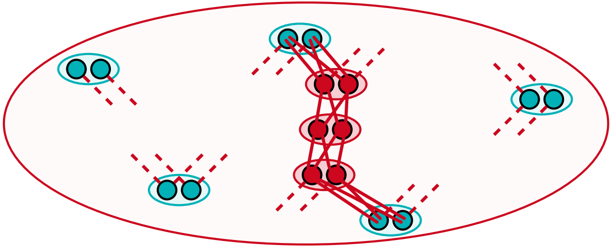
asymmetric structures

orbits not IFPC-definable

Separating IFPC+WSC from IFPC+WSC+I



ordered CFI graph



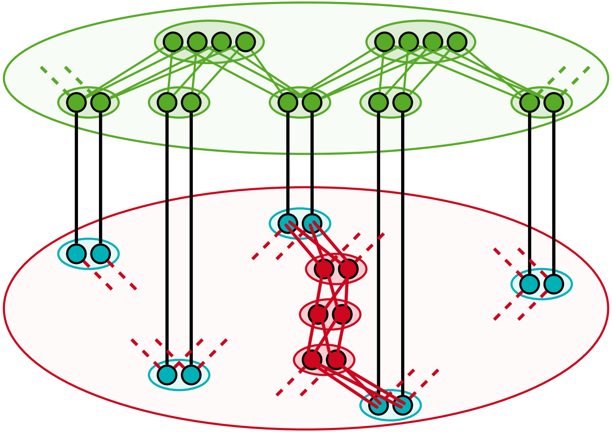
multipede

(Gurevich, Shelah, '96)

asymmetric structures

orbits not IFPC-definable

Separating IFPC+WSC from IFPC+WSC+I



ordered CFI graph

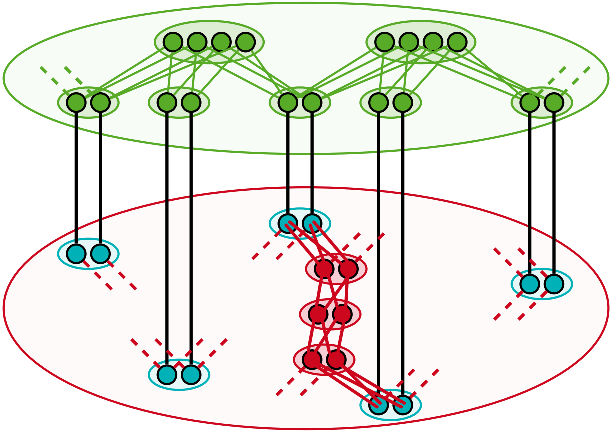
multipede

(Gurevich, Shelah, '96)

asymmetric structures

orbits not IFPC-definable

Separating IFPC+WSC from IFPC+WSC+I



ordered CFI+multipede query

ordered CFI graph

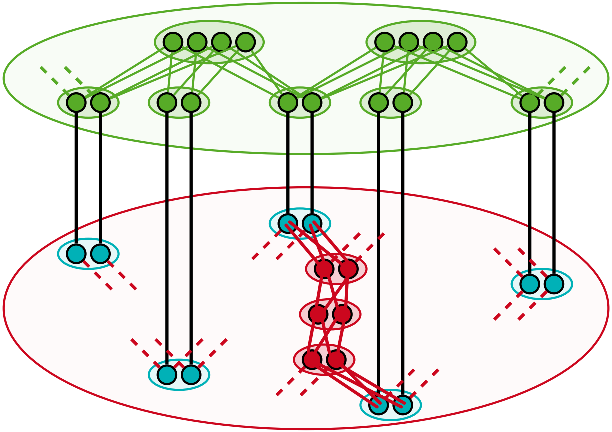
multipede

(Gurevich, Shelah, '96)

asymmetric structures

orbits not IFPC-definable

Separating IFPC+WSC from IFPC+WSC+I



ordered CFI+multipede query

ordered CFI graph

multipede

(Gurevich, Shelah, '96)

asymmetric structures

orbits not IFPC-definable

Theorem (L, '23). IFPC+WSC $<$ IFPC+WSC+I $\leq$ PTIME

IFP(C)+(W)SC does not define the ordered CFI+multipede query.
IFP(C)+(W)SC+I defines the ordered CFI+multipede query.

Towards Separating IFPC+WSC+I from PTIME

Towards Separating IFPC+WSC+I from PTIME

Goal: Prove an operator nesting hierarchy for IFPC+WSC+I on CFI graphs

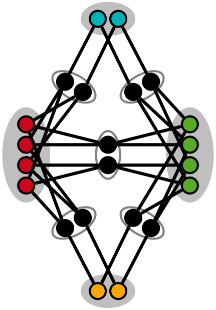
$$\mathsf{IFPC} \leq \mathrm{WSCI}(\mathsf{IFPC}) \leq \mathrm{WSCI}(\mathrm{WSCI}(\mathsf{IFPC})) \leq \cdots \leq \textsc{Ptime}$$

Towards Separating IFPC+WSC+I from PTIME

Goal: Prove an operator nesting hierarchy for IFPC+WSC+I on CFI graphs

$$\text{IFPC} < \text{WSCI(IFPC)} \leq \text{WSCI(WSCI(IFPC))} \leq \cdots \leq \text{PTIME}$$

(Gire, Hoang, '98)

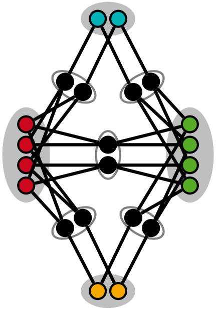


Towards Separating IFPC+WSC+I from PTIME

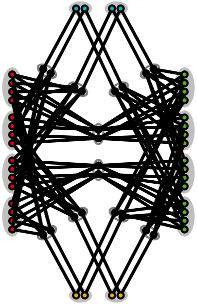
Goal: Prove an operator nesting hierarchy for IFPC+WSC+I on CFI graphs

$$\text{IFPC} < \text{WSCI(IFPC)} < \text{WSCI(WSCI(IFPC))} \leq \cdots \leq \text{PTIME}$$

(Gire, Hoang, '98)



(L., '23)

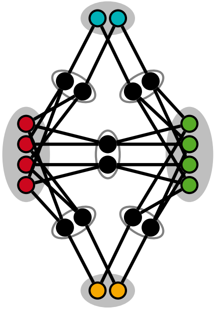


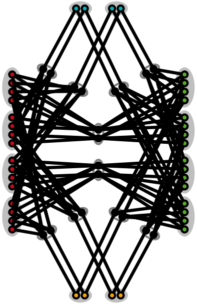
Towards Separating IFPC+WSC+I from PTIME

Goal: Prove an operator nesting hierarchy for IFPC+WSC+I on CFI graphs

$$\text{IFPC} < \text{WSCI(IFPC)} < \text{WSCI(WSCI(IFPC))} \leq \cdots \leq \text{PTIME}$$

(Gire, Hoang, '98) (L., '23) (ongoing work ...)





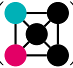
Summary: Witnessed Symmetric Choice

$$\textbf{fp-wsc}\Big(\Phi_{\textbf{step}}(x, y), \Phi_{\textbf{choice}}(x), \Phi_{\textbf{wit}}(x, z), \Phi_{\textbf{out}}(x)\Big)$$

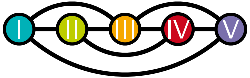
Summary: Witnessed Symmetric Choice

$$\textbf{fp-wsc}\Big(\Phi_{\textbf{step}}(x, y), \Phi_{\textbf{choice}}(x), \Phi_{\textbf{wit}}(x, z), \Phi_{\textbf{out}}(x)\Big)$$

isomorphism
⇔ canonization
in CPT+WSC

inv () = 3

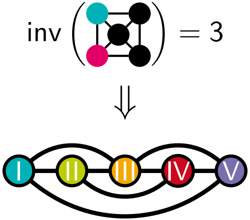
⇓



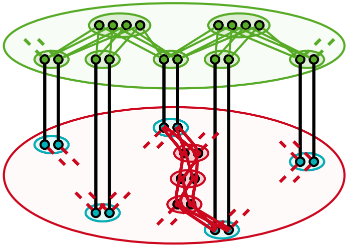
Summary: Witnessed Symmetric Choice

$$\textbf{fp-wsc}\Big(\Phi_{\textbf{step}}(x, y), \Phi_{\textbf{choice}}(x), \Phi_{\textbf{wit}}(x, z), \Phi_{\textbf{out}}(x)\Big)$$

isomorphism
⇔ canonization
in CPT+WSC



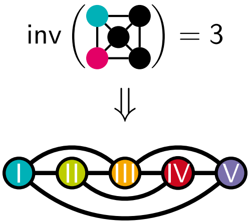
IFPC+WSC
does not capture PTIME



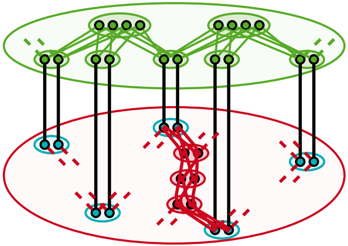
Summary: Witnessed Symmetric Choice

$$\textbf{fp-wsc}\Big(\Phi_{\textbf{step}}(x, y), \Phi_{\textbf{choice}}(x), \Phi_{\textbf{wit}}(x, z), \Phi_{\textbf{out}}(x)\Big)$$

isomorphism
$\Leftrightarrow$ canonization
in CPT+WSC



IFPC+WSC
does not capture PTIME



IFPC+WSC and
interpretations