A tutorial on order- and arb-invariant logics

Nicole Schweikardt

Goethe-Universität Frankfurt am Main

Journées d'Informatique Fondamentale de Paris Diderot
26 April 2013, Paris

In this talk

- consider finite relational structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ over a finite relational signature $\tau = \{R_1, \ldots, R_\ell\}$

- p is a τ-property, if the following is true for all finite τ-structures $\mathcal{A}$ and $\mathcal{B}$:

  if $\mathcal{A} \cong \mathcal{B}$, then $\mathcal{A}$ has property $p \iff \mathcal{B}$ has property p

- q is a k-ary τ-query, if the following is true:

  if $\pi : \mathcal{A} \cong \mathcal{B}$, then for all $a_1, \ldots, a_k \in A$,

  $$(a_1, \ldots, a_k) \in q(\mathcal{A}) \iff (\pi(a_1), \ldots, \pi(a_k)) \in q(\mathcal{B})$$

- I.e., τ-properties and queries are closed under isomorphisms.

In this talk

- consider finite relational structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ over a finite relational signature $\tau = \{R_1, \ldots, R_\ell\}$
- p is a τ-property, if the following is true for all finite τ-structures $\mathcal{A}$ and $\mathcal{B}$:

  if $\mathcal{A} \cong \mathcal{B}$, then $\mathcal{A}$ has property $p \iff \mathcal{B}$ has property p
- q is a k-ary τ-query, if the following is true:

  if $\pi : \mathcal{A} \cong \mathcal{B}$, then for all $a_1, \ldots, a_k \in A$,

  $$(a_1, \ldots, a_k) \in q(\mathcal{A}) \iff (\pi(a_1), \ldots, \pi(a_k)) \in q(\mathcal{B})$$
- I.e., τ-properties and queries are closed under isomorphisms.

In this talk

- consider finite relational structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ over a finite relational signature $\tau = \{R_1, \ldots, R_\ell\}$

- p is a τ-property, if the following is true for all finite τ-structures $\mathcal{A}$ and $\mathcal{B}$:

  if $\mathcal{A} \cong \mathcal{B}$, then $\mathcal{A}$ has property $p \iff \mathcal{B}$ has property p

- q is a k-ary τ-query, if the following is true:

  if $\pi : \mathcal{A} \cong \mathcal{B}$, then for all $a_1, \ldots, a_k \in A$,

  $$\big(a_1, \ldots, a_k\big) \in q(\mathcal{A}) \iff \big(\pi(a_1), \ldots, \pi(a_k)\big) \in q(\mathcal{B})$$

- I.e., τ-properties and queries are closed under isomorphisms.

In this talk

- consider finite relational structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ over a finite relational signature $\tau = \{R_1, \ldots, R_\ell\}$

- p is a τ-property, if the following is true for all finite τ-structures $\mathcal{A}$ and $\mathcal{B}$:

  if $\mathcal{A} \cong \mathcal{B}$, then $\quad \mathcal{A}$ has property $p \iff \mathcal{B}$ has property p

- q is a k-ary τ-query, if the following is true:

  if $\pi : \mathcal{A} \cong \mathcal{B}$, then for all $a_1, \ldots, a_k \in A$,

  $$(a_1, \ldots, a_k) \in q(\mathcal{A}) \iff (\pi(a_1), \ldots, \pi(a_k)) \in q(\mathcal{B})$$

- I.e., τ-properties and queries are closed under isomorphisms.

Logics expressing τ-properties and queries

Classical logics like, e.g.

- FO (first-order logic: Boolean combinations + quantification over nodes)
- LFP (least fixed point logic: FO + inductive definitions of relations)

express τ-properties and queries in a straightforward way.

Example: Consider graphs $\mathcal{A} = (A, E^{\mathcal{A}})$. The query

$$q(\mathcal{A}) \;=\; \{\, x \in A \;:\; x \text{ lies on a triangle} \,\}$$

is expressed in FO via

$$\varphi(x) \;:=\; \exists y\, \exists z\, \big(\, E(x,y) \,\wedge\, E(y,z) \,\wedge\, E(z,x) \,\big)$$

Drawback:

FO and LFP are too weak to express (some) computationally easy properties, e.g., properties concerning the size of A or $E^{\mathcal{A}}$.
Stronger logics like, e.g., SO or ESO can express computationally hard properties and queries.

Logics expressing τ-properties and queries

Classical logics like, e.g.

- FO (first-order logic: Boolean combinations + quantification over nodes)
- LFP (least fixed point logic: FO + inductive definitions of relations)

express τ-properties and queries in a straightforward way.

Example: Consider graphs $\mathcal{A} = (A, E^{\mathcal{A}})$. The query

$$q(\mathcal{A}) \;=\; \{\, x \in A \;:\; x \text{ lies on a triangle} \,\}$$

is expressed in FO via

$$\varphi(x) \;:=\; \exists y\, \exists z\, \big(\, E(x,y) \,\wedge\, E(y,z) \,\wedge\, E(z,x) \,\big)$$

Drawback:

FO and LFP are too weak to express (some) computationally easy properties, e.g., properties concerning the size of A or $E^{\mathcal{A}}$.
Stronger logics like, e.g., SO or ESO can express computationally hard properties and queries.

Overview

Overview

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, \mathit{Halt}, \ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots, \mathit{Halt}, \ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$. These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).
A -formula $\varphi(\vec{x})$ is -invariant on $\mathcal{A} = (A, R_1^A, \ldots, R_\ell^A)$ $\iff$
for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A,

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots,$ *Halt*, $\ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots,$ *Halt*, $\ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$. These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A -formula $\varphi(\vec{x})$ is -invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}}) \iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A,

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, \mathit{Halt}, \ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots, \mathit{Halt}, \ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$.
  These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A $\mathcal{L}(\tau, \prec)$-formula $\varphi(\vec{x})$ is order-invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}}) \iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A,

$$(\mathcal{A}, \prec_1) \models \varphi(\vec{a}) \iff (\mathcal{A}, \prec_2) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, \mathit{Halt}, \ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots, \mathit{Halt}, \ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$.
  These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A $\mathcal{L}(\tau, \prec, +)$-formula $\varphi(\vec{x})$ is addition-invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ $\iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A, and the matching addition relations $+_1, +_2$,

$$(\mathcal{A}, \prec_1, +_1) \models \varphi(\vec{a}) \iff (\mathcal{A}, \prec_2, +_2) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, Halt, \ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots, Halt, \ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$.
  These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A $\mathcal{L}(\tau, \prec, +, \times)$-formula $\varphi(\vec{x})$ is $(+, \times)$-invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}})$ $\iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$,

$$(\mathcal{A}, \prec_1, +_1, \times_1) \models \varphi(\vec{a}) \quad \iff \quad (\mathcal{A}, \prec_2, +_2, \times_2) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots$, *Halt*, $\ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots$, *Halt*, $\ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$. These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A $\mathcal{L}(\tau, \prec, +, \times, \ldots)$-formula $\varphi(\vec{x})$ is Arb-invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}}) \iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$, and other numerical predicates,

$$(\mathcal{A}, \prec_1, +_1, \times_1, \ldots) \models \varphi(\vec{a}) \iff (\mathcal{A}, \prec_2, +_2, \times_2, \ldots) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots$, *Halt*, $\ldots$ on A.
- For this, identify A with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |A|$ and interpret $<, +, \times, \ldots$, *Halt*, $\ldots$ in the natural way.
- To ensure closure under isomorphisms, restrict attention to formulas independent of the particular way of identifying A with $[n]$.
  These formulas are called Arb-invariant.

Definition: Let $\mathcal{L}$ be a logic (e.g., FO, MSO, LFP).

A $\mathcal{L}(\tau, \prec, +, \times, \ldots)$-formula $\varphi(\vec{x})$ is Arb-invariant on $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_\ell^{\mathcal{A}}) \iff$ for all tuples of elements $\vec{a}$ in A, for all linear orders $\prec_1$ and $\prec_2$ on A, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$, and other numerical predicates,

$$(\mathcal{A}, \prec_1, +_1, \times_1, \ldots) \models \varphi(\vec{a}) \iff (\mathcal{A}, \prec_2, +_2, \times_2, \ldots) \models \varphi(\vec{a}).$$

For Arb-invariant sentences, shortly write $\mathcal{A} \models \varphi$ for $(\mathcal{A}, \prec_1, +_1, \times_1 \ldots) \models \varphi$.

Example

- Let $\tau = \emptyset$. An addition-invariant FO$(\tau, \prec, +)$-sentence φ such that

$$\mathcal{A} \models \varphi \iff |A| \text{ is odd.}$$

$$\varphi \ := \ \exists x\, \exists z \Big(x + x = z \ \wedge \ \forall y \, (y \prec z \ \vee \ y = z) \Big)$$

- Similarly, there is an $(+, \times)$-invariant FO$(\tau, \prec, +, \times)$-sentence ψ such that

$$\mathcal{A} \models \psi \iff |A| \text{ is a prime number.}$$

- And there is an Arb-invariant FO$(\tau, \prec, \mathit{Halt})$-sentence χ such that

$$\mathcal{A} \models \chi \iff |A| - 1 \text{ is the index of a Turing machine halting on empty input :}$$

Simply choose $\chi \ := \ \exists x \, \big(\mathit{Halt}(x) \ \wedge \ \forall y \, (y \prec x \ \vee \ y = x) \big)$.

Thus:

order-inv FO $<$ addition-inv FO $<$ $(+, \times)$-inv FO $<$ Arb-invariant FO.

Example

- Let $\tau = \emptyset$. An addition-invariant FO$(\tau, \prec, +)$-sentence φ such that

$$\mathcal{A} \models \varphi \iff |A| \text{ is odd.}$$

$$\varphi := \exists x\, \exists z \left(x + x = z \wedge \forall y \, (y \prec z \vee y = z) \right)$$

- Similarly, there is an $(+, \times)$-invariant FO$(\tau, \prec, +, \times)$-sentence ψ such that

$$\mathcal{A} \models \psi \iff |A| \text{ is a prime number.}$$

- And there is an Arb-invariant FO$(\tau, \prec, \mathit{Halt})$-sentence χ such that

$$\mathcal{A} \models \chi \iff |A|-1 \text{ is the index of a Turing machine halting on empty input :}$$

Simply choose $\chi := \exists x \left(\mathit{Halt}(x) \wedge \forall y \, (y \prec x \vee y = x) \right)$.

Thus:

order-inv FO $<$ addition-inv FO $<$ $(+, \times)$-inv FO $<$ Arb-invariant FO.

Example

- Let $\tau = \emptyset$. An addition-invariant FO$(\tau, \prec, +)$-sentence φ such that

$$\mathcal{A} \models \varphi \quad \iff \quad |A| \text{ is odd.}$$

$$\varphi \;:=\; \exists x\, \exists z \left(x + x = z \;\wedge\; \forall y\, (\, y \prec z \;\vee\; y = z\,)\right)$$

- Similarly, there is an $(+, \times)$-invariant FO$(\tau, \prec, +, \times)$-sentence ψ such that

$$\mathcal{A} \models \psi \quad \iff \quad |A| \text{ is a prime number.}$$

- And there is an Arb-invariant FO$(\tau, \prec, \mathit{Halt})$-sentence χ such that

$$\mathcal{A} \models \chi \quad \iff \quad |A|-1 \text{ is the index of a Turing machine halting on empty input :}$$

Simply choose $\chi \;:=\; \exists x\, (\, \mathit{Halt}(x) \;\wedge\; \forall y\, (y \prec x \;\vee\; y = x)\,)$.

Thus:

order-inv FO $<$ addition-inv FO $<$ $(+, \times)$-inv FO $<$ Arb-invariant FO.

Example

- Let $\tau = \emptyset$. An addition-invariant FO$(\tau, \prec, +)$-sentence φ such that

$$\mathcal{A} \models \varphi \iff |A| \text{ is odd.}$$

$$\varphi \ := \ \exists x\, \exists z \Big(x + x = z \ \wedge \ \forall y \left(y \prec z \ \vee \ y = z \right) \Big)$$

- Similarly, there is an $(+, \times)$-invariant FO$(\tau, \prec, +, \times)$-sentence ψ such that

$$\mathcal{A} \models \psi \iff |A| \text{ is a prime number.}$$

- And there is an Arb-invariant FO$(\tau, \prec, \mathit{Halt})$-sentence χ such that

$$\mathcal{A} \models \chi \iff |A|{-}1 \text{ is the index of a Turing machine halting on empty input :}$$

Simply choose $\chi \ := \ \exists x \left(\mathit{Halt}(x) \ \wedge \ \forall y \left(y \prec x \ \vee \ y = x \right) \right)$.

Thus:

order-inv FO $<$ addition-inv FO $<$ $(+, \times)$-inv FO $<$ Arb-invariant FO.

Example

- Let $\tau = \emptyset$. An addition-invariant FO$(\tau, \prec, +)$-sentence φ such that

$$\mathcal{A} \models \varphi \iff |A| \text{ is odd.}$$

$$\varphi := \exists x \, \exists z \, \Big(x + x = z \ \wedge \ \forall y \, (\, y \prec z \ \vee \ y = z \,) \Big)$$

- Similarly, there is an $(+, \times)$-invariant FO$(\tau, \prec, +, \times)$-sentence ψ such that

$$\mathcal{A} \models \psi \iff |A| \text{ is a prime number.}$$

- And there is an Arb-invariant FO$(\tau, \prec, \mathit{Halt})$-sentence χ such that

$$\mathcal{A} \models \chi \iff |A|-1 \text{ is the index of a Turing machine halting on empty input :}$$

Simply choose $\chi := \exists x \, \big(\mathit{Halt}(x) \ \wedge \ \forall y \, (y \prec x \ \vee \ y = x) \big)$.

Thus:

order-inv FO $<$ addition-inv FO $<$ $(+, \times)$-inv FO $<$ Arb-invariant FO.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable τ-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the τ-properties and queries that belong to the comlexity class $P_{/\text{poly}}$. (Makowsky, 1998)

  $P_{/\text{poly}}$ consists of all problems solvable by circuit families of polynomial size
- Arb-invariant FO precisely captures the τ-properties and queries that belong to the circuit complexity class AC^0.

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth
- $(+, \times)$-invariant FO precisely captures the τ-properties and queries that belong to uniform AC^0.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable τ-properties and queries. (Immerman, Vardi, 1982)

- Arb-invariant LFP precisely captures the τ-properties and queries that belong to the comlexity class $P_{/\text{poly}}$. (Makowsky, 1998)

  $P_{/\text{poly}}$ consists of all problems solvable by circuit families of polynomial size

- Arb-invariant FO precisely captures the τ-properties and queries that belong to the circuit complexity class AC^0.

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth

- $(+, \times)$-invariant FO precisely captures the τ-properties and queries that belong to uniform AC^0.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable τ-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the τ-properties and queries that belong to the comlexity class $P_{/\text{poly}}$. (Makowsky, 1998)

  $P_{/\text{poly}}$ consists of all problems solvable by circuit families of polynomial size
- Arb-invariant FO precisely captures the τ-properties and queries that belong to the circuit complexity class AC^0.

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth
- $(+, \times)$-invariant FO precisely captures the τ-properties and queries that belong to uniform AC^0.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable τ-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the τ-properties and queries that belong to the comlexity class $P_{/\text{poly}}$. (Makowsky, 1998)

  $P_{/\text{poly}}$ consists of all problems solvable by circuit families of polynomial size
- Arb-invariant FO precisely captures the τ-properties and queries that belong to the circuit complexity class AC^0.

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth
- $(+, \times)$-invariant FO precisely captures the τ-properties and queries that belong to uniform AC^0.

Order-invariance is undecidable

Invariant logics are not logics in the strict formal sense:
They have an undecidable syntax. Precisely:

The following problem is undecidable (for binary symbol E and unary symbol C):

ORDER-INVARIANCE ON FINITE $\{E, C\}$-STRUCTURES:

Input: a FO($E, C, \prec$)-sentence φ

Question: Is φ order-invariant on all finite $\{E, C\}$-structures?

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given FO(E)-sentence ψ true for all finite graphs?" is decidable as follows:
  (1) If there is a one-vertex-graph, in which ψ is not true, then stop with output "no". Otherwise, proceed with (2).
  (2) Let χ be a formula that is not order-invariant on structures of size $\geqslant 2$.
  E.g., $\chi := \exists x\big(C(x) \wedge \forall y(x \preceq y)\big)$.
  Stop with output "yes" iff the formula $(\neg\psi \rightarrow \chi)$ is order-invariant.

Order-invariance is undecidable

Invariant logics are not logics in the strict formal sense:
They have an undecidable syntax. Precisely:

The following problem is undecidable (for binary symbol E and unary symbol C):

ORDER-INVARIANCE ON FINITE $\{E, C\}$-STRUCTURES:
Input: a FO($E, C, \prec$)-sentence φ
Question: Is φ order-invariant on all finite $\{E, C\}$-structures?

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given FO(E)-sentence ψ true for all finite graphs?" is decidable as follows:

  (1) If there is a one-vertex-graph, in which ψ is not true, then stop with output "no". Otherwise, proceed with (2).

  (2) Let χ be a formula that is not order-invariant on structures of size $\geqslant 2$.
  E.g., $\chi := \exists x\big(C(x) \wedge \forall y(x \preceq y)\big)$.
  Stop with output "yes" iff the formula $(\neg\psi \rightarrow \chi)$ is order-invariant.

Order-invariance is undecidable

Invariant logics are not logics in the strict formal sense:
They have an undecidable syntax. Precisely:

The following problem is undecidable (for binary symbol E and unary symbol C):

ORDER-INVARIANCE ON FINITE $\{E, C\}$-STRUCTURES:
Input: a FO($E, C, \prec$)-sentence φ
Question: Is φ order-invariant on all finite $\{E, C\}$-structures?

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given FO(E)-sentence ψ true for all finite graphs?" is decidable as follows:
  - (1) If there is a one-vertex-graph, in which ψ is not true, then stop with output "no". Otherwise, proceed with (2).
  - (2) Let χ be a formula that is not order-invariant on structures of size $\geqslant 2$. E.g., $\chi := \exists x\big(C(x) \wedge \forall y(x \preceq y)\big)$. Stop with output "yes" iff the formula $(\neg\psi \to \chi)$ is order-invariant.

Order- and addition-invariance for unary signatures

Let $\tau = \{C_1, \ldots, C_\ell\}$ consist of unary relation symbols.

Theorem: Order-invariance of a given FO(τ)-sentence φ
(on the class of all finite τ-structures) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.
- Commutativity of regular string-languages is decidable.

Theorem: Addition-invariance of a given FO($C, \prec, +$)-sentence φ
(on the class of all finite $\{C\}$-structures) is not decidable.

Order- and addition-invariance for unary signatures

Let $\tau = \{C_1, \ldots, C_\ell\}$ consist of unary relation symbols.

Theorem: Order-invariance of a given FO(τ)-sentence φ (on the class of all finite τ-structures) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.
- Commutativity of regular string-languages is decidable.

Theorem: Addition-invariance of a given FO($C, \prec, +$)-sentence φ (on the class of all finite $\{C\}$-structures) is not decidable.

Order- and addition-invariance for unary signatures

Let $\tau = \{C_1, \ldots, C_\ell\}$ consist of unary relation symbols.

Theorem: Order-invariance of a given FO(τ)-sentence φ (on the class of all finite τ-structures) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.
- Commutativity of regular string-languages is decidable.

Theorem: Addition-invariance of a given FO($C, \prec, +$)-sentence φ (on the class of all finite $\{C\}$-structures) is not decidable.

Order- and addition-invariance for unary signatures

Let $\tau = \{C_1, \ldots, C_\ell\}$ consist of unary relation symbols.

Theorem: Order-invariance of a given FO(τ)-sentence φ (on the class of all finite τ-structures) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.
- Commutativity of regular string-languages is decidable.

Theorem: Addition-invariance of a given FO($C, \prec, +$)-sentence φ (on the class of all finite $\{C\}$-structures) is not decidable.

Order- and addition-invariance for unary signatures

Let $\tau = \{C_1, \ldots, C_\ell\}$ consist of unary relation symbols.

Theorem: Order-invariance of a given FO(τ)-sentence φ (on the class of all finite τ-structures) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.
- Commutativity of regular string-languages is decidable.

Theorem: Addition-invariance of a given FO($C, \prec, +$)-sentence φ (on the class of all finite $\{C\}$-structures) is not decidable.

Overview

FO $<$ order-invariant FO

Theorem (Gurevich):

Let $\tau := \{\subseteq\}$ be a signature consisting of a single binary relation symbol $\subseteq$.
For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-invariant FO($\tau, \prec$)-sentence φ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \varphi_{even} \iff |X| \text{ is even.}$$

But there is no FO(τ)-sentence ψ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \psi_{even} \iff |X| \text{ is even.}$$

Proof:

Part 1: φ_{even} expresses that there is a set z that contains the first (w.r.t. $\prec$) atom of X, every other (w.r.t. $\prec$) atom of X, but not the last (w.r.t. $\prec$) atom of X.

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that $\mathcal{B}_X \equiv_r \mathcal{B}_Y$ for all finite X, Y of cardinality $> 2^r$. □

FO $<$ order-invariant FO

Theorem (Gurevich):

Let $\tau := \{\subseteq\}$ be a signature consisting of a single binary relation symbol $\subseteq$.
For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-invariant FO($\tau, \prec$)-sentence φ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \varphi_{even} \iff |X| \text{ is even.}$$

But there is no FO(τ)-sentence ψ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \psi_{even} \iff |X| \text{ is even.}$$

Proof:

Part 1: φ_{even} expresses that there is a set z that contains the first (w.r.t. $\prec$) atom of X, every other (w.r.t. $\prec$) atom of X, but not the last (w.r.t. $\prec$) atom of X.

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that
$\mathcal{B}_X \equiv_r \mathcal{B}_Y$ for all finite X, Y of cardinality $> 2^r$. □

FO $<$ order-invariant FO

Theorem (Gurevich):

Let $\tau := \{\subseteq\}$ be a signature consisting of a single binary relation symbol $\subseteq$.
For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-invariant FO($\tau, \prec$)-sentence φ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \varphi_{even} \iff |X| \text{ is even.}$$

But there is no FO(τ)-sentence ψ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \psi_{even} \iff |X| \text{ is even.}$$

Proof:

Part 1: φ_{even} expresses that there is a set z that contains the first (w.r.t. $\prec$) atom of X, every other (w.r.t. $\prec$) atom of X, but not the last (w.r.t. $\prec$) atom of X.

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that $\mathcal{B}_X \equiv_r \mathcal{B}_Y$ for all finite X, Y of cardinality $> 2^r$. $\square$

FO $<$ order-invariant FO

Theorem (Gurevich):

Let $\tau := \{\subseteq\}$ be a signature consisting of a single binary relation symbol $\subseteq$.
For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-invariant FO$(\tau, \prec)$-sentence φ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \varphi_{even} \iff |X| \text{ is even.}$$

But there is no FO(τ)-sentence ψ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \psi_{even} \iff |X| \text{ is even.}$$

Proof:

Part 1: φ_{even} expresses that there is a set z that contains the first (w.r.t. $\prec$) atom of X, every other (w.r.t. $\prec$) atom of X, but not the last (w.r.t. $\prec$) atom of X.

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that
$\mathcal{B}_X \equiv_r \mathcal{B}_Y$ for all finite X, Y of cardinality $> 2^r$. □

FO $<$ order-invariant FO

Theorem (Gurevich):

Let $\tau := \{\subseteq\}$ be a signature consisting of a single binary relation symbol $\subseteq$.
For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-invariant FO$(\tau, \prec)$-sentence φ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \varphi_{even} \iff |X| \text{ is even.}$$

But there is no FO(τ)-sentence ψ_{even} such that for every finite set X:

$$\mathcal{B}_X \models \psi_{even} \iff |X| \text{ is even.}$$

Proof:

Part 1: φ_{even} expresses that there is a set z that contains the first (w.r.t. $\prec$) atom of X, every other (w.r.t. $\prec$) atom of X, but not the last (w.r.t. $\prec$) atom of X.

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that
$\mathcal{B}_X \equiv_r \mathcal{B}_Y$ for all finite X, Y of cardinality $> 2^r$. □

Successor-invariant FO

By a much more elaborate construction, one can also show:

Theorem (Rossman, LICS'03)

On the class of all finite structures,
successor-invariant FO is strictly more expressive than FO.

FO+MOD$_2$ $<$ order-invariant FO+MOD$_2$

FO+MOD$_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{r \bmod 2} x\, \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo 2.

Theorem (Niemistö):

Let $\tau := \{E\}$ be a signature consisting of a single binary relation symbol E.

There is an order-invariant FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V, E)$ iff

(1) G is a disjoint union of directed cycles, and

(2) the number of even-length cycles is even.

Proof:

- (1) can be expressed in FO: "every node has in- and out-degree 1"
- Every G satisfying (1) is the cycle decomposition of a permutation π.
- G has an even number of even-length cycles $\iff$
  π is an even permuatation, i.e., sgn(π) $= 1$ $\iff$
  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

FO+MOD$_2$ $<$ order-invariant FO+MOD$_2$

FO+MOD$_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{r \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo 2.

Theorem (Niemistö):

Let $\tau := \{E\}$ be a signature consisting of a single binary relation symbol E.

There is an order-invariant FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V, E)$ iff

(1) G is a disjoint union of directed cycles, and

(2) the number of even-length cycles is even.

Proof:

- (1) can be expressed in FO: “every node has in- and out-degree 1”
- Every G satisfying (1) is the cycle decomposition of a permutation π.
- G has an even number of even-length cycles $\iff$
  π is an even permuatation, i.e., $\text{sgn}(\pi) = 1$ $\iff$
  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

FO+MOD$_2$ $<$ order-invariant FO+MOD$_2$

FO+MOD$_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{r \bmod 2} x\, \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo 2.

Theorem (Niemistö):

Let $\tau := \{E\}$ be a signature consisting of a single binary relation symbol E.

There is an order-invariant FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V, E)$ iff

(1) G is a disjoint union of directed cycles, and

(2) the number of even-length cycles is even.

Proof:

- (1) can be expressed in FO: “every node has in- and out-degree 1”
- Every G satisfying (1) is the cycle decomposition of a permutation π.
- G has an even number of even-length cycles $\iff$

  π is an even permuatation, i.e., $\text{sgn}(\pi) = 1 \iff$

  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

FO+MOD$_2$ $<$ order-invariant FO+MOD$_2$

FO+MOD$_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{r \bmod 2} x\, \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo 2.

Theorem (Niemistö):

Let $\tau := \{E\}$ be a signature consisting of a single binary relation symbol E.

There is an order-invariant FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V, E)$ iff

(1) G is a disjoint union of directed cycles, and

(2) the number of even-length cycles is even.

Proof:

- (1) can be expressed in FO: "every node has in- and out-degree 1"
- Every G satisfying (1) is the cycle decomposition of a permutation π.
- G has an even number of even-length cycles $\iff$
  π is an even permuatation, i.e., $\text{sgn}(\pi) = 1$ $\iff$
  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

FO+MOD$_2$ $<$ order-invariant FO+MOD$_2$

FO+MOD$_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{r \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo 2.

Theorem (Niemistö):

Let $\tau := \{E\}$ be a signature consisting of a single binary relation symbol E.

There is an order-invariant FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V, E)$ iff

(1) G is a disjoint union of directed cycles, and

(2) the number of even-length cycles is even.

Proof:

- (1) can be expressed in FO: "every node has in- and out-degree 1"
- Every G satisfying (1) is the cycle decomposition of a permutation π.
- G has an even number of even-length cycles $\iff$
  π is an even permuatation, i.e., $\mathrm{sgn}(\pi) = 1$ $\iff$
  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, Same\_Row, Same\_Column)$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, \mathit{Same\_Row}, \mathit{Same\_Column})$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, \mathit{Same\_Row}, \mathit{Same\_Column})$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, \mathit{Same\_Row}, \mathit{Same\_Column})$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, Same\_Row, Same\_Column)$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Order-invariant MSO

CMSO : the extension of MSO by modulo counting quantifiers

$\exists^{r \bmod m} x\, \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent r modulo m.

Known:

- On trees:
  Order-invariant MSO $=$ CMSO (Courcelle 1996, Lapoire 1998)

- On the class of all finite structures:
  Order-invariant MSO $>$ CMSO (Ganzow, Rubin 2008)

  The separating example:

  - Consider 2-dimensional grids, represented as structures of the form $(A, \mathit{Same\_Row}, \mathit{Same\_Column})$.
  - Order-invariant MSO can express that the number of columns is a multiple of the number of rows.
  - CMSO cannot (for showing this, use a variant of EF-games).

Overview

Neighborhoods

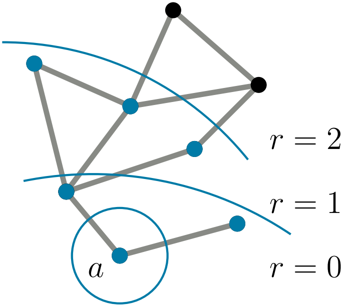
Graph $G = (V, E)$

Distance $dist(u, v)$: length of a shortest path between u, v in G.

Shell $S_r(a)$ of nodes at distance exactly r from a.

Ball $N_r(a)$ of radius r at a in G.

Neighborhood $\mathcal{N}_r(a)$ of radius r at a in G.



Neighborhoods

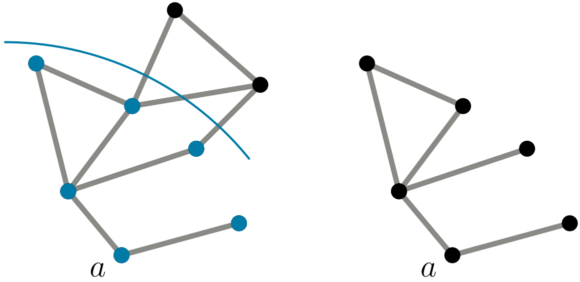
Graph $G = (V, E)$

Distance $dist(u, v)$: length of a shortest path between u, v in G.

Shell $S_r(a)$ of nodes at distance exactly r from a.

Ball $N_r(a)$ of radius r at a in G.

Neighborhood $\mathcal{N}_r(a)$ of radius r at a in G.



Gaifman-local queries

- For a list $a = a_1, \ldots, a_k$ of nodes, $N_r^G(a) = N_r^G(a_1) \cup \cdots \cup N_r^G(a_k)$.
- The r-neighborhood $\mathcal{N}_r^G(a)$ is the structure $(G_{|N_r^G(a)}, a)$ consisting of the induced subgraph of G on $N_r^G(a)$, together with the distinguished nodes a.

Definition: Let q be a k-ary graph query. Let $f : \mathbb{N} \to \mathbb{N}$.
q is called $f(n)$-local if there is an n_0 such that for every $n \geqslant n_0$ and every graph G with $|V^G| = n$, the following is true for all k-tuples a and b of nodes:

$$\text{if} \quad \mathcal{N}_{f(n)}^G(a) \cong \mathcal{N}_{f(n)}^G(b) \quad \text{then} \quad a \in q(G) \iff b \in q(G).$$

Gaifman-locality of FO

Theorem:

- For every graph query q that is FO-definable,
  there is a constant c such that q is c-local.
  (Hella, Libkin, Nurmonen 1990s; Gaifman '82)

- For every graph query q that is FO-definable on ordered graphs
  (for short: q is definable in order-invariant FO),
  there is a constant c such that q is c-local.
  (Grohe, Schwentick '98)

- For every graph query q that is FO-definable on graphs with arbitrary numerical predicates (for short: q is definable in Arb-invariant FO),
  there is a constant c such that q is $(\log n)^c$-local.
  (Anderson, van Melkebeek, S., Segoufin '11)

Gaifman-locality of FO

Theorem:

- For every graph query q that is FO-definable, there is a constant c such that q is c-local.
(Hella, Libkin, Nurmonen 1990s; Gaifman '82)

- For every graph query q that is FO-definable on ordered graphs (for short: q is definable in order-invariant FO), there is a constant c such that q is c-local.
(Grohe, Schwentick '98)

- For every graph query q that is FO-definable on graphs with arbitrary numerical predicates (for short: q is definable in Arb-invariant FO), there is a constant c such that q is $(\log n)^c$-local.
(Anderson, van Melkebeek, S., Segoufin '11)

Gaifman-locality of FO

Theorem:

- For every graph query q that is FO-definable, there is a constant c such that q is c-local.
  (Hella, Libkin, Nurmonen 1990s; Gaifman '82)

- For every graph query q that is FO-definable on ordered graphs (for short: q is definable in order-invariant FO), there is a constant c such that q is c-local.
  (Grohe, Schwentick '98)

- For every graph query q that is FO-definable on graphs with arbitrary numerical predicates (for short: q is definable in Arb-invariant FO), there is a constant c such that q is $(\log n)^c$-local.
  (Anderson, van Melkebeek, S., Segoufin '11)

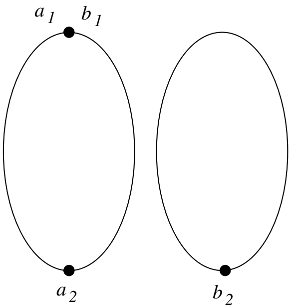
Use locality for proving non-expressibility

Example: The reachability query

$$\text{REACH}(G) := \{ (a_1, a_2) : \text{there is a directed path from } a_1 \text{ to } a_2 \text{ in } G \}$$

is not $\frac{n}{5}$-local an thus cannot be expressed in Arb-invariant FO.

Proof: Consider the graph G:



Use locality for proving non-expressibility

Similarly, one obtains that the following queries are not definable in Arb-invariant FO:

- Does node x lie on a cycle?
- Does node x belong to a connected component that is acyclic?
- Is node x reachable from a node that belongs to a triangle?
- Do nodes x and y have the same distance to node z?

Proof of Gaifman-locality theorem (1/5)

For every query q expressible by Arb-invariant FO, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.

Idea: Use known lower bounds in circuit complexity!

- Let q be expressible by an Arb-invariant FO formula.
- Then, q can be computed by an AC^0 circuit family $\mathcal{C}$ (Immerman '87).
- Assume that q is *not* $(\log n)^c$-local (for any $c \in \mathbb{N}$), and modify $\mathcal{C}$ to obtain an AC^0 circuit family computing

$$\text{PARITY} := \{w \in \{0,1\}^* : |w|_1 \text{ is even}\}.$$

- This contradicts known lower bounds in circuit complexity theory (Håstad'86).

Proof of Gaifman-locality theorem (1/5)

For every query q expressible by Arb-invariant FO, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.

Idea: Use known lower bounds in circuit complexity!

- Let q be expressible by an Arb-invariant FO formula.
- Then, q can be computed by an AC^0 circuit family $\mathcal{C}$ (Immerman '87).
- Assume that q is *not* $(\log n)^c$-local (for any $c \in \mathbb{N}$), and modify $\mathcal{C}$ to obtain an AC^0 circuit family computing

$$\text{PARITY} := \{w \in \{0,1\}^* : |w|_1 \text{ is even}\}.$$

- This contradicts known lower bounds in circuit complexity theory (Håstad'86).

Proof of Gaifman-locality theorem (1/5)

For every query q expressible by Arb-invariant FO*, there is a $c \in \mathbb{N}$ such that q is* $(\log n)^c$*-local.*

Idea: Use known lower bounds in circuit complexity!

- Let q be expressible by an Arb-invariant FO formula.
- Then, q can be computed by an AC^0 circuit family $\mathcal{C}$ (Immerman '87).
- Assume that q is *not* $(\log n)^c$-local (for any $c \in \mathbb{N}$), and modify $\mathcal{C}$ to obtain an AC^0 circuit family computing

$$\text{PARITY} := \{w \in \{0,1\}^* : |w|_1 \text{ is even}\}.$$

- This contradicts known lower bounds in circuit complexity theory (Håstad'86).

Proof of Gaifman-locality theorem (1/5)

For every query q expressible by Arb-invariant FO, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.

Idea: Use known lower bounds in circuit complexity!

- Let q be expressible by an Arb-invariant FO formula.
- Then, q can be computed by an AC^0 circuit family $\mathcal{C}$ (Immerman '87).
- Assume that q is *not* $(\log n)^c$-local (for any $c \in \mathbb{N}$), and modify $\mathcal{C}$ to obtain an AC^0 circuit family computing

$$\text{PARITY} := \{w \in \{0,1\}^* : |w|_1 \text{ is even}\}.$$

- This contradicts known lower bounds in circuit complexity theory (Håstad'86).

Proof of Gaifman-locality theorem (2/5)

How to compute a graph query $q(x)$ by an AC^0 circuit family $\mathcal{C}$?

- Represent graph $G = (V, E)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V$ by the bitstring $\beta(a)$ of the form 0^*10^*, carrying the 1 at position i iff node a corresponds to the i-th row/column of the adjacency matrix.
- Let $Rep(G, a)$ be the set of all bitstrings $\beta(G)\beta(a)$, corresponding to all adjacency matrices of G (i.e., all ways of embedding V in $\{1, \dots, |V|\}$). Thus, $Rep(G, a)$ is the set of all bitstrings representing (G, a).
- A unary graph query $q(x)$ is computed by a circuit family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ iff the following is true:
  for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by a circuit family of constant depth and polynomial size.
  (implicit in Immerman'87)

Proof of Gaifman-locality theorem (2/5)

How to compute a graph query $q(x)$ by an AC^0 circuit family $\mathcal{C}$?

- Represent graph $G = (V, E)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V$ by the bitstring $\beta(a)$ of the form 0^*10^*, carrying the 1 at position i iff node a corresponds to the i-th row/column of the adjacency matrix.
- Let $Rep(G, a)$ be the set of all bitstrings $\beta(G)\beta(a)$, corresponding to all adjacency matrices of G (i.e., all ways of embedding V in $\{1, \ldots, |V|\}$). Thus, $Rep(G, a)$ is the set of all bitstrings representing (G, a).
- A unary graph query $q(x)$ is computed by a circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ iff the following is true:
  for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by a circuit family of constant depth and polynomial size.
  (implicit in Immerman'87)

Proof of Gaifman-locality theorem (2/5)

How to compute a graph query $q(x)$ by an AC^0 *circuit family* $\mathcal{C}$*?*

- Represent graph $G = (V, E)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V$ by the bitstring $\beta(a)$ of the form 0^*10^*, carrying the 1 at position i iff node a corresponds to the i-th row/column of the adjacency matrix.
- Let $Rep(G, a)$ be the set of all bitstrings $\beta(G)\beta(a)$, corresponding to all adjacency matrices of G (i.e., all ways of embedding V in $\{1, \ldots, |V|\}$). Thus, $Rep(G, a)$ is the set of all bitstrings representing (G, a).
- A unary graph query $q(x)$ is computed by a circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ iff the following is true: for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by a circuit family of constant depth and polynomial size. (implicit in Immerman'87)

Proof of Gaifman-locality theorem (2/5)

How to compute a graph query $q(x)$ by an AC^0 circuit family $\mathcal{C}$?

- Represent graph $G = (V, E)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V$ by the bitstring $\beta(a)$ of the form 0^*10^*, carrying the 1 at position i iff node a corresponds to the i-th row/column of the adjacency matrix.
- Let $Rep(G, a)$ be the set of all bitstrings $\beta(G)\beta(a)$, corresponding to all adjacency matrices of G (i.e., all ways of embedding V in $\{1, \ldots, |V|\}$). Thus, $Rep(G, a)$ is the set of all bitstrings representing (G, a).
- A unary graph query $q(x)$ is computed by a circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ iff the following is true: for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by a circuit family of constant depth and polynomial size. (implicit in Immerman'87)

Proof of Gaifman-locality theorem (2/5)

How to compute a graph query $q(x)$ by an AC^0 circuit family $\mathcal{C}$?

- Represent graph $G = (V, E)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V$ by the bitstring $\beta(a)$ of the form 0^*10^*, carrying the 1 at position i iff node a corresponds to the i-th row/column of the adjacency matrix.
- Let $Rep(G, a)$ be the set of all bitstrings $\beta(G)\beta(a)$, corresponding to all adjacency matrices of G (i.e., all ways of embedding V in $\{1, \ldots, |V|\}$). Thus, $Rep(G, a)$ is the set of all bitstrings representing (G, a).
- A unary graph query $q(x)$ is computed by a circuit family $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ iff the following is true: for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by a circuit family of constant depth and polynomial size.
(implicit in Immerman'87)

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$, but $a \in q(G)$ and $b \notin q(G)$.
For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: (Håstad '86)
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell \cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell \cdot m^{1/(d-1)}} > 2^{\ell \cdot (\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$, but $a \in q(G)$ and $b \notin q(G)$.
For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: *(Håstad '86)*
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell\cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell\cdot m^{1/(d-1)}} > 2^{\ell\cdot(\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V,E)$, $a \in V$, $\gamma \in Rep(G,a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V,E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\quad \mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a,b) > 2m$.

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V,E)$, $a, b \in V$ such that $\mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$ and $dist(a,b) > 2m$. Let circuit C accept all strings in $Rep(G,a)$ and reject all strings in $Rep(G,b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: (Håstad '86)

There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell\cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell\cdot m^{1/(d-1)}} > 2^{\ell\cdot(\log n)^2} = n^{\ell \log n} > p(n)$. $\square$

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: (Håstad '86)
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell \cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell \cdot m^{1/(d-1)}} > 2^{\ell \cdot (\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: (Håstad '86)
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell \cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell \cdot m^{1/(d-1)}} > 2^{\ell \cdot (\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\quad \mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$, but $a \in q(G)$ and $b \notin q(G)$.
For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
*Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.*

Theorem: (Håstad '86)
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell \cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell \cdot m^{1/(d-1)}} > 2^{\ell\cdot(\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\quad \mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Theorem: *(Håstad '86)*
There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell\cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell\cdot m^{1/(d-1)}} > 2^{\ell\cdot(\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:

*Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}^G_m(a) \cong \mathcal{N}^G_m(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.*

Theorem: (Håstad '86)

There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell\cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell\cdot m^{1/(d-1)}} > 2^{\ell\cdot(\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (3/5)

Let $q(x)$ be a unary graph query expressible in Arb-invariant FO. Let $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ be a circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V, E)$, $a \in V$, $\gamma \in Rep(G, a)$: $a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.
Thus: For all c, n_0 there exist $n > n_0$, $G = (V, E)$ with n nodes, $a, b \in V$ such that for $m := (\log n)^c$, $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$, but $a \in q(G)$ and $b \notin q(G)$.

For simplicity, consider the special case that $dist(a, b) > 2m$.

Key Lemma:

*Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.*

Theorem: *(Håstad '86)*

There exist $\ell, m_0 > 0$ such that for all $m \geqslant m_0$, no circuit of depth d and size $2^{\ell \cdot m^{1/(d-1)}}$ computes parity on m bits.

Contradiction for $c = 2d$, since $2^{\ell \cdot m^{1/(d-1)}} > 2^{\ell \cdot (\log n)^2} = n^{\ell \log n} > p(n)$. □

Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.

Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.

Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.

Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

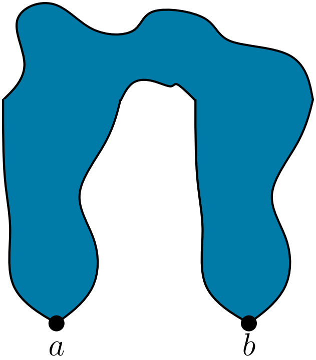
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

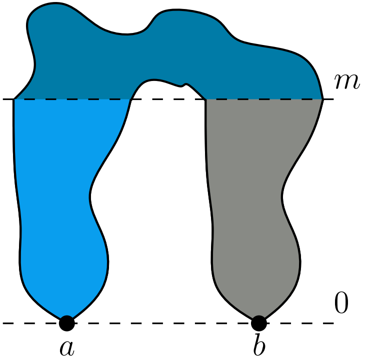
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

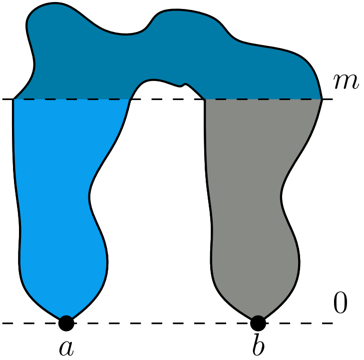
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

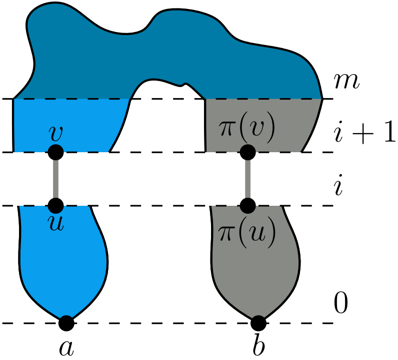
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

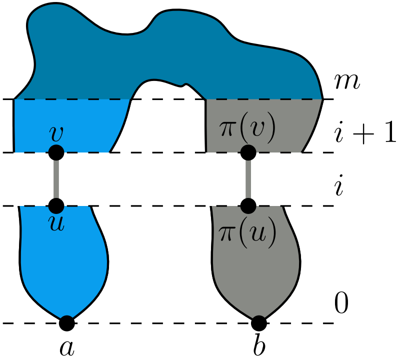
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

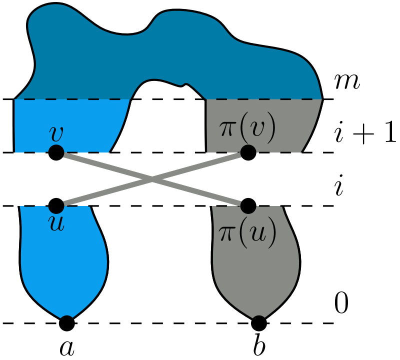
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.

Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

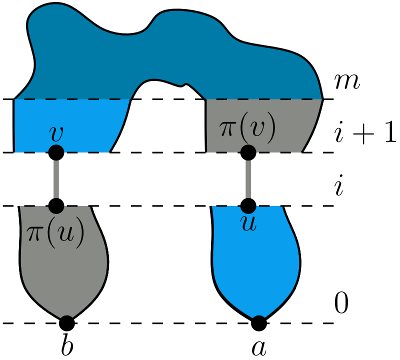
For $i \in \{0, 1, \dots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.

Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

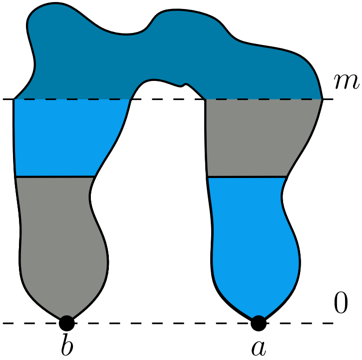
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

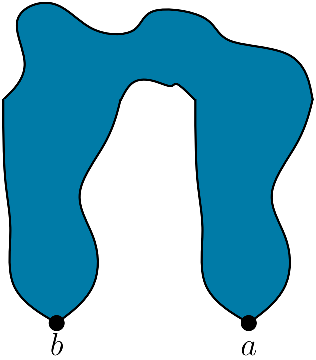
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.

Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

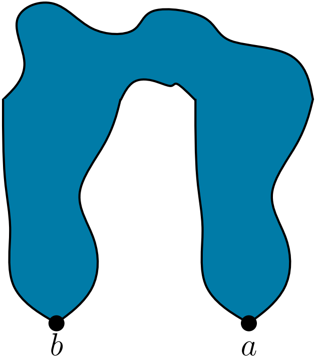
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

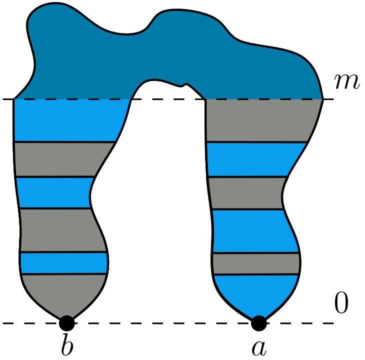
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.

Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

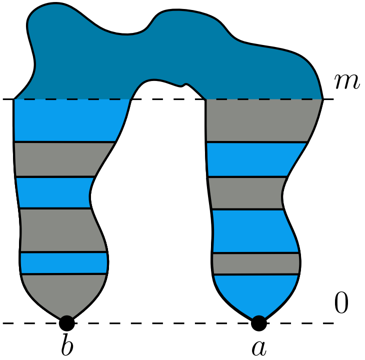
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

Proof:

Consider $w \in \{0, 1\}^m$.

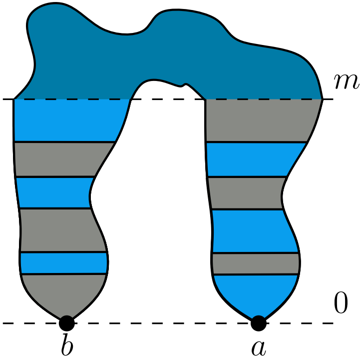
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *Swap the endpoints of the edges leaving $N_i(a)$ with the corresponding endpoints of the edges leaving $N_i(b)$.*

The resulting graph $G_w \cong G$.

$$(G_w, a) \cong \begin{cases} (G, a), & \text{if } |w|_1 \text{ even} \\ (G, b), & \text{if } |w|_1 \text{ odd} \end{cases}$$

Circuit C distinguishes these cases.



Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

How to obtain $\tilde{C}$ from C?

- Consider C for a fixed input string $\gamma \in Rep(G, a)$.
- Fix all input bits (as in γ) that do *not* correspond to potential edges between the shells S_i and S_{i+1}, for $i < m$.
- For all $i < m$ and all $u \in S_i(a)$, $v \in S_{i+1}(a)$ consider the potential edges $e = \{u, v\}$, $e' = \{\pi(u), \pi(v)\}$, $\tilde{e} = \{u, \pi(v)\}$, $\tilde{e}' = \{\pi(u), v\}$.
- Replace input gates of C as follows:

$$e \text{ by } (e \wedge \neg w_i) \qquad e' \text{ by } (e' \wedge \neg w_i)$$
$$\tilde{e} \text{ by } (e \wedge w_i) \qquad \tilde{e}' \text{ by } (e' \wedge w_i)$$

- This yields a circuit $\tilde{C}$ of the same size and depth as C which, on input $w \in \{0, 1\}^m$ does the same as C on input (G_w, a).
  Thus, $\tilde{C}$ accepts iff $|w|_1$ is even. □

Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

How to obtain $\tilde{C}$ from C?

- Consider C for a fixed input string $\gamma \in Rep(G, a)$.
- Fix all input bits (as in γ) that do *not* correspond to potential edges between the shells S_i and S_{i+1}, for $i < m$.
- For all $i < m$ and all $u \in S_i(a)$, $v \in S_{i+1}(a)$ consider the potential edges $e = \{u, v\}$, $e' = \{\pi(u), \pi(v)\}$, $\tilde{e} = \{u, \pi(v)\}$, $\tilde{e}' = \{\pi(u), v\}$.
- Replace input gates of C as follows:

$$e \text{ by } (e \wedge \neg w_i) \qquad e' \text{ by } (e' \wedge \neg w_i)$$
$$\tilde{e} \text{ by } (e \wedge w_i) \qquad \tilde{e}' \text{ by } (e' \wedge w_i)$$

- This yields a circuit $\tilde{C}$ of the same size and depth as C which, on input $w \in \{0,1\}^m$ does the same as C on input (G_w, a).
  Thus, $\tilde{C}$ accepts iff $|w|_1$ is even. □

Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

How to obtain $\tilde{C}$ from C?

- Consider C for a fixed input string $\gamma \in Rep(G, a)$.
- Fix all input bits (as in γ) that do *not* correspond to potential edges between the shells S_i and S_{i+1}, for $i < m$.
- For all $i < m$ and all $u \in S_i(a)$, $v \in S_{i+1}(a)$ consider the potential edges $e = \{u, v\}$, $e' = \{\pi(u), \pi(v)\}$, $\tilde{e} = \{u, \pi(v)\}$, $\tilde{e}' = \{\pi(u), v\}$.
- Replace input gates of C as follows:

$$e \text{ by } (e \wedge \neg w_i) \qquad e' \text{ by } (e' \wedge \neg w_i)$$
$$\tilde{e} \text{ by } (e \wedge w_i) \qquad \tilde{e}' \text{ by } (e' \wedge w_i)$$

- This yields a circuit $\tilde{C}$ of the same size and depth as C which, on input $w \in \{0, 1\}^m$ does the same as C on input (G_w, a).
  Thus, $\tilde{C}$ accepts iff $|w|_1$ is even. □

Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ such that $\mathcal{N}_m^G(a) \cong \mathcal{N}_m^G(b)$ and $dist(a, b) > 2m$. Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$. Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity on m bits.

How to obtain $\tilde{C}$ from C?

- Consider C for a fixed input string $\gamma \in Rep(G, a)$.
- Fix all input bits (as in γ) that do *not* correspond to potential edges between the shells S_i and S_{i+1}, for $i < m$.
- For all $i < m$ and all $u \in S_i(a)$, $v \in S_{i+1}(a)$ consider the potential edges $e = \{u, v\}$, $e' = \{\pi(u), \pi(v)\}$, $\tilde{e} = \{u, \pi(v)\}$, $\tilde{e}' = \{\pi(u), v\}$.
- Replace input gates of C as follows:

$$e \text{ by } (e \wedge \neg w_i) \qquad e' \text{ by } (e' \wedge \neg w_i)$$
$$\tilde{e} \text{ by } (e \wedge w_i) \qquad \tilde{e}' \text{ by } (e' \wedge w_i)$$

- This yields a circuit $\tilde{C}$ of the same size and depth as C which, on input $w \in \{0, 1\}^m$ does the same as C on input (G_w, a). Thus, $\tilde{C}$ accepts iff $|w|_1$ is even. □

Summary: Gaifman-locality of Arb-invariant FO (1/2)

Theorem: *(Anderson, Melkebeek, S., Segoufin '11)*

(a) *For every query q expressible by Arb-invariant FO there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.*

(b) *For every $d \in \mathbb{N}$ there is a $(+,\times)$-invariant FO query that is not $(\log n)^d$-local.*

The query $q_d(x)$ states:

(1) The graph has at most $(\log n)^{d+1}$ non-isolated vertices.

(Use the polylog-counting capability of FO$(+,\times)$)

(2) Node x is reachable from a node that belongs to a triangle.

(Show that in graphs satisfying (1), reachability by paths of length $(\log n)^{d+1}$ can be expressed in $(+,\times)$-invariant FO)

Note: This query is not $(\log n)^d$-local.

Summary: Gaifman-locality of Arb-invariant FO (1/2)

Theorem: *(Anderson, Melkebeek, S., Segoufin '11)*

(a) *For every query q expressible by Arb-invariant* FO *there is a* $c \in \mathbb{N}$ *such that q is* $(\log n)^c$*-local.*

(b) *For every* $d \in \mathbb{N}$ *there is a* $(+,\times)$*-invariant* FO *query that is not* $(\log n)^d$*-local.*

The query $q_d(x)$ states:

(1) The graph has at most $(\log n)^{d+1}$ non-isolated vertices.

(Use the polylog-counting capability of FO$(+,\times)$)

(2) Node x is reachable from a node that belongs to a triangle.

(Show that in graphs satisfying (1), reachability by paths of length $(\log n)^{d+1}$ can be expressed in $(+,\times)$-invariant FO)

Note: This query is not $(\log n)^d$-local.

Summary: Gaifman-locality of Arb-invariant FO (1/2)

Theorem: *(Anderson, Melkebeek, S., Segoufin '11)*

(a) *For every query q expressible by Arb-invariant FO there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.*

(b) *For every $d \in \mathbb{N}$ there is a $(+, \times)$-invariant FO query that is not $(\log n)^d$-local.*

The query $q_d(x)$ states:

(1) The graph has at most $(\log n)^{d+1}$ non-isolated vertices.

(Use the polylog-counting capability of FO$(+, \times)$)

(2) Node x is reachable from a node that belongs to a triangle.

(Show that in graphs satisfying (1), reachability by paths of length $(\log n)^{d+1}$ can be expressed in $(+, \times)$-invariant FO)

Note: This query is not $(\log n)^d$-local.

Summary: Gaifman-locality of Arb-invariant FO (1/2)

Theorem: *(Anderson, Melkebeek, S., Segoufin '11)*

(a) *For every query q expressible by Arb-invariant* FO *there is a* $c \in \mathbb{N}$ *such that* q *is* $(\log n)^c$*-local.*

(b) *For every* $d \in \mathbb{N}$ *there is a* $(+, \times)$*-invariant* FO *query that is not* $(\log n)^d$*-local.*

The query $q_d(x)$ states:

(1) The graph has at most $(\log n)^{d+1}$ non-isolated vertices.

(Use the polylog-counting capability of FO$(+, \times)$)

(2) Node x is reachable from a node that belongs to a triangle.

(Show that in graphs satisfying (1), reachability by paths of length $(\log n)^{d+1}$ can be expressed in $(+, \times)$-invariant FO)

Note: This query is not $(\log n)^d$-local.

Summary: Gaifman-locality of Arb-invariant FO (1/2)

Theorem: *(Anderson, Melkebeek, S., Segoufin '11)*

(a) *For every query q expressible by Arb-invariant FO there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-local.*

(b) *For every $d \in \mathbb{N}$ there is a $(+, \times)$-invariant FO query that is not $(\log n)^d$-local.*

The query $q_d(x)$ states:

(1) The graph has at most $(\log n)^{d+1}$ non-isolated vertices.

(Use the polylog-counting capability of FO$(+, \times)$)

(2) Node x is reachable from a node that belongs to a triangle.

(Show that in graphs satisfying (1), reachability by paths of length $(\log n)^{d+1}$ can be expressed in $(+, \times)$-invariant FO)

Note: This query is not $(\log n)^d$-local.

Summary: Gaifman-locality of Arb-invariant FO (2/2)

Goal: Show that in graphs with $\leqslant (\log n)^c$ non-isolated vertices, reachability by paths of length $(\log n)^c$ can be expressed in $(+, \times)$-invariant FO.

Lemma: *(Durand, Lautemann, More '07)*

For every $c \in \mathbb{N}$ there is a $\mathrm{FO}(<, +, \times, S)$*-formula $bij_c(x, y)$ such that for all $n \in \mathbb{N}$, all $S \subseteq [n] := \{0, \ldots, n-1\}$, all $a, i < n$ we have*

$$([n], <, +, \times, S) \models bij_c(a, i) \iff |S| < (\log n)^c \text{ and } a \text{ is the } i\text{-th smallest element of } S.$$

- Using this, identify the non-isolated vertices with numbers $< (\log n)^c$ and represent them by bitstrings of length $c \log \log n$.
- Identify an arbitrary vertex of G with a number $< n$, whose binary representation encodes a sequence of $\ell(n) := \frac{\log n}{c \log \log n}$ non-isolated vertices.
- Use this to express that there is a path of length $\ell(n)$ from node x to node y.
- Iterate this for $c+1$ times to express that there is a path of length $\ell(n)^{c+1} \geqslant (\log n)^c$ from x to y. □

Summary: Gaifman-locality of Arb-invariant FO (2/2)

Goal: Show that in graphs with $\leqslant (\log n)^c$ non-isolated vertices, reachability by paths of length $(\log n)^c$ can be expressed in $(+,\times)$-invariant FO.

Lemma: *(Durand, Lautemann, More '07)*

For every $c \in \mathbb{N}$ there is a FO$(<,+,\times,S)$*-formula $bij_c(x,y)$ such that for all $n \in \mathbb{N}$, all $S \subseteq [n] := \{0,\ldots,n{-}1\}$, all $a, i < n$ we have*

$$([n],<,+,\times,S) \models bij_c(a,i) \iff |S| < (\log n)^c \text{ and } a \text{ is the } i\text{-th smallest element of } S.$$

- Using this, identify the non-isolated vertices with numbers $< (\log n)^c$ and represent them by bitstrings of length $c \log\log n$.
- Identify an arbitrary vertex of G with a number $< n$, whose binary representation encodes a sequence of $\ell(n) := \frac{\log n}{c \log\log n}$ non-isolated vertices.
- Use this to express that there is a path of length $\ell(n)$ from node x to node y.
- Iterate this for $c{+}1$ times to express that there is a path of length $\ell(n)^{c+1} \geqslant (\log n)^c$ from x to y.

Summary: Gaifman-locality of Arb-invariant FO (2/2)

Goal: Show that in graphs with $\leqslant (\log n)^c$ non-isolated vertices, reachability by paths of length $(\log n)^c$ can be expressed in $(+,\times)$-invariant FO.

Lemma: *(Durand, Lautemann, More '07)*

For every $c \in \mathbb{N}$ there is a $\mathrm{FO}(<,+,\times,S)$*-formula* $bij_c(x,y)$ *such that for all* $n \in \mathbb{N}$*, all* $S \subseteq [n] := \{0,\ldots,n{-}1\}$*, all* $a, i < n$ *we have*

$$([n],<,+,\times,S) \models bij_c(a,i) \iff |S| < (\log n)^c \text{ and } a \text{ is the } i\text{-th smallest element of } S.$$

- Using this, identify the non-isolated vertices with numbers $< (\log n)^c$ and represent them by bitstrings of length $c \log\log n$.
- Identify an arbitrary vertex of G with a number $< n$, whose binary representation encodes a sequence of $\ell(n) := \frac{\log n}{c \log\log n}$ non-isolated vertices.
- Use this to express that there is a path of length $\ell(n)$ from node x to node y.
- Iterate this for $c{+}1$ times to express that there is a path of length $\ell(n)^{c+1} \geqslant (\log n)^c$ from x to y.

Summary: Gaifman-locality of Arb-invariant FO (2/2)

Goal: Show that in graphs with $\leqslant (\log n)^c$ non-isolated vertices, reachability by paths of length $(\log n)^c$ can be expressed in $(+, \times)$-invariant FO.

Lemma: *(Durand, Lautemann, More '07)*

For every $c \in \mathbb{N}$ there is a $\mathrm{FO}(<, +, \times, S)$*-formula* $bij_c(x, y)$ *such that for all* $n \in \mathbb{N}$*, all* $S \subseteq [n] := \{0, \ldots, n{-}1\}$*, all* $a, i < n$ *we have*

$$([n], <, +, \times, S) \models bij_c(a, i) \iff |S| < (\log n)^c \text{ and } a \text{ is the } i\text{-th smallest element of } S.$$

- Using this, identify the non-isolated vertices with numbers $< (\log n)^c$ and represent them by bitstrings of length $c \log \log n$.
- Identify an arbitrary vertex of G with a number $< n$, whose binary representation encodes a sequence of $\ell(n) := \frac{\log n}{c \log \log n}$ non-isolated vertices.
- Use this to express that there is a path of length $\ell(n)$ from node x to node y.
- Iterate this for $c{+}1$ times to express that there is a path of length $\ell(n)^{c+1} \geqslant (\log n)^c$ from x to y. □

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

Theorem: *(Harwath, S., 2013)*

Let p be a prime power and let $k \in \mathbb{N}$ be coprime with p.

For every k-ary query q expressible in Arb-invariant FO+MOD$_p$, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-shift-local w.r.t. k.

Definition: Let q be a k-ary graph query. Let $f : \mathbb{N} \to \mathbb{N}$.

q is called $f(n)$-shift-local w.r.t. k if there is an n_0 such that for every $n \geqslant n_0$ and every graph G with $|V^G| = n$, the following is true for all k-tuples $(a_0, \ldots, a_{k-1})$ of nodes:

if the $f(n)$-neighborhoods of the a_i are disjoint and isomorphic,

then $(a_0, a_1, \ldots, a_{k-1}) \in q(G) \iff (a_1, \ldots, a_{k-1}, a_0) \in q(G)$.

Proof: Use Smolensky's result for AC$^0[p]$-circuits.

Corollary: Reachability is not definable in Arb-invariant FO+MOD$_p$ (for prime power p).

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

> ***Theorem:*** *(Harwath, S., 2013)*
>
> *Let p be a prime power and let $k \in \mathbb{N}$ be coprime with p.*
>
> *For every k-ary query q expressible in Arb-invariant FO+MOD$_p$, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-shift-local w.r.t. k.*

Definition: Let q be a k-ary graph query. Let $f : \mathbb{N} \to \mathbb{N}$.

q is called $f(n)$-shift-local w.r.t. k if there is an n_0 such that for every $n \geqslant n_0$ and every graph G with $|V^G| = n$, the following is true for all k-tuples $(a_0, \ldots, a_{k-1})$ of nodes:

if the $f(n)$-neighborhoods of the a_i are disjoint and isomorphic,

then $(a_0, a_1, \ldots, a_{k-1}) \in q(G) \iff (a_1, \ldots, a_{k-1}, a_0) \in q(G)$.

Proof: Use Smolensky's result for AC$^0[p]$-circuits.

Corollary: Reachability is not definable in Arb-invariant FO+MOD$_p$ (for prime power p).

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

Theorem: *(Harwath, S., 2013)*

Let p be a prime power and let $k \in \mathbb{N}$ be coprime with p.

For every k-ary query q expressible in Arb-invariant FO+MOD$_p$, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-shift-local w.r.t. k.

Definition: Let q be a k-ary graph query. Let $f : \mathbb{N} \to \mathbb{N}$.

q is called $f(n)$-shift-local w.r.t. k if there is an n_0 such that for every $n \geqslant n_0$ and every graph G with $|V^G| = n$, the following is true for all k-tuples $(a_0, \ldots, a_{k-1})$ of nodes:

if the $f(n)$-neighborhoods of the a_i are disjoint and isomorphic,

then $(a_0, a_1, \ldots, a_{k-1}) \in q(G) \iff (a_1, \ldots, a_{k-1}, a_0) \in q(G)$.

Proof: Use Smolensky's result for AC$^0[p]$-circuits.

Corollary: Reachability is not definable in Arb-invariant FO+MOD$_p$ (for prime power p).

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

Theorem: *(Harwath, S., 2013)*

Let p be a prime power and let $k \in \mathbb{N}$ be coprime with p.

For every k-ary query q expressible in Arb-invariant FO+MOD$_p$*, there is a $c \in \mathbb{N}$ such that q is $(\log n)^c$-shift-local w.r.t. k.*

Definition: Let q be a k-ary graph query. Let $f : \mathbb{N} \to \mathbb{N}$.

q is called $f(n)$-shift-local w.r.t. k if there is an n_0 such that for every $n \geqslant n_0$ and every graph G with $|V^G| = n$, the following is true for all k-tuples $(a_0, \ldots, a_{k-1})$ of nodes:

if *the $f(n)$-neighborhoods of the a_i are disjoint and isomorphic,*

then $(a_0, a_1, \ldots, a_{k-1}) \in q(G) \iff (a_1, \ldots, a_{k-1}, a_0) \in q(G)$.

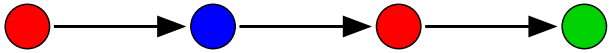
Proof: Use Smolensky's result for AC$^0[p]$-circuits.

Corollary: Reachability is not definable in Arb-invariant FO+MOD$_p$ (for prime power p).

Overview

Represent words as labeled graphs

(labeled) chain-graphs . . . . this chain-graph represents the string *rbrg*.



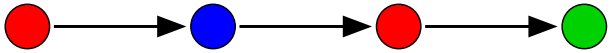
Edges correspond to the successor relation "*succ*" on the positions of the string.

Write $\prec$ -inv-FO(*succ*) for order-invariant FO on these graphs.

Write $+$-inv-FO(*succ*) for addition-invariant FO on these graphs.

Represent words as labeled graphs

(labeled) chain-graphs .... this chain-graph represents the string *rbrg*.



Edges correspond to the successor relation "*succ*" on the positions of the string.

Write $\prec$ -inv-FO(*succ*) for order-invariant FO on these graphs.
Write $+$-inv-FO(*succ*) for addition-invariant FO on these graphs.

Known results

$\mathrm{FO}(\mathit{succ}) = $ locally threshold testable languages (Thomas)

$\mathrm{FO}(<) = $ star-free regular languages (McNaughton, Papert)

$\mathrm{MSO}(<) = $ regular languages (Büchi, Elgot, Trakhtenbrot)

$\mathrm{MSO}(<) = \mathrm{MSO}(\mathit{succ}) = \mathrm{MLFP}(<) = \mathrm{MLFP}(\mathit{succ})$

$\prec$-inv-$\mathrm{FO}(\mathit{succ}) = \mathrm{FO}(\mathit{succ})$ (Benedikt, Segoufin, 2005)

$+$-inv-$\mathrm{MLFP}(\mathit{succ}) \supseteq$ DLIN (S., 2004)

$+$-inv-$\mathrm{MSO}(\mathit{succ}) = $ the linear time hierarchy (More & Olive 1997, S. 2004)

$\mathrm{FO}(<) = \prec\text{-inv-}\mathrm{FO}(<) \subsetneq +\text{-inv-}\mathrm{FO}(<) \subsetneq (+,\times)\text{-inv-}\mathrm{FO}(<) \subseteq$ uniform AC^0.

The following problem is undecidable: (Benedikt, Segoufin, 2005)

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:

Input: a $\mathrm{FO}(\prec, E)$-sentence φ

Question: Is φ order-invariant on all finite labeled chain-graphs?

Known results

$\mathrm{FO}(\mathit{succ})$ = locally threshold testable languages (Thomas)

$\mathrm{FO}(<)$ = star-free regular languages (McNaughton, Papert)

$\mathrm{MSO}(<)$ = regular languages (Büchi, Elgot, Trakhtenbrot)

$\mathrm{MSO}(<) = \mathrm{MSO}(\mathit{succ}) = \mathrm{MLFP}(<) = \mathrm{MLFP}(\mathit{succ})$

$\prec$-inv-$\mathrm{FO}(\mathit{succ}) = \mathrm{FO}(\mathit{succ})$ (Benedikt, Segoufin, 2005)

$+$-inv-$\mathrm{MLFP}(\mathit{succ}) \supseteq$ DLIN (S., 2004)

$+$-inv-$\mathrm{MSO}(\mathit{succ})$ = the linear time hierarchy (More & Olive 1997, S. 2004)

$\mathrm{FO}(<) = \prec\text{-inv-}\mathrm{FO}(<) \subsetneq +\text{-inv-}\mathrm{FO}(<) \subsetneq (+,\times)\text{-inv-}\mathrm{FO}(<) \subseteq$ uniform AC^0.

The following problem is undecidable: (Benedikt, Segoufin, 2005)

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:

Input: a $\mathrm{FO}(\prec, E)$-sentence φ

Question: Is φ order-invariant on all finite labeled chain-graphs?

Known results

FO(*succ*) = locally threshold testable languages (Thomas)

FO($<$) = star-free regular languages (McNaughton, Papert)

MSO($<$) = regular languages (Büchi, Elgot, Trakhtenbrot)

MSO($<$) = MSO(*succ*) = MLFP($<$) = MLFP(*succ*)

$\prec$-inv-FO(*succ*) = FO(*succ*) (Benedikt, Segoufin, 2005)

$+$-inv-MLFP(*succ*) $\supseteq$ DLIN (S., 2004)

$+$-inv-MSO(*succ*) = the linear time hierarchy (More & Olive 1997, S. 2004)

$$\text{FO}(<) \;=\; \prec\text{-inv-FO}(<) \;\subsetneq\; +\text{-inv-FO}(<) \;\subsetneq\; (+,\times)\text{-inv-FO}(<) \;\subseteq\; \text{uniform AC}^0.$$

The following problem is undecidable: (Benedikt, Segoufin, 2005)

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:

Input: a FO($\prec$, E)-sentence φ

Question: Is φ order-invariant on all finite labeled chain-graphs?

Known results

$\mathrm{FO}(\mathit{succ})$ = locally threshold testable languages (Thomas)

$\mathrm{FO}(<)$ = star-free regular languages (McNaughton, Papert)

$\mathrm{MSO}(<)$ = regular languages (Büchi, Elgot, Trakhtenbrot)

$\mathrm{MSO}(<) = \mathrm{MSO}(\mathit{succ}) = \mathrm{MLFP}(<) = \mathrm{MLFP}(\mathit{succ})$

$\prec$-inv-$\mathrm{FO}(\mathit{succ}) = \mathrm{FO}(\mathit{succ})$ (Benedikt, Segoufin, 2005)

$+$-inv-$\mathrm{MLFP}(\mathit{succ}) \supseteq$ DLIN (S., 2004)

$+$-inv-$\mathrm{MSO}(\mathit{succ})$ = the linear time hierarchy (More & Olive 1997, S. 2004)

$\mathrm{FO}(<) = \prec\text{-inv-}\mathrm{FO}(<) \subsetneq +\text{-inv-}\mathrm{FO}(<) \subsetneq (+,\times)\text{-inv-}\mathrm{FO}(<) \subseteq \text{uniform } \mathrm{AC}^0$.

The following problem is undecidable: (Benedikt, Segoufin, 2005)

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:

Input: a $\mathrm{FO}(\prec, E)$-sentence φ

Question: Is φ order-invariant on all finite labeled chain-graphs?

The “Algebraic” Approach

Let L_1 and L_2 be logics, and let C be a class of structures.

Goal: Show that L_1 can define exactly the same properties of C-structures as L_2.

Approach:

(0) Identify a suitable set of operations $\mathcal{O}$ on structures in C.

(1) Show that a property p of C-structures is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$. I.e., for every $\mathcal{A} \in C$:

$$\mathcal{A} \text{ has property } p \iff op(\mathcal{A}) \text{ has property } p.$$

(2) Show that a property p of C-structures is closed under every operation $op \in \mathcal{O}$ iff it is definable in L_2.

The "Algebraic" Approach

Let L_1 and L_2 be logics, and let C be a class of structures.

Goal: Show that L_1 can define exactly the same properties of C-structures as L_2.

Approach:

(0) Identify a suitable set of operations $\mathcal{O}$ on structures in C.

(1) Show that a property p of C-structures is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$. I.e., for every $\mathcal{A} \in C$:

$$\mathcal{A} \text{ has property } p \iff op(\mathcal{A}) \text{ has property } p.$$

(2) Show that a property p of C-structures is closed under every operation $op \in \mathcal{O}$ iff it is definable in L_2.

The "Algebraic" Approach

Let L_1 and L_2 be logics, and let C be a class of structures.

Goal: Show that L_1 can define exactly the same properties of C-structures as L_2.

Approach:

(0) Identify a suitable set of operations $\mathcal{O}$ on structures in C.

(1) Show that a property p of C-structures is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$. I.e., for every $\mathcal{A} \in C$:

$$\mathcal{A} \text{ has property } p \iff op(\mathcal{A}) \text{ has property } p.$$

(2) Show that a property p of C-structures is closed under every operation $op \in \mathcal{O}$ iff it is definable in L_2.

The "Algebraic" Approach

Let L_1 and L_2 be logics, and let C be a class of structures.

Goal: Show that L_1 can define exactly the same properties of C-structures as L_2.

Approach:

(0) Identify a suitable set of operations $\mathcal{O}$ on structures in C.

(1) Show that a property p of C-structures is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$. I.e., for every $\mathcal{A} \in C$:

$$\mathcal{A} \text{ has property } p \iff op(\mathcal{A}) \text{ has property } p.$$

(2) Show that a property p of C-structures is closed under every operation $op \in \mathcal{O}$ iff it is definable in L_2.

The "Algebraic" Approach

Let L_1 and L_2 be logics, and let C be a class of structures.

Goal: Show that L_1 can define exactly the same properties of C-structures as L_2.

Approach:

(0) Identify a suitable set of operations $\mathcal{O}$ on structures in C.

(1) Show that a property p of C-structures is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$. I.e., for every $\mathcal{A} \in C$:

$$\mathcal{A} \text{ has property } p \iff op(\mathcal{A}) \text{ has property } p.$$

(2) Show that a property p of C-structures is closed under every operation $op \in \mathcal{O}$ iff it is definable in L_2.

An example

Theorem (Benedikt, Segoufin, '09):
A string-language is definable in $\prec$-inv-FO(*succ*) iff it is definable in FO(*succ*).

Main ingredients of the proof:

- Use a result by Beauquier and Pin (1989) stating that a string-language is definable in FO(*succ*) iff it is aperiodic and closed under swaps.
  - A string language L is aperiodic iff there exists a number $\ell \in \mathbb{N}$ such that for all strings u, x, v we have

$$u\,x^{\ell}\,v \in L \iff u\,x^{\ell+1}\,v \in L.$$

  - L is closed under swaps iff for all strings u, v, e, x, y, z such that e, f are idempotents (i.e., for all u, v we have $uev \in L$ iff $ue^2v \in L$), we have

$$u\;e\,x\,f\;y\;e\,z\,f\;v \in L \iff u\;e\,z\,f\;y\;e\,x\,f\;v \in L.$$

- Show that every string-language definable in $\prec$-inv-FO(*succ*) is aperiodic and closed under swaps.

  (For this, you can use Ehrenfeucht-Fraïssé games.)

An example

Theorem (Benedikt, Segoufin, '09):
A string-language is definable in $\prec$-inv-FO(*succ*) iff it is definable in FO(*succ*).

Main ingredients of the proof:

- Use a result by Beauquier and Pin (1989) stating that a string-language is definable in FO(*succ*) iff it is aperiodic and closed under swaps.
  - A string language L is aperiodic iff there exists a number $\ell \in \mathbb{N}$ such that for all strings u, x, v we have
  $$u\, x^{\ell}\, v \in L \iff u\, x^{\ell+1}\, v \in L.$$
  - L is closed under swaps iff for all strings u, v, e, x, y, z such that e, f are idempotents (i.e., for all u, v we have $uev \in L$ iff $ue^2v \in L$), we have
  $$u\; e\,x\,f\; y\; e\,z\,f\; v \in L \iff u\; e\,z\,f\; y\; e\,x\,f\; v \in L.$$
- Show that every string-language definable in $\prec$-inv-FO(*succ*) is aperiodic and closed under swaps.

  (For this, you can use Ehrenfeucht-Fraïssé games.)

An example

Theorem (Benedikt, Segoufin, '09):
A string-language is definable in $\prec$-inv-FO(*succ*) iff it is definable in FO(*succ*).

Main ingredients of the proof:

- Use a result by Beauquier and Pin (1989) stating that a string-language is definable in FO(*succ*) iff it is aperiodic and closed under swaps.
  - A string language L is aperiodic iff there exists a number $\ell \in \mathbb{N}$ such that for all strings u, x, v we have

$$u\, x^{\ell}\, v \in L \quad \Longleftrightarrow \quad u\, x^{\ell+1}\, v \in L.$$

  - L is closed under swaps iff for all strings u, v, e, x, y, z such that e, f are idempotents (i.e., for all u, v we have $uev \in L$ iff $ue^2v \in L$), we have

$$u\; e\, x\, f\; y\; e\, z\, f\; v \;\in\; L \quad \Longleftrightarrow \quad u\; e\, z\, f\; y\; e\, x\, f\; v \;\in\; L.$$

- Show that every string-language definable in $\prec$-inv-FO(*succ*) is aperiodic and closed under swaps.

  (For this, you can use Ehrenfeucht-Fraïssé games.)

An example

Theorem (Benedikt, Segoufin, '09):
A string-language is definable in $\prec$-inv-FO(*succ*) iff it is definable in FO(*succ*).

Main ingredients of the proof:

- Use a result by Beauquier and Pin (1989) stating that a string-language is definable in FO(*succ*) iff it is aperiodic and closed under swaps.
  - A string language L is aperiodic iff there exists a number $\ell \in \mathbb{N}$ such that for all strings u, x, v we have
    $$u\, x^{\ell}\; v \in L \quad \Longleftrightarrow \quad u\, x^{\ell+1}\; v \in L.$$
  - L is closed under swaps iff for all strings u, v, e, x, y, z such that e, f are idempotents (i.e., for all u, v we have $uev \in L$ iff $ue^2v \in L$), we have
    $$u\; e\, x\, f\; y\; e\, z\, f\; v \;\in\; L \quad \Longleftrightarrow \quad u\; e\, z\, f\; y\; e\, x\, f\; v \;\in\; L.$$
- Show that every string-language definable in $\prec$-inv-FO(*succ*) is aperiodic and closed under swaps.

  (For this, you can use Ehrenfeucht-Fraïssé games.)

An example

Theorem (Benedikt, Segoufin, '09):
A string-language is definable in $\prec$-inv-FO(*succ*) iff it is definable in FO(*succ*).

Main ingredients of the proof:

- Use a result by Beauquier and Pin (1989) stating that a string-language is definable in FO(*succ*) iff it is aperiodic and closed under swaps.
  - A string language L is aperiodic iff there exists a number $\ell \in \mathbb{N}$ such that for all strings u, x, v we have
  $$u\, x^{\ell}\, v \in L \iff u\, x^{\ell+1}\, v \in L.$$
  - L is closed under swaps iff for all strings u, v, e, x, y, z such that e, f are idempotents (i.e., for all u, v we have $uev \in L$ iff $ue^2v \in L$), we have
  $$u\; e\,x\,f\; y\; e\,z\,f\; v \;\in\; L \iff u\; e\,z\,f\; y\; e\,x\,f\; v \;\in\; L.$$
- Show that every string-language definable in $\prec$-inv-FO(*succ*) is aperiodic and closed under swaps.

  (For this, you can use Ehrenfeucht-Fraïssé games.)

Some further results proved using this method

Theorem:

- A tree-language is definable in $<$-invariant FO(*Succ*) iff it is definable in FO(*Succ*). (Benedikt, Segoufin '09)
  (They use aperiodicity and closure under guarded swaps.)

- A colored finite set is definable in $+$-invariant FO iff it is definable in $\mathrm{FO}_{\mathrm{card}}$ (i.e., FO with predicates testing the cardinality of the universe modulo fixed numbers). (S., Segoufin '10)

- A regular string- or tree-language is definable in $+$-invariant FO(*Succ*) iff it is definable in $\mathrm{FO}_{\mathrm{card}}(succ)$. (S., Segoufin '10 and Harwath, S. '12)
  (They use closure under transfers and closure under guarded swaps.)

- A regular string-language is definable in Arb-invariant FO(*Succ*) iff it is definable in $\mathrm{FO}_{\mathrm{card}}(Succ)$. (Anderson, van Melkebeek, S., Segoufin '11)

Some further results proved using this method

Theorem:

- A tree-language is definable in $<$-invariant FO(*Succ*) iff it is definable in FO(*Succ*). (Benedikt, Segoufin '09)
  (They use aperiodicity and closure under guarded swaps.)

- A colored finite set is definable in $+$-invariant FO iff it is definable in FO_{card} (i.e., FO with predicates testing the cardinality of the universe modulo fixed numbers). (S., Segoufin '10)

- A regular string- or tree-language is definable in $+$-invariant FO(*Succ*) iff it is definable in $FO_{card}(succ)$. (S., Segoufin '10 and Harwath, S. '12)
  (They use closure under transfers and closure under guarded swaps.)

- A regular string-language is definable in Arb-invariant FO(*Succ*) iff it is definable in $FO_{card}(Succ)$. (Anderson, van Melkebeek, S., Segoufin '11)

Some further results proved using this method

Theorem:

- A tree-language is definable in $<$-invariant FO(*Succ*) iff it is definable in FO(*Succ*). (Benedikt, Segoufin '09)
  (They use aperiodicity and closure under guarded swaps.)

- A colored finite set is definable in $+$-invariant FO iff it is definable in FO_{card} (i.e., FO with predicates testing the cardinality of the universe modulo fixed numbers). (S., Segoufin '10)

- A regular string- or tree-language is definable in $+$-invariant FO(*Succ*) iff it is definable in $FO_{card}(succ)$. (S., Segoufin '10 and Harwath, S. '12)
  (They use closure under transfers and closure under guarded swaps.)

- A regular string-language is definable in Arb-invariant FO(*Succ*) iff it is definable in $FO_{card}(Succ)$. (Anderson, van Melkebeek, S., Segoufin '11)

Some further results proved using this method

Theorem:

- A tree-language is definable in $<$-invariant FO(*Succ*) iff it is definable in FO(*Succ*). (Benedikt, Segoufin '09)
(They use aperiodicity and closure under guarded swaps.)

- A colored finite set is definable in $+$-invariant FO iff it is definable in FO_{card} (i.e., FO with predicates testing the cardinality of the universe modulo fixed numbers). (S., Segoufin '10)

- A regular string- or tree-language is definable in $+$-invariant FO(*Succ*) iff it is definable in $FO_{card}(succ)$. (S., Segoufin '10 and Harwath, S. '12)
(They use closure under transfers and closure under guarded swaps.)

- A regular string-language is definable in Arb-invariant FO(*Succ*) iff it is definable in $FO_{card}(Succ)$. (Anderson, van Melkebeek, S., Segoufin '11)

An open question

Open Question:

Are all languages definable in addition-invariant FO regular?

Known: (S., Segoufin, 2010)

- Arb-invariant FO can define non-regular languages, e.g.,
  $L = \{w \in \{1\}^* : |w| \text{ is a prime number }\}$.
- Every deterministic context-free language definable in addition-invariant FO is regular.
- Every commutative language definable in addition-invariant FO is regular.
- Every bounded language definable in addition-invariant FO is regular.

Bounded languages

Definition: (Ginsburg & Spanier, 1964)

$L \subseteq \Sigma^*$ is bounded $\iff$

$\exists\, k \in \mathbb{N}$ and k strings $w_1, \ldots, w_k \in \Sigma^*$ such that $L \subseteq w_1^* w_2^* \cdots w_k^*$.

Theorem: *(S., Segoufin, 2010)*

Every bounded language definable in $+$-inv-FO($<$) is regular.

Proof method:

- Identify $w_1^* w_2^* \cdots w_k^*$ with $\mathbb{N}^k$ via $(x_1, \ldots, x_k) \in \mathbb{N}^k \;\widehat{=}\; w_1^{x_1} w_2^{x_2} \cdots w_k^{x_k}$.
  Thus: $L \subseteq w_1^* w_2^* \cdots w_k^* \;\widehat{=}\; S(L) \subseteq \mathbb{N}^k$.
- Note that $S(L)$ is semi-linear, since L is definable in $+$-inv-FO($<$).
- Reason about semi-linear sets …

Corollary:

Every commutative language definable in $+$-inv-FO($<$) is regular.

Bounded languages

Definition: (Ginsburg & Spanier, 1964)

$L \subseteq \Sigma^*$ is bounded $\iff$

$\exists\, k \in \mathbb{N}$ and k strings $w_1, \ldots, w_k \in \Sigma^*$ such that $L \subseteq w_1^* w_2^* \cdots w_k^*$.

Theorem: *(S., Segoufin, 2010)*

Every bounded language definable in $+$-*inv*-FO$(<)$ *is regular.*

Proof method:

- Identify $w_1^* w_2^* \cdots w_k^*$ with $\mathbb{N}^k$ via $(x_1, \ldots, x_k) \in \mathbb{N}^k \;\widehat{=}\; w_1^{x_1} w_2^{x_2} \cdots w_k^{x_k}$.
  Thus: $L \subseteq w_1^* w_2^* \cdots w_k^* \;\widehat{=}\; S(L) \subseteq \mathbb{N}^k$.
- Note that $S(L)$ is semi-linear, since L is definable in $+$-inv-FO$(<)$.
- Reason about semi-linear sets ...

Corollary:

Every commutative language definable in $+$-*inv*-FO$(<)$ *is regular.*

Bounded languages

Definition: (Ginsburg & Spanier, 1964)

$L \subseteq \Sigma^*$ is bounded $\iff$

$\exists\, k \in \mathbb{N}$ and k strings $w_1, \ldots, w_k \in \Sigma^*$ such that $L \subseteq w_1^* w_2^* \cdots w_k^*$.

Theorem: *(S., Segoufin, 2010)*

Every bounded language definable in $+$-inv-FO($<$) is regular.

Proof method:

- Identify $w_1^* w_2^* \cdots w_k^*$ with $\mathbb{N}^k$ via $(x_1, \ldots, x_k) \in \mathbb{N}^k \;\widehat{=}\; w_1^{x_1} w_2^{x_2} \cdots w_k^{x_k}$.
  Thus: $L \subseteq w_1^* w_2^* \cdots w_k^* \;\widehat{=}\; S(L) \subseteq \mathbb{N}^k$.
- Note that $S(L)$ is semi-linear, since L is definable in $+$-inv-FO($<$).
- Reason about semi-linear sets ...

Corollary:

Every commutative language definable in $+$-inv-FO($<$) is regular.

Characterization of colored sets definable in $+$-inv-FO

Definition: A colored finite set is a finite relational structure over a finite signature that contains only unary relation symbols.

> *Theorem:* *(S., Segoufin, 2010)*
>
> *Over the class of colored finite sets, $+$-inv-FO$(=)$ and $\mathrm{FO}_{\mathrm{Card}}(=)$ have the same expressive power.*

Proof:

- Every $+$-inv-FO$(=)$ sentence over colored sets defines a commutative language.
- Every commutative language definable in $+$-inv-FO$(<)$ is regular.
- Every regular language definable in $+$-inv-FO$(=)$ is definable in $\mathrm{FO}_{\mathrm{Card}}(=)$.

□

Characterization of colored sets definable in $+$-inv-FO

Definition: A colored finite set is a finite relational structure over a finite signature that contains only unary relation symbols.

Theorem: *(S., Segoufin, 2010)*

Over the class of colored finite sets, $+$-inv-FO$(=)$ and $\mathrm{FO}_{\mathrm{Card}}(=)$ have the same expressive power.

Note: $\mathrm{FO}_{\mathrm{Card}}(=)$ is a logic (with a decidable syntax); $+$-inv-FO$(=)$ is not.

More precisely: The following problem is undecidable:

Input: a FO$(\prec, +, C)$-sentence φ (C a unary relation symbol)

Question: Is φ addition-invariant on all finite $\{C\}$-structures ?

Overview

Gaifman-locality

If $\mathcal{N}_r^G(a) \cong \mathcal{N}_r^G(b)$ then $\big(a \in q(G) \iff b \in q(G)\big)$.

Known:

- Queries definable in order-invariant FO are Gaifman-local with respect to a constant locality radius. (Grohe, Schwentick '98)
- Queries definable in Arb-invariant FO are Gaifman-local with respect to a poly-logarithmic locality radius. (Anderson, Melkebeek, S., Segoufin '11)

Open Question:

- How about addition-invariant FO — is it Gaifman-local with respect to a constant locality radius?

Hanf-locality

A graph property p is Hanf-local w.r.t. locality radius r, if
any two graphs having the same r-neighbourhood types with the same multiplicities,
are not distinguished by p.

Known:

- Properties of graphs definable in FO are Hanf-local w.r.t. a constant locality radius. (Fagin, Stockmeyer, Vardi '95)
- Properties of strings or trees definable by order-invariant FO are Hanf-local w.r.t. a constant locality radius. (Benedikt, Segoufin '09)
- Properties of strings definable by Arb-invariant FO are Hanf-local w.r.t. a poly-logarithmic locality radius. (Anderson, van Melkebeek, S., Segoufin '11)
- Properties of strings definable by Arb-invariant FO+MOD$_p$, for odd prime powers p, are Hanf-local w.r.t. a poly-logarithmic locality radius. (Harwath, S. '13)

Open Question:

- Which of these results generalise from strings to arbitrary finite graphs?

Decidable Characterisations

Open Question:

Are there decidable characterisations of

- order-invariant FO?
- addition-invariant FO?
- $(+, \times)$-invariant FO?

Known:

- On finite strings and trees: order-invariant FO $\equiv$ FO. (Benedikt, Segoufin '10)
- On finite coloured sets: addition-invariant FO $\equiv$ FO enriched by "cardinality modulo" quantifiers. (S., Segoufin '10)

Thank You!