A tutorial on order-invariant logics

Nicole Schweikardt

Humboldt-Universität zu Berlin

In this tutorial

- Consider finite directed graphs $G = (V^G, E^G)$.

  Sometimes, nodes are additionally colored by a symbol from a finite alphabet Σ.

In this tutorial

- Consider finite directed graphs $G = (V^G, E^G)$.

  Sometimes, nodes are additionally colored by a symbol from a finite alphabet Σ.

- p is a graph property, if the following is true for all graphs G and H:

  if $G \cong H$, then G has property $p \iff H$ has property p

In this tutorial

- Consider finite directed graphs $G = (V^G, E^G)$.

  Sometimes, nodes are additionally colored by a symbol from a finite alphabet Σ.

- p is a graph property, if the following is true for all graphs G and H:

  if $G \cong H$, then $\quad G$ has property $p \iff H$ has property p

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

  if $\pi : G \cong H$, then for all $a_1, \ldots, a_k \in V^G$,

  $$(a_1, \ldots, a_k) \in q(G) \iff (\pi(a_1), \ldots, \pi(a_k)) \in q(H)$$

In this tutorial

- Consider finite directed graphs $G = (V^G, E^G)$.

  Sometimes, nodes are additionally colored by a symbol from a finite alphabet Σ.

- p is a graph property, if the following is true for all graphs G and H:

  if $G \cong H$, then G has property $p \iff H$ has property p

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

  if $\pi : G \cong H$, then for all $a_1, \dots, a_k \in V^G$,

  $$(a_1, \dots, a_k) \in q(G) \iff (\pi(a_1), \dots, \pi(a_k)) \in q(H)$$

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

Logics expressing graph-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 graph-properties and queries in a straightforward way.

Example: The query

$$q(G) \;=\; \{\, x \in V^G \;:\; x \text{ lies on a triangle} \,\}$$

is expressed in FO via

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

Logics expressing graph-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 graph-properties and queries in a straightforward way.

Example: The query

$$q(G) \;=\; \{\, x \in V^G \;:\; x \text{ lies on a triangle} \,\}$$

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 V^G or E^G.
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 $<$, $+$, $\times$, ..., *Halt*, ... on V^G.
- For this, identify V^G with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |V^G|$ and interpret $<$, $+$, $\times$, ..., *Halt*, ... in the natural way.

Invariant logics

Idea:

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

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 V^G.
- For this, identify V^G with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |V^G|$ 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 V^G with $[n]$.
  These formulas are called Arb-invariant.

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

A $\mathcal{L}(E, <)$-formula $\varphi(\vec{x})$ is order-invariant on $G = (V^G, E^G)$ $\iff$
for all tuples of nodes $\vec{a}$ in V^G, for all linear orders $<_1$ and $<_2$ on V^G,

$$(G, <_1) \models \varphi(\vec{a}) \quad \iff \quad (G, <_2) \models \varphi(\vec{a}).$$

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 V^G.
- For this, identify V^G with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |V^G|$ 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 V^G with $[n]$.
These formulas are called Arb-invariant.

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

A $\mathcal{L}(E, <, +)$-formula $\varphi(\vec{x})$ is addition-invariant on $G = (V^G, E^G)$ $\iff$ for all tuples of nodes $\vec{a}$ in V^G, for all linear orders $<_1$ and $<_2$ on V^G, and the matching addition relations $+_1$, $+_2$,

$$(G, <_1, +_1) \models \varphi(\vec{a}) \quad \iff \quad (G, <_2, +_2) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, \mathit{Halt}, \ldots$ on V^G.
- For this, identify V^G with the set $[n] := \{0, 1, \ldots, n-1\}$ for $n = |V^G|$ 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 V^G with $[n]$.
These formulas are called Arb-invariant.

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

A $\mathcal{L}(E, <, +, \times)$-formula $\varphi(\vec{x})$ is $(+, \times)$-invariant on $G = (V^G, E^G)$ $\iff$ for all tuples of nodes $\vec{a}$ in V^G, for all linear orders $<_1$ and $<_2$ on V^G, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$,

$$(G, <_1, +_1, \times_1) \models \varphi(\vec{a}) \quad \iff \quad (G, <_2, +_2, \times_2) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

- Extend the expressive power of a logic by allowing formulas to also use arithmetic predicates like $<, +, \times, \ldots, \mathit{Halt}, \ldots$ on V^G.
- For this, identify V^G with the set $[n] := \{0, 1, \ldots, n{-}1\}$ for $n = |V^G|$ 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 V^G with $[n]$.
  These formulas are called Arb-invariant.

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

A $\mathcal{L}(E, <, +, \times, \ldots)$-formula $\varphi(\vec{x})$ is Arb-invariant on $G = (V^G, E^G)$ $\iff$ for all tuples of nodes $\vec{a}$ in V^G, for all linear orders $<_1$ and $<_2$ on V^G, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$, and other numerical predicates,

$$(G, <_1, +_1, \times_1, \ldots) \models \varphi(\vec{a}) \quad \iff \quad (G, <_2, +_2, \times_2, \ldots) \models \varphi(\vec{a}).$$

Invariant logics

Idea:

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

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

A $\mathcal{L}(E, <, +, \times, \dots)$-formula $\varphi(\vec{x})$ is Arb-invariant on $G = (V^G, E^G)$ $\iff$ for all tuples of nodes $\vec{a}$ in V^G, for all linear orders $<_1$ and $<_2$ on V^G, and the matching addition relations $+_1, +_2$, and the according multiplications $\times_1, \times_2$, and other numerical predicates,

$$(G, <_1, +_1, \times_1, \dots) \models \varphi(\vec{a}) \quad \iff \quad (G, <_2, +_2, \times_2, \dots) \models \varphi(\vec{a}).$$

For Arb-invariant sentences, shortly write $G \models \varphi$ for $(G, <_1, +_1, \times_1 \dots) \models \varphi$.

Example

- An addition-invariant FO$(E, <, +)$-sentence φ such that

$$G \models \varphi \quad \Longleftrightarrow \quad |V^G| \text{ is odd.}$$

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

Example

- An addition-invariant FO$(E,<,+)$-sentence φ such that

$$G \models \varphi \iff |V^G| \text{ is odd}.$$

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

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

$$G \models \psi \iff |V^G|-1 \text{ is a prime number}.$$

Example

- An addition-invariant FO$(E, <, +)$-sentence φ such that

$$G \models \varphi \quad \iff \quad |V^G| \text{ is odd.}$$

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

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

$$G \models \psi \quad \iff \quad |V^G| - 1 \text{ is a prime number.}$$

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

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

Example

- An addition-invariant $\mathrm{FO}(E,<,+)$-sentence φ such that

$$G \models \varphi \iff |V^G| \text{ is odd}.$$

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

- Similarly, there is an $(+,\times)$-invariant $\mathrm{FO}(E,<,+,\times)$-sentence ψ such that

$$G \models \psi \iff |V^G|-1 \text{ is a prime number}.$$

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

$G \models \chi \iff |V^G|-1$ is the index of a Turing machine halting on empty input:

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

Example

- An addition-invariant $\mathrm{FO}(E,<,+)$-sentence φ such that

$$G \models \varphi \iff |V^G| \text{ is odd.}$$

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

- Similarly, there is an $(+,\times)$-invariant $\mathrm{FO}(E,<,+,\times)$-sentence ψ such that

$$G \models \psi \iff |V^G|-1 \text{ is a prime number.}$$

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

  $G \models \chi \iff |V^G|-1$ is the index of a Turing machine halting on empty input:

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

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 graph-properties and queries. (Immerman, Vardi, 1982)

Expressive power of invariant logics

Known results:

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

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

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable graph-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the graph-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 graph-properties and queries that belong to the circuit complexity class AC^0. (Immerman, 1980s)

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth, using NOT-gates and AND- and OR-gates of unbounded fan-in.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable graph-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the graph-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 graph-properties and queries that belong to the circuit complexity class AC^0. (Immerman, 1980s)

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth, using NOT-gates and AND- and OR-gates of unbounded fan-in.
- Arb-invariant FO+MOD precisely captures the graph-properties and queries that belong to the circuit complexity class ACC.

  ACC consists of all problems solvable by circuit families of polynomial size and constant depth, using also MOD-gates of unbounded fan-in.

Expressive power of invariant logics

Known results:

- Order-invariant LFP precisely captures the polynomial time computable graph-properties and queries. (Immerman, Vardi, 1982)
- Arb-invariant LFP precisely captures the graph-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 graph-properties and queries that belong to the circuit complexity class AC^0. (Immerman, 1980s)

  AC^0 consists of all problems solvable by circuit families of polynomial size and constant depth, using NOT-gates and AND- and OR-gates of unbounded fan-in.
- Arb-invariant FO$+$MOD precisely captures the graph-properties and queries that belong to the circuit complexity class ACC.

  ACC consists of all problems solvable by circuit families of polynomial size and constant depth, using also MOD-gates of unbounded fan-in.
- $(+, \times)$-invariant FO precisely captures the graph-properties and queries that belong to uniform AC^0.

Overview

Order-invariance is undecidable

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

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 COLORED GRAPHS:
>
> *Input:* a FO($E, C, <$)-sentence φ
>
> *Question:* Is φ order-invariant on all finite colored graphs?

Proof: By a reduction using Trakhtenbrot's theorem.

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 COLORED GRAPHS:

Input: a $\text{FO}(E, C, <)$-sentence φ

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

Proof: By a reduction using Trakhtenbrot's theorem.

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

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 COLORED GRAPHS:

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

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

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given $\mathrm{FO}(E)$-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".

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 COLORED GRAPHS:
>
> *Input:* a $\mathrm{FO}(E, C, <)$-sentence φ
>
> *Question:* Is φ order-invariant on all finite colored graphs?

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given $\mathrm{FO}(E)$-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).

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 COLORED GRAPHS:
Input: a FO($E, C, <$)-sentence φ
Question: Is φ order-invariant on all finite colored graphs?

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 $\geqslant 2$.

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 COLORED GRAPHS:
>
> *Input:* a $\mathrm{FO}(E, C, <)$-sentence φ
>
> *Question:* Is φ order-invariant on all finite colored graphs?

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given $\mathrm{FO}(E)$-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., $\chi := \exists x\big(C(x) \wedge \forall y(x \leqslant y)\big)$.

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 COLORED GRAPHS:

Input: a $\text{FO}(E, C, <)$-sentence φ

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

Proof: By a reduction using Trakhtenbrot's theorem.

- Assume, for contradiction, that order-invariance is decidable.
- Then, also the problem "Is a given $\text{FO}(E)$-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., $\chi := \exists x\big(C(x) \wedge \forall y(x \leqslant y)\big)$.
    Stop with output "yes" iff the formula $(\neg\psi \rightarrow \chi)$ is order-invariant. $\square$

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):
Exercise: Get rid of C!

ORDER-INVARIANCE ON FINITE COLORED GRAPHS:
Input: a FO($E, C, <$)-sentence φ
Question: Is φ order-invariant on all finite colored graphs?

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., $\chi := \exists x\big(C(x) \wedge \forall y(x \leqslant y)\big)$.
    Stop with output "yes" iff the formula $(\neg\psi \ \rightarrow \chi)$ is order-invariant. □

Order-invariance for unary signatures is decidable

Let $\{C_1, \ldots, C_\ell\}$ consist of unary relation symbols (i.e., node colors).

Theorem: Order-invariance of a given $\mathrm{FO}(C_1, \ldots, C_\ell, <)$-sentence φ (on the class of all finite colored sets) is decidable.

Order-invariance for unary signatures is decidable

Let $\{C_1, \ldots, C_\ell\}$ consist of unary relation symbols (i.e., node colors).

Theorem: Order-invariance of a given FO$(C_1, \ldots, C_\ell, <)$-sentence φ (on the class of all finite colored sets) is decidable.

Decision procedure:

- φ defines a language L of finite strings.

Order-invariance for unary signatures is decidable

Let $\{C_1, \dots, C_\ell\}$ consist of unary relation symbols (i.e., node colors).

> *Theorem:* Order-invariance of a given FO$(C_1, \dots, C_\ell, <)$-sentence φ (on the class of all finite colored sets) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.

  Definition: L is commutative iff
  for all $w = a_1 \cdots a_n$ and all permutations $\pi \in S_n$
  we have $w \in L \iff a_{\pi(1)} \cdots a_{\pi(n)} \in L$.

Order-invariance for unary signatures is decidable

Let $\{C_1, \dots, C_\ell\}$ consist of unary relation symbols (i.e., node colors).

Theorem: Order-invariance of a given FO($C_1, \dots, C_\ell, <$)-sentence φ (on the class of all finite colored sets) is decidable.

Decision procedure:

- φ defines a language L of finite strings.
- φ is order-invariant $\iff$ L is commutative.

  Definition: L is commutative iff
  for all $w = a_1 \cdots a_n$ and all permutations $\pi \in S_n$
  we have $w \in L \iff a_{\pi(1)} \cdots a_{\pi(n)} \in L$.
- Commutativity of regular string-languages is decidable.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C, <, +$)-sentence φ (on the class of all finite colored sets) is not decidable.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C, <, +$)-sentence φ (on the class of all finite colored sets) is not decidable.

Proof:

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO($C, <, +$)-sentence $\tilde{\psi}_n$ such that $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C, <, +$)-sentence φ (on the class of all finite colored sets) is not decidable.

Proof:

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO($C, <, +$)-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C, <, +$)-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0, 1, \ldots, N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO($C, <, +$)-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO($<, +, \times$)-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:

$$n \in H \iff \text{there is an } N > n \text{ such that } ([N], <, +, \times) \models \psi(n).$$

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO$(C, <, +)$-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0, 1, \ldots, N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO$(C, <, +)$-sentence $\tilde{\psi}_n$ such that $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO$(<, +, \times)$-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:

$$n \in H \iff \text{there is an } N > n \text{ such that } ([N], <, +, \times) \models \psi(n).$$

- Let C be the set of all square numbers. Known:

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO$(C,<,+)$-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0,1,\dots,N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO$(C,<,+)$-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO$(<,+,\times)$-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:

$$n \in H \iff \text{there is an } N > n \text{ such that } ([N],<,+,\times) \models \psi(n).$$

- Let C be the set of all square numbers. Known:
  - $\times$ is FO$(<,+,C)$-definable in $([N],<,+,C)$ via $\varphi_\times(x,y,z)$.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C, <, +$)-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0, 1, \ldots, N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO($C, <, +$)-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO($<, +, \times$)-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:
  $$n \in H \iff \text{there is an } N > n \text{ such that } ([N], <, +, \times) \models \psi(n).$$
- Let C be the set of all square numbers. Known:
  - $\times$ is FO($<, +, C$)-definable in $([N], <, +, C)$ via $\varphi_\times(x, y, z)$.
  - "C = *square numbers*" is FO($<, +, C$)-definable via $\varphi_{C=squares}$.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO($C,<,+$)-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0, 1, \ldots, N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO($C,<,+$)-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO($<,+,\times$)-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:
  $$n \in H \iff \text{there is an } N > n \text{ such that } ([N],<,+,\times) \models \psi(n).$$
- Let C be the set of all square numbers. Known:
  - $\times$ is FO($<,+,C$)-definable in $([N],<,+,C)$ via $\varphi_\times(x,y,z)$.
  - "$C = \textit{square numbers}$" is FO($<,+,C$)-definable via $\varphi_{C=squares}$.
- Substitute $\varphi_\times$ for $\times$ in $\psi(z)$ and let, for each $n \in \mathbb{N}$, $\tilde{\psi}_n := \psi(n) \wedge \varphi_{C=squares}$.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO$(C,<,+)$-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0,1,\ldots,N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO$(C,<,+)$-sentence $\tilde{\psi}_n$ such that
  $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO$(<,+,\times)$-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:

$$n \in H \iff \text{there is an } N > n \text{ such that } ([N],<,+,\times) \models \psi(n).$$

- Let C be the set of all square numbers. Known:
  - $\times$ is FO$(<,+,C)$-definable in $([N],<,+,C)$ via $\varphi_\times(x,y,z)$.
  - "$C = \textit{square numbers}$" is FO$(<,+,C)$-definable via $\varphi_{C=squares}$.
- Substitute $\varphi_\times$ for $\times$ in $\psi(z)$ and let, for each $n \in \mathbb{N}$, $\tilde{\psi}_n := \psi(n) \wedge \varphi_{C=squares}$.
  Note: $\tilde{\psi}_n$ is an FO$(<,+,C)$-sentence.

Addition-invariance for unary signatures is undecidable

Theorem: Addition-invariance of a given FO$(C,<,+)$-sentence φ (on the class of all finite colored sets) is not decidable.

Proof: Notation: $[N] := \{0,1,\ldots,N\}$

- Let $H \subseteq \mathbb{N}$ be recursively enumerable, but not decidable.
- **Goal:** Construct, for each $n \in \mathbb{N}$, a FO$(C,<,+)$-sentence $\tilde{\psi}_n$ such that $n \notin H \iff \tilde{\psi}_n$ is addition-invariant.
- Since H is r.e., it is FO-definable in bounded arithmetic. Thus, there is a FO$(<,+,\times)$-formula $\psi(z)$ such that for all $n \in \mathbb{N}$ we have:

$$n \in H \iff \text{there is an } N > n \text{ such that } ([N],<,+,\times) \models \psi(n).$$

- Let C be the set of all square numbers. Known:
  - $\times$ is FO$(<,+,C)$-definable in $([N],<,+,C)$ via $\varphi_\times(x,y,z)$.
  - "$C = \textit{square numbers}$" is FO$(<,+,C)$-definable via $\varphi_{C=squares}$.
- Substitute $\varphi_\times$ for $\times$ in $\psi(z)$ and let, for each $n \in \mathbb{N}$, $\tilde{\psi}_n := \psi(n) \wedge \varphi_{C=squares}$.
  Note: $\tilde{\psi}_n$ is an FO$(<,+,C)$-sentence.
- Furthermore, $n \notin H \iff \tilde{\psi}_n$ is addition-invariant. □

Overview

FO $<$ order-invariant FO — Gurevich's proof

Theorem (Gurevich):

For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

FO $<$ order-invariant FO — Gurevich's proof

Theorem (Gurevich):

For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-inv. FO($\subseteq, <$)-sentence φ_{even} such that for every finite set X:

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

FO $<$ order-invariant FO — Gurevich's proof

Theorem (Gurevich):

For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-inv. FO($\subseteq, <$)-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($\subseteq$)-sentence ψ_{even} such that for every finite set X:

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

FO $<$ order-invariant FO — Gurevich's proof

Theorem (Gurevich):

For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-inv. FO($\subseteq, <$)-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($\subseteq$)-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. $<$) atom of X, every other (w.r.t. $<$) atom of X, but not the last (w.r.t. $<$) atom of X.

FO $<$ order-invariant FO — Gurevich's proof

Theorem (Gurevich):

For a finite set X let $\mathcal{B}_X := (2^X, \subseteq)$ be the Boolean algebra over X.

There is an order-inv. FO($\subseteq, <$)-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($\subseteq$)-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. $<$) atom of X, every other (w.r.t. $<$) atom of X, but not the last (w.r.t. $<$) 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 — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

FO $<$ order-invariant FO — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

There is an order-inv. FO$(E, D, <)$-sentence φ_{even} such that
for every finite **full** binary tree T we have:

$$\mathcal{A}_T \models \varphi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

FO $<$ order-invariant FO — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

There is an order-inv. FO$(E, D, <)$-sentence φ_{even} such that
for every finite **full** binary tree T we have:

$$\mathcal{A}_T \models \varphi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

But there is no FO(E, D)-sentence ψ_{even} such that for every full binary tree T:

$$\mathcal{A}_T \models \psi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

FO $<$ order-invariant FO — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

There is an order-inv. FO$(E, D, <)$-sentence φ_{even} such that
for every finite **full** binary tree T we have:

$$\mathcal{A}_T \models \varphi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

But there is no FO(E, D)-sentence ψ_{even} such that for every full binary tree T:

$$\mathcal{A}_T \models \psi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

Proof:

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that $\mathcal{A}_T \equiv_r \mathcal{A}_{T'}$ for all unordered full binary trees of height $\geqslant 2^{r+1}$.

FO $<$ order-invariant FO — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

There is an order-inv. FO$(E, D, <)$-sentence φ_{even} such that for every finite **full** binary tree T we have:

$$\mathcal{A}_T \models \varphi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

But there is no FO(E, D)-sentence ψ_{even} such that for every full binary tree T:

$$\mathcal{A}_T \models \psi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

Proof:

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that $\mathcal{A}_T \equiv_r \mathcal{A}_{T'}$ for all unordered full binary trees of height $\geqslant 2^{r+1}$.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

FO $<$ order-invariant FO — Potthoff's proof

Theorem (Potthoff):

Represent a finite unordered binary tree T by a $\{E, D\}$-structure $\mathcal{A}_T$ where $(A, E^{\mathcal{A}}) = T$ and $D^{\mathcal{A}}$ is the transitive closure of $E^{\mathcal{A}}$ (i.e., the descendant-relation).

There is an order-inv. FO$(E, D, <)$-sentence φ_{even} such that for every finite **full** binary tree T we have:

$$\mathcal{A}_T \models \varphi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

But there is no FO(E, D)-sentence ψ_{even} such that for every full binary tree T:

$$\mathcal{A}_T \models \psi_{even} \iff \text{every leaf of } T \text{ has even height.}$$

Proof:

Part 2: Use an Ehrenfeucht-Fraïssé game argument to show that $\mathcal{A}_T \equiv_r \mathcal{A}_{T'}$ for all unordered full binary trees of height $\geqslant 2^{r+1}$.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

Exercise: Work out the details for the general case!

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T:

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.
- Consider the zig-zag-path which starts in the root, visits the root's 1-child, that node's 2-child, that node's 1-child, etc.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.
- Consider the zig-zag-path which starts in the root, visits the root's 1-child, that node's 2-child, that node's 1-child, etc.
- A **full** binary tree T has even height iff the last node on the zig-zag-path is a 2-child.

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.
- Consider the zig-zag-path which starts in the root, visits the root's 1-child, that node's 2-child, that node's 1-child, etc.
- A **full** binary tree T has even height iff the last node on the zig-zag-path is a 2-child. This can be expressed in order-inv. FO$(E, D, <)$ which states that
  - exists x_0, x_h such that
  - root(x_0), leaf(x_h), x_h is a 2-child,

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.
- Consider the zig-zag-path which starts in the root, visits the root's 1-child, that node's 2-child, that node's 1-child, etc.
- A **full** binary tree T has even height iff the last node on the zig-zag-path is a 2-child. This can be expressed in order-inv. FO($E, D, <$) which states that
  - exists x_0, x_h such that
  - root(x_0), leaf(x_h), x_h is a 2-child,
  - the node x_1 satisfying $E(x_0, x_1) \wedge D(x_1, x_h)$ is a 1-child

Part 1: For simplicity, we here only construct φ_{even} that is order-invariant on $\mathcal{A}_T$ for **full** binary trees T.

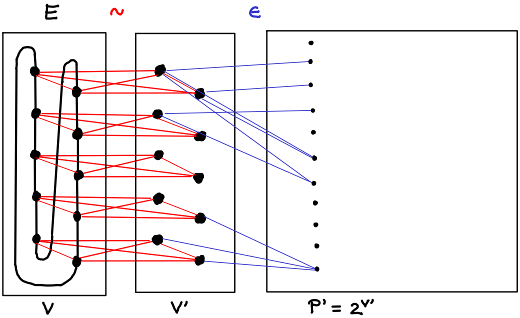
- Use $<$ to order the children of each node a of T: If $b_1 < b_2$ are the children of a, then b_1 is called the 1-child and b_2 the 2-child of a.
- Consider the zig-zag-path which starts in the root, visits the root's 1-child, that node's 2-child, that node's 1-child, etc.
- A **full** binary tree T has even height iff the last node on the zig-zag-path is a 2-child. This can be expressed in order-inv. FO$(E, D, <)$ which states that
  - exists x_0, x_h such that
  - root(x_0), leaf(x_h), x_h is a 2-child,
  - the node x_1 satisfying $E(x_0, x_1) \wedge D(x_1, x_h)$ is a 1-child
  - for any 3 nodes u, v, w such that
    $E(u, v)$ and $E(v, w)$ and $(D(w, x_h) \vee w = x_h)$ we have:
    v is a 1-child iff w is a 2-child.

$\square$

FO < order-invariant FO — Otto's proof

Let $\sigma = \{E, \sim, \in, V, V', P'\}$ with binary $E, \sim, \in$ and unary V, V', P'.

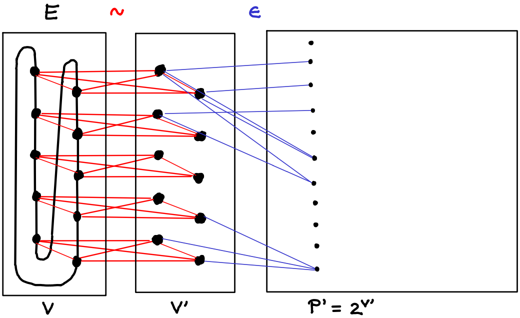
Represent a graph $G = (V, E)$ on $2n$ nodes by the σ-structure $\mathcal{S}_{2n}(G)$:



FO < order-invariant FO — Otto's proof

Let $\sigma = \{E, \sim, \in, V, V', P'\}$ with binary $E, \sim, \in$ and unary V, V', P'.

Represent a graph $G = (V, E)$ on $2n$ nodes by the σ-structure $\mathcal{S}_{2n}(G)$:



Theorem (Otto):

There is an order-inv. FO($\sigma, <$)-sentence φ_{conn}, but no FO(σ)-sentence, such that for all $n \in \mathbb{N}$ and all graphs G on $2n$ nodes we have

$$\mathcal{S}_{2n}(G) \models \varphi_{conn} \iff G \text{ is connected.}$$

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

$\exists^{0 \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

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

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

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

$\mathrm{FO{+}MOD}_2 \; < \;$ order-invariant $\mathrm{FO{+}MOD}_2$

$\mathrm{FO{+}MOD}_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{0 \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

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

$\text{FO+MOD}_2 \ < \ \text{order-invariant FO+MOD}_2$

FO+MOD_2 : the extension of FO by modulo 2 counting quantifiers

$\exists^{0 \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

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

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

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

$\exists^{0 \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

There is an order-invariant FO+MOD$_2(E)$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V^G, E^G)$ 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 permutation, i.e., $\mathrm{sgn}(\pi) = 1 \iff$

$\mathrm{FO}+\mathrm{MOD}_2 \ < \ $ order-invariant $\mathrm{FO}+\mathrm{MOD}_2$

$\mathrm{FO}+\mathrm{MOD}_2$: the extension of FO by modulo 2 counting quantifiers

$\exists^{0 \bmod 2} x\ \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

There is an order-invariant $\mathrm{FO}+\mathrm{MOD}_2(E)$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V^G, E^G)$ 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 permutation, i.e., $\mathrm{sgn}(\pi) = 1 \iff$
  π has an even number of inversions (i,j) such that $i < j$ and $\pi(i) > \pi(j)$.

$\square$

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.

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)

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)

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 $\big(V^G, \mathit{Same\_Row}, \mathit{Same\_Column}\big)$.

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 $\big(V^G, \mathit{Same\_Row}, \mathit{Same\_Column}\big)$.
  - Order-invariant MSO can express that
    the number of columns is a multiple of the number of rows.

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 $\big(V^G, \mathit{Same\_Row}, \mathit{Same\_Column}\big)$.
  - 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

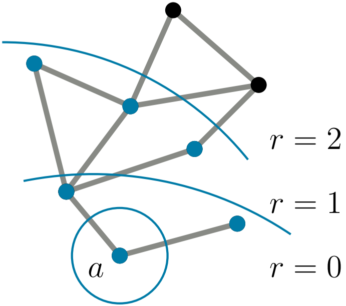
Neighborhoods

Graph $G = (V^G, E^G)$

Distance *dist*(u, v) : length of shortest path from u to v in undirected version of G.

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

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



Neighborhoods

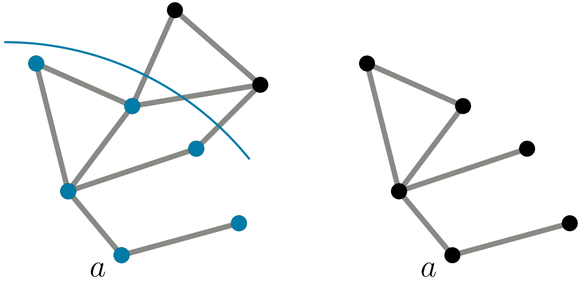
Graph $G = (V^G, E^G)$

Distance $dist(u, v)$: length of shortest path from u to v in undirected version of 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 $\mathcal{N}_r^G(a)$ of radius r at a in G : induced subgraph of G on $N_r(a)$.



Gaifman-local queries

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

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:

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

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)

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
  (i.e., q is definable in order-invariant FO),
  there is a constant c such that q is c-local.
  (Grohe, Schwentick '98)

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
  (i.e., 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 (i.e., 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)

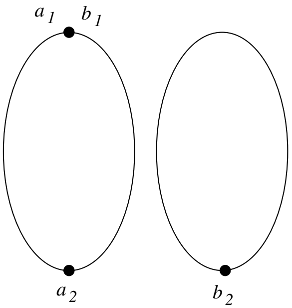
Use locality for proving non-expressibility

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.

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!

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).

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 a 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 a circuit family $\mathcal{C}$?

- Represent graph $G = (V^G, E^G)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.

Proof of Gaifman-locality theorem (2/5)

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

- Represent graph $G = (V^G, E^G)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.

- Represent a node $a \in V^G$ 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.

Proof of Gaifman-locality theorem (2/5)

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

- Represent graph $G = (V^G, E^G)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V^G$ 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).

Proof of Gaifman-locality theorem (2/5)

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

- Represent graph $G = (V^G, E^G)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V^G$ 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 $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

Proof of Gaifman-locality theorem (2/5)

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

- Represent graph $G = (V^G, E^G)$ by a bitstring $\beta(G)$ corresponding to an adjacency matrix for G.
- Represent a node $a \in V^G$ 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 $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.
- *Known:* A unary graph query $q(x)$ is definable in Arb-invariant FO $\iff$ it is computed by an AC^0-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.

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G,a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ accepts γ.

For contradiction, assume $q(x)$ is not $(\log n)^c$-local, for any $c \in \mathbb{N}$.

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ 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 $n > n_0$, $G = (V^G, E^G)$ with n nodes, $a, b \in V^G$ such that

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ 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 $n > n_0$, $G = (V^G, E^G)$ with n nodes, $a, b \in V^G$ such that for $m := (\log n)^c$, $\quad (\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, but $a \in q(G)$ and $b \notin q(G)$.

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G,a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ 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 $n > n_0$, $G = (V^G, E^G)$ with n nodes, $a, b \in V^G$ such that for $m := (\log n)^c$, $\quad (\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, but $a \in q(G)$ and $b \notin q(G)$.

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

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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G, a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ 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 $n > n_0$, $G = (V^G, E^G)$ with n nodes, $a, b \in V^G$ such that for $m := (\log n)^c$, $\quad (\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, but $a \in q(G)$ and $b \notin q(G)$.

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

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 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 an AC^0-circuit family of constant depth d and polynomial size $p(n)$ computing q.
I.e., for all $G = (V^G, E^G)$, $a \in V^G$, $\gamma \in Rep(G,a)$: $\quad a \in q(G) \iff C_{|\gamma|}$ 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 $n > n_0$, $G = (V^G, E^G)$ with n nodes, $a, b \in V^G$ such that for $m := (\log n)^c$, $\quad (\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, but $a \in q(G)$ and $b \notin q(G)$.
For simplicity, consider the special case that $dist(a,b) > 2m$.

Key Lemma:

*Let $m \in \mathbb{N}$, $G = (V,E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a,b) > 2m$.
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 (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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:

Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w \in \{0, 1\}^m$.

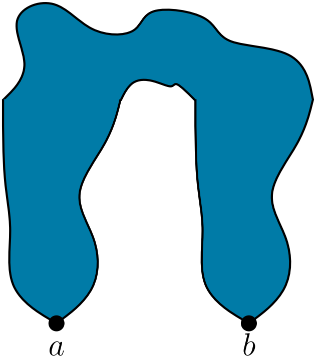
Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w \in \{0, 1\}^m$.



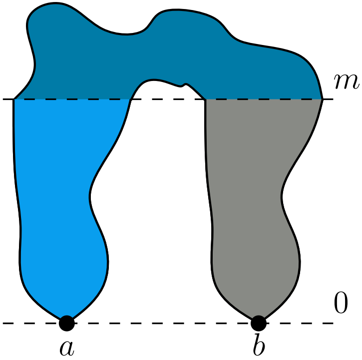
Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w \in \{0, 1\}^m$.



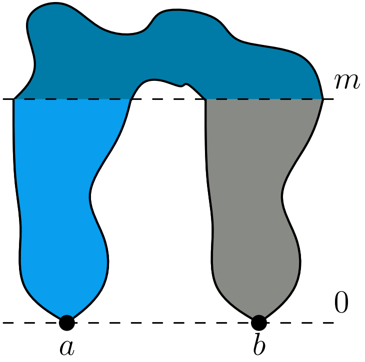
Proof of Gaifman-locality theorem (4/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w_i = 1$:



Proof of Gaifman-locality theorem (4/5)

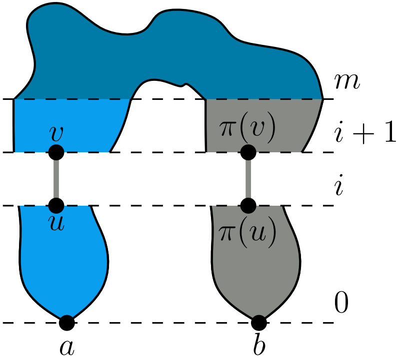
Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w_i = 1$:



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

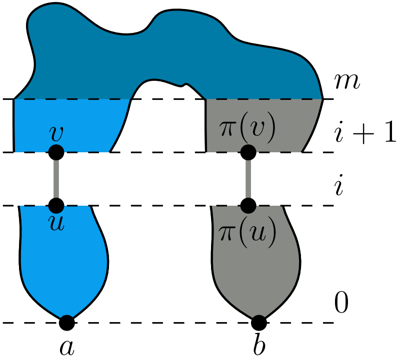
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

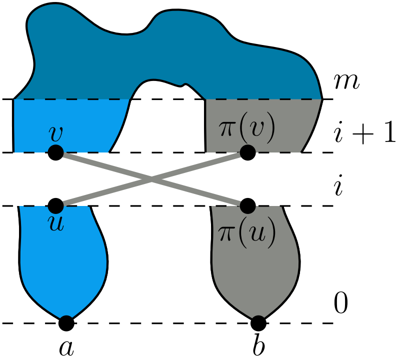
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

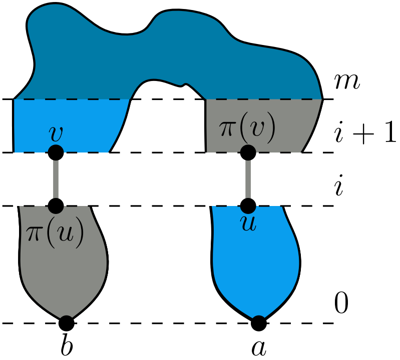
*Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

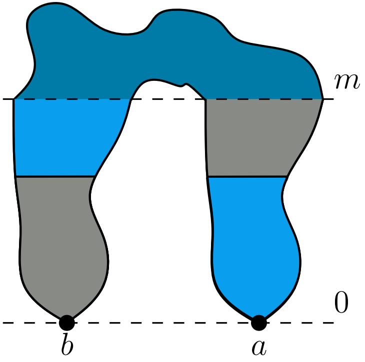
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

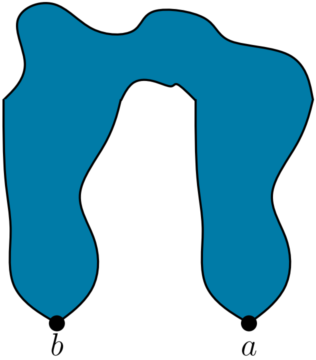
*Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
Let circuit C accept all strings in $Rep(G, a)$ and reject all strings in $Rep(G, b)$.
Then there is a circuit $\tilde{C}$ of the same size & depth as C computing parity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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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.

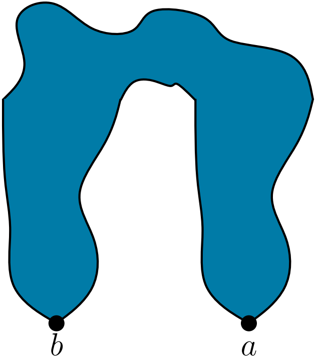
Proof:

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 $G_w \cong G$.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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.

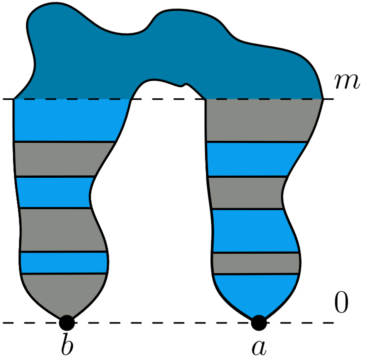
Proof:

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 $G_w \cong G$.



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$. 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:

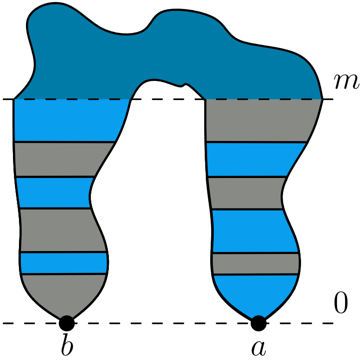
Consider $w \in \{0, 1\}^m$.

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 $G_w \cong G$.

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



Proof of Gaifman-locality theorem (4/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $w \in \{0, 1\}^m$.

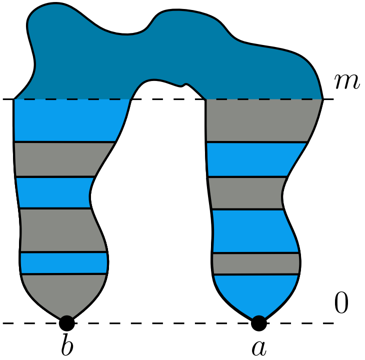
For $i \in \{0, 1, \ldots, m-1\}$ with $w_i = 1$:

> *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 $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $i < m$.

Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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 $e = \{u, v\}$, $e' = \{\pi(u), \pi(v)\}$, $\tilde{e} = \{u, \pi(v)\}$, $\tilde{e}' = \{\pi(u), v\}$.

Proof of Gaifman-locality theorem (5/5)

Key Lemma:

Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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:

$$\begin{array}{llll} e & \text{by} & (e \wedge \neg w_i) & \qquad e' \text{ by } (e' \wedge \neg w_i) \\ \tilde{e} & \text{by} & (e \wedge w_i) & \qquad \tilde{e}' \text{ by } (e' \wedge w_i) \end{array}$$

Proof of Gaifman-locality theorem (5/5)

Key Lemma:
Let $m \in \mathbb{N}$, $G = (V, E)$, $a, b \in V$ s.t. $(\mathcal{N}_m^G(a), a) \cong (\mathcal{N}_m^G(b), b)$, $dist(a, b) > 2m$.
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:

$$\begin{array}{llll} e \text{ by } (e \wedge \neg w_i) & \qquad & e' \text{ by } (e' \wedge \neg w_i) \\ \tilde{e} \text{ by } (e \wedge w_i) & \qquad & \tilde{e}' \text{ by } (e' \wedge w_i) \end{array}$$

- 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.*

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* $q_d(x)$ *that 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 $q_d(x)$ 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.

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

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* $q_d(x)$ *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.

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 $q_d(x)$ 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.

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 $c \log \log n$.

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.

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., '13)*

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.*

Locality of Arb-invariant FO+MOD_p

In a similar way, we can also prove:

Theorem: *(Harwath, S., '13)*

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

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

Theorem: *(Harwath, S., '13)*

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 the Razborov-Smolensky result for AC$^0[p]$-circuits.

Locality of Arb-invariant FO+MOD$_p$

In a similar way, we can also prove:

Theorem: *(Harwath, S., '13)*

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

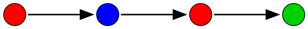
Proof: Use the Razborov-Smolensky result for AC$^0[p]$-circuits.

Corollary: Easy proof that reachability is not definable in Arb-invariant FO+MOD$_p$, for prime power p.

Overview

Represent words as labeled graphs

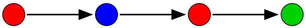
- (labeled) chain-graphs . . . . .this chain-graph represents the string $rbrg$.



Edges correspond to the successor relation "*succ*" on the positions of the string.
Write $<$ -inv-FO(*succ*) for order-invariant FO on these graphs.

Represent words as labeled graphs

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

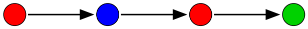


Edges correspond to the successor relation "*succ*" on the positions of the string.

Write $<$ -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

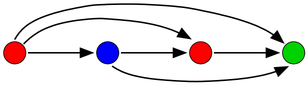
- (labeled) chain-graphs .... this chain-graph represents the string $rbrg$.



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

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

- transitive closure of (labeled) chain-graphs

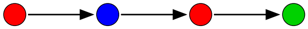


Edges correspond to the linear order "$<$" on the positions of the string.

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

Represent words as labeled graphs

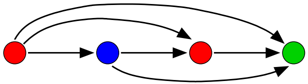
- (labeled) chain-graphs .... this chain-graph represents the string *rbrg*.



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

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

- transitive closure of (labeled) chain-graphs



Edges correspond to the linear order "$<$" on the positions of the string.

Write $+$-inv-FO($<$) for addition-invariant FO on these graphs.
Note that on these graphs, $<$ -inv-FO($<$) is the same as FO($<$).

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*)

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)

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

$<$-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)

Known results

$\text{FO}(\textit{succ})$ = locally threshold testable languages (Thomas)

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

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

$\text{MSO}(<) = \text{MSO}(\textit{succ}) = \text{MLFP}(<) = \text{MLFP}(\textit{succ})$

$<\text{-inv-FO}(\textit{succ}) = \text{FO}(\textit{succ})$ (Benedikt, Segoufin, 2005)

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

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

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

Known results

$\mathrm{FO}(succ)$ = locally threshold testable languages (Thomas)
$\mathrm{FO}(<)$ = star-free regular languages (McNaughton, Papert)
$\mathrm{MSO}(<)$ = regular languages (Büchi, Elgot, Trakhtenbrot)
$\mathrm{MSO}(<) = \mathrm{MSO}(succ) = \mathrm{MLFP}(<) = \mathrm{MLFP}(succ)$

$<$-inv-$\mathrm{FO}(succ) = \mathrm{FO}(succ)$ (Benedikt, Segoufin, 2005)
$+$-inv-$\mathrm{MLFP}(succ) \supseteq$ DLIN (S., 2004)
$+$-inv-$\mathrm{MSO}(succ)$ = the linear time hierarchy (More & Olive 1997, S. 2004)

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

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

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:
Input: a $\mathrm{FO}(<, E, C_1, \ldots, C_\ell)$-sentence φ
Question: Is φ order-invariant on all finite labeled chain-graphs?

Known results

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

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

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

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

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

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

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

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

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

ORDER-INVARIANCE ON FINITE LABELED CHAIN-GRAPHS:

Input: a $\mathrm{FO}(<, E, C_1, \ldots, C_\ell)$-sentence φ

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

The "Algebraic" Approach

Let L_1 and L_2 be logics.

Goal: Show that L_1 can define exactly the same string-languages as L_2.

Approach:

The “Algebraic” Approach

Let L_1 and L_2 be logics.

Goal: Show that L_1 can define exactly the same string-languages as L_2.

Approach:

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

The “Algebraic” Approach

Let L_1 and L_2 be logics.

Goal: Show that L_1 can define exactly the same string-languages as L_2.

Approach:

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

(1) Show that a property p of strings is definable in L_1 iff it is closed under every operation $op \in \mathcal{O}$.

The "Algebraic" Approach

Let L_1 and L_2 be logics.

Goal: Show that L_1 can define exactly the same string-languages as L_2.

Approach:

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

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

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

The "Algebraic" Approach

Let L_1 and L_2 be logics.

Goal: Show that L_1 can define exactly the same string-languages as L_2.

Approach:

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

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

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

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

An example

Theorem (Benedikt, Segoufin, '09): $<$-inv-FO(*succ*) $=$ FO(*succ*)

A string-language is definable in $<$-inv-FO(*succ*) iff it is definable in FO(*succ*).

An example

Theorem (Benedikt, Segoufin, '09): $<$-inv-FO(*succ*) $=$ FO(*succ*)
A string-language is definable in $<$-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.

An example

> *Theorem (Benedikt, Segoufin, '09):* $<$-inv-FO(*succ*) $=$ FO(*succ*)
>
> A string-language is definable in $<$-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

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

An example

Theorem (Benedikt, Segoufin, '09): $<$-inv-FO($succ$) $=$ FO($succ$)

A string-language is definable in $<$-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

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

  - 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

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

An example

Theorem (Benedikt, Segoufin, '09): $<$-inv-FO($succ$) $=$ FO($succ$)
A string-language is definable in $<$-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

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

  - 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

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

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

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

Different situation for FO+MOD_2

Proposition (Harwath, S., '13): $<$-inv-FO+MOD_2(*succ*) $\neq$ FO+MOD_2(*succ*)
There exists a string-language L which is
definable in $<$-inv-FO+MOD_2(*succ*), but not in FO+MOD_2(*succ*).

Different situation for FO+MOD$_2$

Proposition (Harwath, S., '13): $<$-inv-FO+MOD$_2$(*succ*) $\neq$ FO+MOD$_2$(*succ*)
There exists a string-language L which is
definable in $<$-inv-FO+MOD$_2$(*succ*), but not in FO+MOD$_2$(*succ*).

L looks as follows:

- $L \;=\; L_1 \cup L_2$

Different situation for FO+MOD_2

Proposition (Harwath, S., '13): $<$-inv-FO+$\text{MOD}_2(\mathit{succ}) \neq$ FO+$\text{MOD}_2(\mathit{succ})$

There exists a string-language L which is
definable in $<$-inv-FO+$\text{MOD}_2(\mathit{succ})$, but not in FO+$\text{MOD}_2(\mathit{succ})$.

L looks as follows:

- $L = L_1 \cup L_2$
- $L_1 = \{ w \in 1^* \, 2 \, 0^* \, 1^* \, 0^* \ : \ |w|_1 \text{ odd} \}$

Different situation for FO+MOD$_2$

Proposition (Harwath, S., '13): $<$-inv-FO+MOD$_2$(*succ*) $\neq$ FO+MOD$_2$(*succ*)
There exists a string-language L which is
definable in $<$-inv-FO+MOD$_2$(*succ*), but not in FO+MOD$_2$(*succ*).

L looks as follows:

- $L = L_1 \cup L_2$
- $L_1 = \{ w \in 1^* \, 2 \, 0^* \, 1^* \, 0^* \; : \; |w|_1 \text{ odd} \}$
- $L_2 = \{ w \in 1^* \, 0^* \, 1^* \, 2 \, 0^* \; : \; |w|_1 \text{ even} \}$

Different situation for FO+MOD$_2$

Proposition (Harwath, S., '13): $<$-inv-FO+MOD$_2$(*succ*) $\neq$ FO+MOD$_2$(*succ*)

There exists a string-language L which is
definable in $<$-inv-FO+MOD$_2$(*succ*), but not in FO+MOD$_2$(*succ*).

L looks as follows:

- $L \; = \; L_1 \cup L_2$
- $L_1 \; = \; \{ \, w \, \in \, 1^* \, 2 \, 0^* \, 1^* \, 0^* \; : \; |w|_1 \text{ odd} \, \}$
- $L_2 \; = \; \{ \, w \, \in \, 1^* \, 0^* \, 1^* \, 2 \, 0^* \; : \; |w|_1 \text{ even} \, \}$

L is NOT definable in FO+MOD$_2$(*succ*):

- Use the Hanf-locality of FO+MOD(*succ*)

Different situation for FO+MOD$_2$

Proposition (Harwath, S., '13): $<$-inv-FO+MOD$_2$(*succ*) $\neq$ FO+MOD$_2$(*succ*)
There exists a string-language L which is definable in $<$-inv-FO+MOD$_2$(*succ*), but not in FO+MOD$_2$(*succ*).

L looks as follows:

- $L \;=\; L_1 \cup L_2$
- $L_1 \;=\; \{\, w \in 1^*\,2\,0^*\,1^*\,0^* \;:\; |w|_1 \text{ odd} \,\}$
- $L_2 \;=\; \{\, w \in 1^*\,0^*\,1^*\,2\,0^* \;:\; |w|_1 \text{ even} \,\}$

L is NOT definable in FO+MOD$_2$(*succ*):

- Use the Hanf-locality of FO+MOD(*succ*)

L is definable in $<$-inv-FO+MOD$_2$(*succ*):

- By a FO-reduction using Niemistö's $<$-inv-FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$, discussed at the beginning of this tutorial.

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

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

$\exists^{0 \bmod 2} x\, \psi(x)$: the number of nodes x satisfying $\psi(x)$ is congruent 0 modulo 2.

Theorem (Niemistö):

There is an order-invariant FO+MOD$_2(E)$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite directed graph $G = (V^G, E^G)$ 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 permutation, i.e., $\operatorname{sgn}(\pi) = 1$ $\iff$
  π has an even number of inversions (i, j) such that $i < j$ and $\pi(i) > \pi(j)$.

$\square$

- $L \;=\; L_1 \cup L_2$
- $L_1 \;=\; \{\, w \in 1^*\,2\,0^*\,1^*\,0^* \;:\; |w|_1 \text{ odd} \,\}$
- $L_2 \;=\; \{\, w \in 1^*\,0^*\,1^*\,2\,0^* \;:\; |w|_1 \text{ even} \,\}$

L is definable in $<$-inv-FO+MOD$_2$($succ$):

- By a FO-reduction using Niemistö's $<$-inv-FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$, expressing that the number of even-length cycles is even:

Simulate a disjoint union of cycles by a FO-formula that adds edges

- from the unique 2-position to the first position

- $L = L_1 \cup L_2$
- $L_1 = \{ w \in 1^* \, 2 \, 0^* \, 1^* \, 0^* \ : \ |w|_1 \text{ odd} \}$
- $L_2 = \{ w \in 1^* \, 0^* \, 1^* \, 2 \, 0^* \ : \ |w|_1 \text{ even} \}$

L is definable in $<$-inv-FO+MOD$_2$($succ$):

- By a FO-reduction using Niemistö's $<$-inv-FO+MOD$_2(E)$-sentence $\varphi_{\textit{even cycles}}$, expressing that the number of even-length cycles is even:

Simulate a disjoint union of cycles by a FO-formula that adds edges

- from the unique 2-position to the first position
- from the unique 1 directly left to a 0 to the unique 1 directly right to 0,

- $L \; = \; L_1 \cup L_2$
- $L_1 \; = \; \{\, w \in 1^*\, 2\, 0^*\, 1^*\, 0^* \;:\; |w|_1 \text{ odd} \,\}$
- $L_2 \; = \; \{\, w \in 1^*\, 0^*\, 1^*\, 2\; 0^* \;:\; |w|_1 \text{ even} \,\}$

L is definable in $<$-inv-FO+MOD$_2$($succ$):

- By a FO-reduction using Niemistö's $<$-inv-FO+MOD$_2$(E)-sentence $\varphi_{\textit{even cycles}}$, expressing that the number of even-length cycles is even:

Simulate a disjoint union of cycles by a FO-formula that adds edges

- from the unique 2-position to the first position
- from the unique 1 directly left to a 0 to the unique 1 directly right to 0,

and then deletes all 0-positions.

- $L \;=\; L_1 \cup L_2$
- $L_1 \;=\; \{\, w \in 1^*\, 2\, 0^*\, 1^*\, 0^* \;:\; |w|_1 \text{ odd} \,\}$
- $L_2 \;=\; \{\, w \in 1^*\, 0^*\, 1^*\, 2\; 0^* \;:\; |w|_1 \text{ even} \,\}$

L is definable in $<$-inv-FO+MOD$_2$($succ$):

- By a FO-reduction using Niemistö's $<$-inv-FO+MOD$_2$(E)-sentence $\varphi_{\text{even cycles}}$, expressing that the number of even-length cycles is even:

Simulate a disjoint union of cycles by a FO-formula that adds edges

- from the unique 2-position to the first position
- from the unique 1 directly left to a 0 to the unique 1 directly right to 0,

and then deletes all 0-positions.

Situation:

- $w \in 1^*\, 2\, 0^*\, 1^*\, 0^* \quad\Longrightarrow\quad$ 2 cycles, sum of lenghts: $|w|_1 + 1$
- $w \in 1^*\, 0^*\, 1^*\, 2\; 0^* \quad\Longrightarrow\quad$ 1 cycle, length: $|w|_1 + 1$

$\square$

Further results proved by the algebraic approach

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.)

Further results proved by the algebraic approach

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)

Further results proved by the algebraic approach

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 $\text{FO}_{\text{card}}(\mathit{succ})$. (S., Segoufin '10 and Harwath, S. '12)
  (They use closure under transfers and closure under guarded swaps.)

Further results proved by the algebraic approach

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)

Further results proved by the algebraic approach

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 $\text{FO}_{\text{card}}(\mathit{succ})$. (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 $\text{FO}_{\text{card}}(\mathit{succ})$. (Anderson, van Melkebeek, S., Segoufin '11)

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in* $+$-*inv*-FO($<$).

(2) *L is definable in* $\text{FO}_{\text{Card}}(<)$.

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

Definition: $\mathrm{FO}_{\mathrm{Card}}(<)$

FO($<$) with length modulo predicates $lm(i,q)$, for all $i, q \in \mathbb{N}$:

$$\forall\, w \in \Sigma^* \;:\; w \models lm(i,q) \iff |w| \equiv i \bmod q.$$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

Definition: $\mathrm{FO}_{\mathrm{Card}}(<)$

FO($<$) with length modulo predicates $lm(i, q)$, for all $i, q \in \mathbb{N}$:

$$\forall\, w \in \Sigma^* \;:\; w \models lm(i,q) \iff |w| \equiv i \bmod q.$$

Regular languages definable in $+$-inv-FO$(<)$

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO$(<)$.*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO$(<)$-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Definition: $\mathrm{FO}_{\mathrm{Card}}(<)$

FO$(<)$ with length modulo predicates $lm(i,q)$, for all $i, q \in \mathbb{N}$:

$$\forall\, w \in \Sigma^* \;:\; w \models lm(i,q) \iff |w| \equiv i \bmod q.$$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\text{FO}_{\text{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Definition: L is closed under transfers $\iff$ $\forall\, x, y, z \in \Sigma^*$ we have:

$$\text{if} \quad |x| = |z|, \quad \text{then} \quad x^r x y z^r =_L x^r y z z^r.$$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \ = \ \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Definition: L is closed under transfers $\iff$ $\forall\, x, y, z \in \Sigma^*$ we have:

$$\text{if} \quad |x| = |z|, \quad \text{then} \quad x^r x y z^r =_L x^r y z z^r .$$

syntactic congruence $=_L$: $\quad x =_L y \iff \big(\forall\, u, v \in \Sigma^* \ : \ uxv \in L \iff uyv \in L\,\big)$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- *$Z_i^q \ = \ \{w : |w| \equiv i \bmod q\}$.*

(4) *L is closed under transfers.*

Definition: L is closed under transfers $\iff$ $\forall\, x, y, z \in \Sigma^*$ we have:

$$\text{if} \quad |x| = |z|, \quad \text{then} \quad x^r x y z^r =_L x^r y z z^r .$$

syntactic congruence $=_L$: $\quad x =_L y \iff \big(\forall\, u, v \in \Sigma^* \ : \ uxv \in L \iff uyv \in L\,\big)$

$z \in \Sigma^*$ is called idempotent $\iff$ $zz \ =_L \ z$.

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Definition: L is closed under transfers $\iff$ $\forall\, x, y, z \in \Sigma^*$ we have:

$$\text{if} \quad |x| = |z|, \quad \text{then} \quad x^r x y z^r =_L x^r y z z^r.$$

syntactic congruence $=_L$: $\quad x =_L y \iff \big(\forall\, u, v \in \Sigma^* \;:\; uxv \in L \iff uyv \in L\,\big)$

$z \in \Sigma^*$ is called idempotent $\iff$ $zz \;=_L\; z$.

L regular $\implies \exists\, r \in \mathbb{N} : \forall\, x \in \Sigma^*$, x^r is idempotent

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\text{FO}_{\text{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- *$Z_i^q \ = \ \{w : |w| \equiv i \bmod q\}$.*

(4) *L is closed under transfers.*

Definition: L is closed under transfers $\iff$ $\forall\, x, y, z \in \Sigma^*$ we have:

$$\text{if} \quad |x| = |z|, \quad \text{then} \quad x^r x y z^r =_L x^r y z z^r.$$

Observation:

Given an automaton for a regular language L, it is decidable whether L is closed under transfers.

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \; = \; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Proof: (2)$\Longleftrightarrow$(3): easy.

Regular languages definable in $+$-inv-FO($<$)

Theorem: (S., Segoufin, 2010)

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\text{FO}_{\text{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Proof: (2)$\iff$(3): easy.

(2)$\implies$(1): easy. E.g.: $|w| \equiv 1 \bmod 2 \quad\iff$

$$w \models \exists x\, \exists z\, \big(x + x = z \wedge \forall y\,(\,y < z \vee y = z\,)\big)$$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in* $+$-*inv*-FO($<$).

(2) *L is definable in* $\mathrm{FO}_{\mathrm{Card}}(<)$.

(3) *L is a finite union of languages of the form* $S \cap Z_i^q$, *where*

- *S is star-free regular (i.e., S is* FO($<$)*-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Proof:

(2)$\Longleftrightarrow$(3): easy.

(2)$\Longrightarrow$(1): easy. E.g.: $|w| \equiv 1 \bmod 2 \iff$

$$w \models \exists x\, \exists z\, \big(x + x = z \,\wedge\, \forall y\, (\, y < z \vee y = z\,)\big)$$

(1)$\Longrightarrow$(4): use Ehrenfeucht-Fraïssé games.

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Proof:

(2)$\Longleftrightarrow$(3): easy.

(2)$\Longrightarrow$(1): easy. E.g.: $|w| \equiv 1 \bmod 2 \quad \Longleftrightarrow$

$$w \models \exists x\, \exists z\, \big(x + x = z \,\wedge\, \forall y\, (\, y < z \vee y = z\,)\big)$$

(1)$\Longrightarrow$(4): use Ehrenfeucht-Fraïssé games.

(4)$\Longrightarrow$(2): use tools from algebraic automata theory.

Proof of (4)$\Longrightarrow$(2): L is regular & closed under transfers.
Goal: Show that L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.

Choose a suitable number $q > 0$.

skip proof

For $0 \leqslant i < q$ let $L_i \ := \ L \cap Z_i^q$, where $Z_i^q \ := \ \{w \in \Sigma^* : |w| \equiv i \bmod q\}$.

Clearly, $L = \bigcup_{0 \leqslant i < q} L_i$.

Proof of (4)$\Longrightarrow$(2): L is regular & closed under transfers.

Goal: Show that L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.

Choose a suitable number $q > 0$.

skip proof

For $0 \leqslant i < q$ let $L_i \ := \ L \cap Z_i^q$, where $Z_i^q \ := \ \{w \in \Sigma^* : |w| \equiv i \bmod q\}$.

Clearly, $L = \bigcup_{0 \leqslant i < q} L_i$.

Goal: Show that L_i is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.

Proof of (4)$\Longrightarrow$(2): L is regular & closed under transfers.
Goal: Show that L is definable in $\text{FO}_{\text{Card}}(<)$.

Choose a suitable number $q > 0$.

skip proof

For $0 \leqslant i < q$ let $L_i := L \cap Z_i^q$, where $Z_i^q := \{w \in \Sigma^* : |w| \equiv i \bmod q\}$.

Clearly, $L = \bigcup_{0 \leqslant i < q} L_i$.

Goal: Show that L_i is definable in $\text{FO}_{\text{Card}}(<)$.

Approach: Find a regular language M_i such that

- $L_i = M_i \cap Z_i^q$,
- The minimal DFA for M_i does not contain any counter.

Then, apply

Theorem: *(McNaughton & Papert, 1971)*

Let M be a regular language. Then, the following are equivalent:

(1) The minimal DFA for M does not contain any counter.

(2) M is definable in FO($<$) *(i.e., M is star-free regular).*

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).
Goal: Show that L is closed under transfers.

skip proof

For contradiction, assume that L is not closed under transfers. Then:

$$\exists\, x, y, z, u, v \in \Sigma^* \;:\quad |x| = |z| \quad \text{and}$$
$$u\, x^r\, x\, y\, z^r\, v \;\in\; L \quad \text{and} \quad u\, x^r\, y\, z\, z^r\, v \;\notin\; L.$$

Thus:

$$\forall\, \alpha,\ \beta \geqslant 1 \;:\quad \Big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \;\in\; L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v \;\notin\; L \,\Big).$$

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ : \quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\big).$$

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ :\quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyv\,x\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyv\,x\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ :\quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1 \ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2 \ :\ w_2 \not\models \psi.$$

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).
Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1\ :\quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v\ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz \mid zz)^*\ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz \mid zz)^*\ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1\ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2\ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<, +$)-formula that separates L_1 from L_2.

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ :\quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz\,|\,zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz\,|\,zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1 \ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2 \ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<,+$)-formula that separates L_1 from L_2.

Proof: Construct a FO($<,+$) interpretation that, on $w \in uyvx\,(xz\,|\,zz)^*$, evaluates φ on the corrensponding string w' of the form $u\,(x)^*\,y\,(z)^*\,v$.

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ :\quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v \ \notin\ L\,\big).$$

Idea: Consider the languages

$$L_1 := \{\, w \in\ u y v\, x\, (xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\,\},$$

$$L_2 := \{\, w \in\ u y v\, x\, (xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\,\}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1 \ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2 \ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<,+$)-formula that separates L_1 from L_2.

Proof: Construct a FO($<,+$) interpretation that, on $w \in\ u y v\, x\, (xz \mid zz)^*$, evaluates φ on the corrensponding string w' of the form $u\,(x)^*\, y\,(z)^*\, v$.

Clearly,
- $w \in L_1 \implies w' \in L \implies w' \models \varphi$,
- $w \in L_2 \implies w' \notin L \implies w' \not\models \varphi$. □

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ : \quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v \ \notin\ L \,\big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz \mid zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1 \ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2 \ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<, +$)-formula that separates L_1 from L_2.

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ : \quad \left(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\right).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz\,|\,zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz\,|\,zz)^* \ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\ w_1 \in L_1\ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\ w_2 \in L_2\ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<, +$)-formula that separates L_1 from L_2.

Lemma 2: No formula of FO($<, +$) can separate L_1 from L_2.

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1\ :\quad \left(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v\ \in\ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v\ \notin\ L\,\right).$$

Idea: Consider the languages

$$L_1 := \{\ w \in\ uyvx\,(xz\,|\,zz)^*\ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 1\ [r],\ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in\ uyvx\,(xz\,|\,zz)^*\ :\ |w|_x, |w|_z \geqslant r,\ \ |w|_x \equiv 0\ [r],\ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1\ :\ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2\ :\ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<, +$)-formula that separates L_1 from L_2.

Lemma 2: No formula of FO($<, +$) can separate L_1 from L_2.

Proof idea: Use Ehrenfeucht-Fraïssé games.

Proof of (1)$\Longrightarrow$(4): Let L be regular and definable by a φ in $+$-inv-FO($<$).

Goal: Show that L is closed under transfers. **Proof by contradiction:**

Situation: Fixed $x, y, z, u, v \in \Sigma^*$ with $|x| = |z|$ such that

$$\forall\, \alpha,\ \beta \geqslant 1 \ : \quad \big(\, u\, x^{\alpha r}\, x\, y\, z^{\beta r}\, v \ \in \ L \quad \text{and} \quad u\, x^{\alpha r}\, y\, z\, z^{\beta r}\, v \ \notin \ L\, \big).$$

Idea: Consider the languages

$$L_1 := \{\ w \in \ uyv\,x\,(xz\,|\,zz)^* \ : \ |w|_x, |w|_z \geqslant r, \ \ |w|_x \equiv 1\ [r], \ \ |w|_z \equiv 0\ [r]\ \},$$

$$L_2 := \{\ w \in \ uyv\,x\,(xz\,|\,zz)^* \ : \ |w|_x, |w|_z \geqslant r, \ \ |w|_x \equiv 0\ [r], \ \ |w|_z \equiv 1\ [r]\ \}.$$

Definition: A formula ψ separates L_1 from L_2 $\iff$

$$\forall\, w_1 \in L_1 \ : \ w_1 \models \psi \qquad \text{and} \qquad \forall\, w_2 \in L_2 \ : \ w_2 \not\models \psi.$$

Lemma 1: If L is definable in $+$-inv-FO($<$), then there is a FO($<,+$)-formula that separates L_1 from L_2.

Lemma 2: No formula of FO($<,+$) can separate L_1 from L_2.

CONTRADICTION. $\square$

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q = \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Regular languages definable in $+$-inv-FO($<$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($<$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(<)$.*

(3) *L is a finite union of languages of the form $S \cap Z_i^q$, where*

- *S is star-free regular (i.e., S is FO($<$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers.*

Question: What happens if the linear order $<$ on the string is not available?

We first consider the case where only the successor relation "*succ*" is available.

Regular languages definable in $+$-inv-FO($succ$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($succ$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(succ)$.*

(3) *L is a finite union of languages of the form $T \cap Z_i^q$, where*

- *T is locally threshold testable (i.e., T is FO($succ$)-definable)*
- $Z_i^q \;=\; \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers and under swaps.*

Proof method: Similar as for the previous theorem.

Definition: L is closed under swaps $\iff$

for all $e, f, x, y, z \in \Sigma^*$ such that e, f are idempotent we have

$$e\,x\,f\,y\,e\,z\,f \;\;=_L\;\; e\,z\,f\,y\,e\,x\,f$$

Observation: Given an automaton for a regular language L, it is decidable whether L is closed under transfers and under swaps.

Regular languages definable in $+$-inv-FO($succ$)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO($succ$).*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(succ)$.*

(3) *L is a finite union of languages of the form $T \cap Z_i^q$, where*

- *T is locally threshold testable (i.e., T is FO($succ$)-definable)*
- $Z_i^q \ = \ \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers and under swaps.*

By combining this with the poly-logarithmic-locality of Arb-invariant FO, we obtain:

Theorem: *(Anderson, Melkebeek, S., Segoufin, 2011)*

Let L be a regular language. Then,
L is definable in Arb-invariant FO($succ$) iff L is definable in $\mathrm{FO}_{\mathrm{Card}}(succ)$.

Regular languages definable in $+$-inv-FO(*succ*)

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO(succ).*

(2) *L is definable in $\text{FO}_{\text{Card}}(succ)$.*

(3) *L is a finite union of languages of the form $T \cap Z_i^q$, where*

- *T is locally threshold testable (i.e., T is FO(succ)-definable)*
- $Z_i^q \ = \ \{w : |w| \equiv i \bmod q\}$.

(4) *L is closed under transfers and under swaps.*

The result extends from words to trees:

Theorem: *(Harwath, S., 2012)*

Let L be a regular tree language. The following are equivalent:

(1) *L is definable in $+$-inv-FO(S_1, S_2).*

(2) *L is definable in $\text{FO}_{\text{Card}}(S_1, S_2)$.*

(3) *L is closed under transfers and swaps.*

Regular languages definable in $+$-inv-FO$(=)$

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

Let L be a regular language. The following are equivalent:

(1) *L is definable in $+$-inv-FO$(=)$.*

(2) *L is definable in $\mathrm{FO}_{\mathrm{Card}}(=)$.*

(3) *L is commutative, closed under transfers and under swaps.*

Definition: L is commutative $\iff$

$$\forall\, m \in \mathbb{N} \quad \forall\, a_1, \ldots, a_m \in \Sigma \quad \forall \text{ permutations } \pi \text{ of } \{1, \ldots, m\} \;:$$

$$a_1 a_2 \cdots a_m \;\in\; L \quad \iff \quad a_{\pi(1)} a_{\pi(2)} \cdots a_{\pi(m)} \;\in\; L.$$

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 $\iff$

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

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

Every bounded 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 ...

Bounded languages

Definition: (Ginsburg & Spanier, 1964)

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

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

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 $\mathrm{FO}_{\mathrm{Card}}(=)$.

$\square$

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$(<, +, C)$-sentence φ (C a unary relation symbol)

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

Overview

Gaifman-locality

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

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$+$MODp, for **odd** prime powers p, are Hanf-local w.r.t. a poly-logarithmic locality radius. For **even** p, they aren't. (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!