Width Minimization for Existential Positive Queries

Simone Bova

Technische Universität Wien

joint work with Hubie Chen

14th International Workshop on
Logic and Computational Complexity (LCC'13)
Torino, 6 September 2013

Outline

Outline

Research Motivation

Previous Work

Our Result

Other Results and Open Problems

Minimization (or Expressibility) Problem

Notation

- FO *denotes relational first-order sentences;*
- FO^k *denotes* FO-*sentences using at most k variable symbols;*
- $\mathrm{width}(\phi)$ *is the the maximum number of free variables over subformulas of* ϕ.

Minimizing number of variable symbols in FO-sentences (decision version):

Problem FO-Express

Instance $\phi \in \mathrm{FO}$ and $k \in \mathbb{N}$.

Question Is there $\psi \in \mathrm{FO}^k$ such that ϕ is logically equivalent to ψ?

Minimization (or Expressibility) Problem

Notation

- FO *denotes relational first-order sentences;*
- FO^k *denotes* FO*-sentences using at most k variable symbols;*
- $\mathrm{width}(\phi)$ *is the the maximum number of free variables over subformulas of* ϕ*.*

Minimizing number of variable symbols in FO-sentences (decision version):

Problem FO-EXPRESS

Instance $\phi \in \mathrm{FO}$ and $k \in \mathbb{N}$.

Question Is there $\psi \in \mathrm{FO}^k$ such that ϕ is logically equivalent to ψ?

Theorem (Folklore)
FO-EXPRESS *is undecidable.*

Minimization | Model Checking

However, minimization and expressibility are important wrt *algorithmic* and *complexity* aspects of the ModelChecking problem:

Given a finite structure $\mathbf{A}$ *and a* FO-*sentence* ϕ, *decide whether* $\mathbf{A} \models \phi$.

Minimization | Model Checking

However, minimization and expressibility are important wrt *algorithmic* and *complexity* aspects of the MODELCHECKING problem:

Given a finite structure $\mathbf{A}$ *and a* FO-*sentence* ϕ, *decide whether* $\mathbf{A} \models \phi$.

The MODELCHECKING problem is PSPACE-complete. But, analyzing the time complexity of the natural recursive evaluation of ϕ in $\mathbf{A}$ . . .

Proposition (Vardi)

Let $(\mathbf{A}, \phi)$ *be an instance of* MODELCHECKING.
The question, $\mathbf{A} \models \phi$?, *is decidable in time* $O(|A|^{\mathrm{width}(\phi)})$.

Minimization | Model Checking

However, minimization and expressibility are important wrt *algorithmic* and *complexity* aspects of the MODELCHECKING problem:

Given a finite structure $\mathbf{A}$ *and a* FO*-sentence* ϕ*, decide whether* $\mathbf{A} \models \phi$*.*

The MODELCHECKING problem is PSPACE-complete. But, analyzing the time complexity of the natural recursive evaluation of ϕ in $\mathbf{A}$ . . .

Proposition (Vardi)

Let $(\mathbf{A}, \phi)$ *be an instance of* MODELCHECKING*.*
The question, $\mathbf{A} \models \phi$*?, is decidable in time* $O(|A|^{\mathrm{width}(\phi)})$*.*

If $\phi \in \mathrm{FO}^k$, then $\mathrm{width}(\phi) \leq k$.

Minimization | Model Checking

However, minimization and expressibility are important wrt *algorithmic* and *complexity* aspects of the MODELCHECKING problem:

> *Given a finite structure* $\mathbf{A}$ *and a* FO-*sentence* ϕ*, decide whether* $\mathbf{A} \models \phi$.

The MODELCHECKING problem is PSPACE-complete. But, analyzing the time complexity of the natural recursive evaluation of ϕ in $\mathbf{A}$ . . .

Proposition (Vardi)

Let $(\mathbf{A}, \phi)$ *be an instance of* MODELCHECKING.
The question, $\mathbf{A} \models \phi$*?, is decidable in time* $O(|A|^{\text{width}(\phi)})$.

If $\phi \in \text{FO}^k$, then $\text{width}(\phi) \leq k$. If $\text{width}(\phi) \leq k$, then there is a polytime computable $\phi' \in \text{FO}^k$ st $\phi' \equiv \phi$.

Minimization | Model Checking

However, minimization and expressibility are important wrt *algorithmic* and *complexity* aspects of the MODELCHECKING problem:

Given a finite structure $\mathbf{A}$ *and a* FO*-sentence* ϕ*, decide whether* $\mathbf{A} \models \phi$*.*

The MODELCHECKING problem is PSPACE-complete. But, analyzing the time complexity of the natural recursive evaluation of ϕ in $\mathbf{A}$ . . .

Proposition (Vardi)

Let $(\mathbf{A}, \phi)$ *be an instance of* MODELCHECKING*.*
The question, $\mathbf{A} \models \phi$*?, is decidable in time* $O(|A|^{\mathrm{width}(\phi)})$*.*

If $\phi \in \mathrm{FO}^k$, then $\mathrm{width}(\phi) \leq k$. If $\mathrm{width}(\phi) \leq k$, then there is a polytime computable $\phi' \in \mathrm{FO}^k$ st $\phi' \equiv \phi$.

Minimization in Model Checking: Minimizing the number of variables in ϕ also minimizes the exponent in the runtime of the natural model checking algorithm.

Expressibility | Complexity Criterion

A relevant example of the MODELCHECKING problem is *query evaluation*, evaluating a *query* ϕ over a relational *database* $\mathbf{A}$.

Expressibility | Complexity Criterion

A relevant example of the MODELCHECKING problem is *query evaluation*, evaluating a *query* ϕ over a relational *database* $\mathbf{A}$.

Typically the query is relatively *small* and the database is relatively *large*.

Expressibility | Complexity Criterion

A relevant example of the MODELCHECKING problem is *query evaluation*, evaluating a *query* ϕ over a relational *database* $\mathbf{A}$.

Typically the query is relatively *small* and the database is relatively *large*.

In this setting, one considers computationally tractable a two stage procedure where a *query optimization* algorithm of possibly high complexity (eg, a *width minimization* algorithm) is followed by an *evaluation algorithm* that evaluates the optimized query in polytime.

Expressibility | Complexity Criterion

A relevant example of the ModelChecking problem is *query evaluation*, evaluating a *query* ϕ over a relational *database* $\mathbf{A}$.

Typically the query is relatively *small* and the database is relatively *large*.

In this setting, one considers computationally tractable a two stage procedure where a *query optimization* algorithm of possibly high complexity (eg, a *width minimization* algorithm) is followed by an *evaluation algorithm* that evaluates the optimized query in polytime.

This approach yields a relaxation of the notion of polynomial-time tractability, called *fixed-parameter tractability*, viable in the database setting.

Expressibility | Complexity Criterion

A relevant example of the MODELCHECKING problem is *query evaluation*, evaluating a *query* ϕ over a relational *database* $\mathbf{A}$.

Typically the query is relatively *small* and the database is relatively *large*.

In this setting, one considers computationally tractable a two stage procedure where a *query optimization* algorithm of possibly high complexity (eg, a *width minimization* algorithm) is followed by an *evaluation algorithm* that evaluates the optimized query in polytime.

This approach yields a relaxation of the notion of polynomial-time tractability, called *fixed-parameter tractability*, viable in the database setting.

Expressibility as Complexity Criterion: With respect to basic and fundamental classes of queries in database theory, "expressibility characterizes tractability" in a precise sense.

Expressibility | Complexity Criterion

Let PP $\subseteq$ FO be the class of *primitive positive* ($\exists$, $\wedge$) sentences (aka *conjunctive queries* in database theory).

*L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let PP $\subseteq$ FO be the class of *primitive positive* ($\exists, \wedge$) sentences (aka *conjunctive queries* in database theory).

Theorem (Dalmau, Kolaitis, and Vardi; Grohe)

Let L $\subseteq$ PP *be a class of sentences. The following are equivalent:* *

- MODELCHECKING *restricted to* L *is fixed-parameter tractable.*
- *There exists* $k \geq 1$ *st* L $\subseteq$ PP^k-EXPRESS.

*L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let PP $\subseteq$ FO be the class of *primitive positive* ($\exists$, $\wedge$) sentences (aka *conjunctive queries* in database theory).

Theorem (Dalmau, Kolaitis, and Vardi; Grohe)

Let L $\subseteq$ PP *be a class of sentences. The following are equivalent:* *

- MODELCHECKING *restricted to* L *is fixed-parameter tractable.*
- *There exists* $k \geq 1$ *st* L $\subseteq$ PP^k-EXPRESS.

Theorem (Dalmau, Kolaitis, and Vardi)

PP^k-EXPRESS *is* NP*-complete* ($k \geq 2$).

---

*L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let EP $\subseteq$ FO be the class of *existential positive* ($\exists$, $\wedge$, $\vee$) sentences (aka *union of conjunctive queries* in database theory).

† L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let EP $\subseteq$ FO be the class of *existential positive* ($\exists, \wedge, \vee$) sentences (aka *union of conjunctive queries* in database theory).

Theorem (Chen)

Let L $\subseteq$ EP *be a class of sentences. The following are equivalent:* †

- MODELCHECKING *restricted to* L *is fixed-parameter tractable.*
- *There exists* $k \geq 1$ *st* L $\subseteq$ EP^k-EXPRESS.

† L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let EP $\subseteq$ FO be the class of *existential positive* ($\exists, \wedge, \vee$) sentences (aka *union of conjunctive queries* in database theory).

Theorem (Chen)

Let L $\subseteq$ EP *be a class of sentences. The following are equivalent:* †

- MODELCHECKING *restricted to* L *is fixed-parameter tractable.*
- *There exists* $k \geq 1$ *st* L $\subseteq$ EP^k-EXPRESS.

Question

What is the complexity of EP^k-EXPRESS*?*

† L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Expressibility | Complexity Criterion

Let EP $\subseteq$ FO be the class of *existential positive* ($\exists, \wedge, \vee$) sentences (aka *union of conjunctive queries* in database theory).

Theorem (Chen)

Let L $\subseteq$ EP *be a class of sentences. The following are equivalent:* †

- MODELCHECKING *restricted to* L *is fixed-parameter tractable.*
- *There exists* $k \geq 1$ *st* L $\subseteq$ EP^k-EXPRESS.

Question

What is the complexity of EP^k-EXPRESS*?*

Theorem (B, Chen)

EP^k-EXPRESS *is* Π_2^p*-complete* ($k \geq 3$).

---

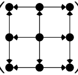
†L has bounded arity. Unless W[1] $\subseteq$ nuFPT.

Outline

Primitive Positive Logic | Minimization

Example (Grid)

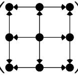
$$\gamma = \exists x_1 \ldots \exists x_9 \Big(\quad \Big)$$



Primitive Positive Logic | Minimization

Example (Grid)

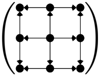
$$\gamma = \exists x_1 \ldots \exists x_9 \Bigg(\Bigg)$$



$$= \exists x_1 \ldots \exists x_9 \Big(\bigwedge_{i=2,4,6,8} E x_5 x_i \wedge \bigwedge_{i=1,3} E x_2 x_i \wedge \bigwedge_{i=1,7} E x_4 x_i \wedge \bigwedge_{i=3,9} E x_6 x_i \wedge \bigwedge_{i=7,9} E x_8 x_i \Big)$$

Primitive Positive Logic | Minimization

Example (Grid)

$$\gamma = \exists x_1 \ldots \exists x_9 \Big($$  $$\Big)$$

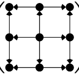
$$= \exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} E x_5 x_i \wedge \bigwedge_{i=1,3} E x_2 x_i \wedge \bigwedge_{i=1,7} E x_4 x_i \wedge \bigwedge_{i=3,9} E x_6 x_i \wedge \bigwedge_{i=7,9} E x_8 x_i)$$

$\equiv$ ... best possible equivalence preserving syntactic rewriting ...

Primitive Positive Logic | Minimization

Example (Grid)

$$\gamma = \exists x_1 \ldots \exists x_9 \Bigg(\quad \Bigg)$$



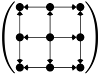
$$= \exists x_1 \ldots \exists x_9 \Big(\bigwedge_{i=2,4,6,8} E x_5 x_i \wedge \bigwedge_{i=1,3} E x_2 x_i \wedge \bigwedge_{i=1,7} E x_4 x_i \wedge \bigwedge_{i=3,9} E x_6 x_i \wedge \bigwedge_{i=7,9} E x_8 x_i \Big)$$

$\equiv \ldots$ best possible equivalence preserving syntactic rewriting $\ldots$

$$\equiv \exists x_1 \exists x_2 \exists x_3 \exists x_4 (E x_4 x_1 \wedge E x_2 x_1 \wedge E x_2 x_3$$
$$\wedge \exists x_1 (E x_1 x_4 \wedge E x_1 x_2 \wedge E x_2 x_3$$
$$\wedge \exists x_2 (E x_1 x_4 \wedge E x_1 x_2 \wedge E x_2 x_3$$
$$\wedge \exists x_3 (E x_4 x_3 \wedge E x_1 x_4 \wedge E x_1 x_2$$
$$\wedge \exists x_4 (E x_4 x_3 \wedge E x_1 x_4 \wedge E x_1 x_2$$
$$\wedge \exists x_1 (E x_4 x_3 \wedge E x_1 x_4 \wedge E x_1 x_2)))))) \in \mathrm{PP}^4.$$

Primitive Positive Logic | Minimization

Example (Grid)

$\gamma = \exists x_1 \dots \exists x_9 \Big($  $\Big)$

$$= \exists x_1 \dots \exists x_9 \Big(\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i \Big)$$

$\equiv$ ... best possible equivalence preserving syntactic rewriting ...

$$\begin{aligned}
&\equiv \exists x_1 \exists x_2 \exists x_3 \exists x_4 (Ex_4x_1 \wedge Ex_2x_1 \wedge Ex_2x_3 \\
&\qquad \wedge \exists x_1 (Ex_1x_4 \wedge Ex_1x_2 \wedge Ex_2x_3 \\
&\qquad \wedge \exists x_2 (Ex_1x_4 \wedge Ex_1x_2 \wedge Ex_2x_3 \\
&\qquad \wedge \exists x_3 (Ex_4x_3 \wedge Ex_1x_4 \wedge Ex_1x_2 \\
&\qquad \wedge \exists x_4 (Ex_4x_3 \wedge Ex_1x_4 \wedge Ex_1x_2 \\
&\qquad \wedge \exists x_1 (Ex_4x_3 \wedge Ex_1x_4 \wedge Ex_1x_2)))))) \in \mathrm{PP}^4.
\end{aligned}$$

$\gamma \in \mathrm{PP}^4$-EXPRESS, but $\gamma \in \mathrm{PP}^2$-EXPRESS (claim), ie, the result is *not* optimal.

Primitive Positive Logic | Minimization

Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

Primitive Positive Logic | Minimization

Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

The right approach would instead proceed through the following two steps.

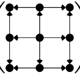
Primitive Positive Logic | Minimization

Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

The right approach would instead proceed through the following two steps.

Semantic Minimization (Hard): Observe in "modest complexity" that

$$\gamma = \exists x_1 \ldots \exists x_9 \left(\right. $$  $$ \left. \right) \equiv \exists x_1 \exists x_2 \exists x_3 \left(\right. $$  $$ \left. \right). $$

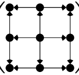
Primitive Positive Logic | Minimization

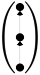
Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

The right approach would instead proceed through the following two steps.

Semantic Minimization (Hard): Observe in "modest complexity" that

$\gamma = \exists x_1 \ldots \exists x_9$ () $\equiv \exists x_1 \exists x_2 \exists x_3$ ().

Syntactic Minimization (Easy): Minimize the variables in $\exists x_1 \exists x_2 \exists x_3$ ()

by "best possible" syntactic rewriting.

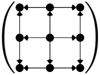
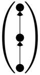
Primitive Positive Logic | Minimization

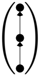
Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

The right approach would instead proceed through the following two steps.

Semantic Minimization (Hard): Observe in "modest complexity" that

$\gamma = \exists x_1 \ldots \exists x_9$  $\equiv \exists x_1 \exists x_2 \exists x_3$ .

Syntactic Minimization (Easy): Minimize the variables in $\exists x_1 \exists x_2 \exists x_3$ 

by "best possible" syntactic rewriting.

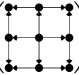
Primitive Positive Logic | Minimization

Example (Grid, Cont'd)

Syntactic rewriting is not a complete method for the minimization problem.

The right approach would instead proceed through the following two steps.

Semantic Minimization (Hard): Observe in "modest complexity" that

$$\gamma = \exists x_1 \ldots \exists x_9 \Bigg($$  $$\Bigg) \underset{\text{“NP away”}}{\equiv} \exists x_1 \exists x_2 \exists x_3 \Bigg($$  $$\Bigg).$$

Syntactic Minimization (Easy): Minimize the variables in $\exists x_1 \exists x_2 \exists x_3 \Bigg($  $\Bigg)$

by optimal polytime syntactic rewriting.

Primitive Positive Logic | Treewidth

Two pieces of theory from graph combinatorics:

Primitive Positive Logic | Treewidth

Two pieces of theory from graph combinatorics:

1. *cores* lead the semantic (hard) step;

Primitive Positive Logic | Treewidth

Two pieces of theory from graph combinatorics:

1. *cores* lead the semantic (hard) step;
2. *treewidth* leads the syntactic (easy) step.

Primitive Positive Logic | Treewidth

Two pieces of theory from graph combinatorics:

1. *cores* lead the semantic (hard) step;
2. *treewidth* leads the syntactic (easy) step.

Let $\mathbf{A}$ be a σ-structure. The *treewidth* $\mathrm{tw}(\mathbf{A})$ of $\mathbf{A}$ is a number $w \geq 1$ "measuring the similarity of $\mathbf{A}$ with a tree".

Primitive Positive Logic | Treewidth

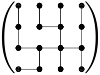
Two pieces of theory from graph combinatorics:

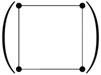
1. *cores* lead the semantic (hard) step;
2. *treewidth* leads the syntactic (easy) step.

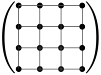
Let $\mathbf{A}$ be a σ-structure. The *treewidth* $\text{tw}(\mathbf{A})$ of $\mathbf{A}$ is a number $w \geq 1$ "measuring the similarity of $\mathbf{A}$ with a tree".

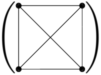
Low treewidth indicates high similarity with trees.

Example (Treewidth)

tw  $= 1$

tw  $= 2$

tw  $= 4$

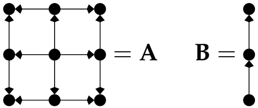
tw  $= 3$

$\text{tw}(\text{a tree}) = 1$ $\text{tw}(\text{a cycle}) = 2$ $\text{tw}(\text{a } k\text{-grid}) = k$ $\text{tw}(\text{a } k\text{-clique}) = k - 1$

Primitive Positive Logic | Homomorphisms

Let **A** and **B** be σ-structures. A *homomorphism* from **A** to **B** is a mapping $h\colon A \to B$ such that for all $R \in \sigma$ and all $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)}$, if $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$.

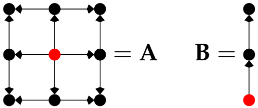
Example



Primitive Positive Logic | Homomorphisms

Let **A** and **B** be σ-structures. A *homomorphism* from **A** to **B** is a mapping $h\colon A \to B$ such that for all $R \in \sigma$ and all $(a_1, \dots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)}$, if $(a_1, \dots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, then $(h(a_1), \dots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$.

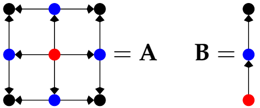
Example



Primitive Positive Logic | Homomorphisms

Let **A** and **B** be σ-structures. A *homomorphism* from **A** to **B** is a mapping $h\colon A \to B$ such that for all $R \in \sigma$ and all $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)}$, if $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$.

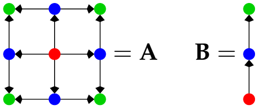
Example



Primitive Positive Logic | Homomorphisms

Let **A** and **B** be σ-structures. A *homomorphism* from **A** to **B** is a mapping $h\colon A \to B$ such that for all $R \in \sigma$ and all $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)}$, if $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$.

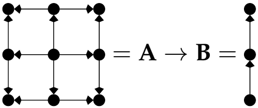
Example



Primitive Positive Logic | Homomorphisms

Let **A** and **B** be σ-structures. A *homomorphism* from **A** to **B** is a mapping $h\colon A \to B$ such that for all $R \in \sigma$ and all $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)}$, if $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$.

Example



Primitive Positive Logic | Cores

Let **A** and **B** be σ-structures.

B is a *core* iff every homomorphism $\mathbf{B} \to \mathbf{B}$ is bijective
(ie, every endomorphism of **B** is an automorphism).

Example

$= \mathbf{B}$

$= \mathbf{B}'$

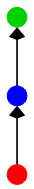
Primitive Positive Logic | Cores

Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures.

$\mathbf{B}$ is a *core* iff every homomorphism $\mathbf{B} \to \mathbf{B}$ is bijective
(ie, every endomorphism of $\mathbf{B}$ is an automorphism).

$\mathbf{B}$ is a *core of* $\mathbf{A}$ if (i) $\mathbf{B}$ is a core, (ii) $\mathbf{B}$ is a substructure of $\mathbf{A}$, (iii) $\mathbf{B} \leftrightarrow \mathbf{A}$.
Every finite structure $\mathbf{A}$ has a unique core up to isomorphism, $\text{core}(\mathbf{A})$.

Example

 $= \mathbf{B}$

 $= \mathbf{B}'$

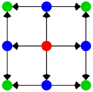
Primitive Positive Logic | Cores

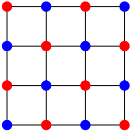
Let **A** and **B** be σ-structures.

B is a *core* iff every homomorphism **B** $\to$ **B** is bijective
(ie, every endomorphism of **B** is an automorphism).

B is a *core of* **A** if (i) **B** is a core, (ii) **B** is a substructure of **A**, (iii) **B** $\leftrightarrow$ **A**.
Every finite structure **A** has a unique core up to isomorphism, core(**A**).

Example

A =  $\leftrightarrow$  = **B** = core(**A**)

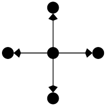
A$'$ =  $\leftrightarrow$  = **B**$'$ = core(**A**$'$)

Primitive Positive Logic | Notation

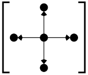
Natural correspondence between PP-sentences and relational structures.

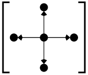
Example (Canonical Structure of a Query)

$$\mathbf{C}[\exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 (Ex_3x_1 \wedge Ex_3x_2 \wedge Ex_3x_4 \wedge Ex_3x_5)] =$$



Example (Canonical Query of a Structure)

F [] $= Ex_3x_1 \wedge Ex_3x_2 \wedge Ex_3x_4 \wedge Ex_3x_5$

Q [] $= \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 (Ex_3x_1 \wedge Ex_3x_2 \wedge Ex_3x_4 \wedge Ex_3x_5)$

Primitive Positive Logic | Characterization

A combinatorial characterization of k-variable expressibility for PP-logic.

Theorem (Dalmau et al.)
Let $\phi \in \mathrm{PP}_\sigma$. *Then,* $\phi \in \mathrm{PP}^k$-Express *if and only if* $\mathrm{tw}(\mathrm{core}(\mathbf{C}[\phi])) < k$.

Primitive Positive Logic | Grid Revisited

Example (Grid Revisited)

$$\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5 x_i \wedge \bigwedge_{i=1,3} Ex_2 x_i \wedge \bigwedge_{i=1,7} Ex_4 x_i \wedge \bigwedge_{i=3,9} Ex_6 x_i \wedge \bigwedge_{i=7,9} Ex_8 x_i) \in \text{PP}^2\text{-Express}$$

Primitive Positive Logic | Grid Revisited

Example (Grid Revisited)

$$\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i) \in \mathrm{PP}^2\text{-Express}$$

$$\Updownarrow$$

$$\mathrm{tw}(\mathrm{core}(\mathbf{C}[\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i)])) < 2$$

Primitive Positive Logic | Grid Revisited

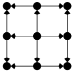
Example (Grid Revisited)

$$\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i) \in \mathrm{PP}^2\text{-}\mathrm{EXPRESS}$$

$$\Updownarrow$$

$$\mathrm{tw}(\mathrm{core}(\mathbf{C}[\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i)])) < 2$$

$$\Updownarrow$$

tw (core ()) < 2

Primitive Positive Logic | Grid Revisited

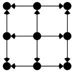
Example (Grid Revisited)

$$\exists x_1 \ldots \exists x_9 \big(\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i \big) \in \mathrm{PP}^2\text{-EXPRESS}$$

$$\Updownarrow$$

$$\mathrm{tw}(\mathrm{core}(\mathbf{C}[\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i)])) < 2$$

$$\Updownarrow$$

tw (core ()) < 2

$$\Updownarrow$$

$1 =$ tw () < 2

Primitive Positive Logic | Grid Revisited

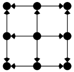
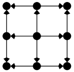
Example (Grid Revisited)

$$\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i) \in \mathrm{PP}^2\text{-EXPRESS}$$

$$\Updownarrow$$

$$\mathrm{tw}(\mathrm{core}(\mathbf{C}[\exists x_1 \ldots \exists x_9 (\bigwedge_{i=2,4,6,8} Ex_5x_i \wedge \bigwedge_{i=1,3} Ex_2x_i \wedge \bigwedge_{i=1,7} Ex_4x_i \wedge \bigwedge_{i=3,9} Ex_6x_i \wedge \bigwedge_{i=7,9} Ex_8x_i)])) < 2$$

$$\Updownarrow$$

tw (core ()) < 2 ≤ tw () = 3

$$\Updownarrow$$

1 = tw () < 2

Primitive Positive Logic | Classification

A complexity classification of k-variable expressibility for PP-logic.

Theorem (Dalmau et al.)
PP^k_σ-EXPRESS *is* NP-*complete for all* $k \geq 2$ *and all* $\sigma \supseteq \{E\}$.

Primitive Positive Logic | Classification

A complexity classification of k-variable expressibility for PP-logic.

Theorem (Dalmau et al.)
PP^k_σ-EXPRESS *is* NP-*complete for all* $k \geq 2$ *and all* $\sigma \supseteq \{E\}$.

Proof (Sketch).
The upper bound follows from the characterization. A reduction from the graph k-colorability problem gives the lower bound for $k \geq 3$ (extra work required for $k = 2$). □

Outline

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Example

An irredundant disjunctive form of ϕ is obtained as follows:

$$\phi$$

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Example

An irredundant disjunctive form of ϕ is obtained as follows:

$$\phi \equiv \exists x_1 \dots x_n \phi'$$

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Example

An irredundant disjunctive form of ϕ is obtained as follows:

$$\phi \equiv \exists x_1 \ldots x_n \phi' \equiv \exists x_1 \ldots x_n \bigvee_{i \in I} \phi'_i$$

where the ϕ'_i are PP-formulas,

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Example

An irredundant disjunctive form of ϕ is obtained as follows:

$$\phi \equiv \exists x_1 \ldots x_n \phi' \equiv \exists x_1 \ldots x_n \bigvee_{i \in I} \phi'_i \equiv \bigvee_{i \in I} \exists x_1 \ldots x_n \phi'_i$$

where the ϕ'_i are PP-formulas,

Existential Positive Logic | Implicants

Let $\phi \in$ EP (recall EP is the class of existential positive sentences).

A PP-sentence τ is an *implicant* of ϕ if $\tau \models \phi$.

A *disjunctive form* of ϕ is a disjunction of implicants of ϕ equivalent to ϕ.

A disjunctive form of ϕ is *irredundant* if,
for all two distinct implicants τ and τ' in the disjunction,
$\tau \not\models \tau'$ and $\tau' \not\models \tau$.

Example

An irredundant disjunctive form of ϕ is obtained as follows:

$$\phi \equiv \exists x_1 \ldots x_n \phi' \equiv \exists x_1 \ldots x_n \bigvee_{i \in I} \phi'_i \equiv \bigvee_{i \in I} \exists x_1 \ldots x_n \phi'_i \equiv \bigvee_{j \in J} \tau_j,$$

where the ϕ'_i are PP-formulas,
$\{\tau_j \mid j \in J\} \subseteq$ PP and $\tau_j \not\models \tau_{j'}$ for all $j, j' \in J, j \neq j'$.

Existential Positive Logic | Characterization

A combinatorial characterization of EP^k-Express.

Theorem (B, Chen)

Let $\phi \in \text{EP}_\sigma$. *Then,* $\phi \in \text{EP}^k_\sigma$-Express *if and only if* $\text{tw}(\text{core}(\mathbf{C}[\tau])) < k$, *for all implicants* τ *in an irredundant disjunctive form of* ϕ.

Existential Positive Logic | Characterization

A combinatorial characterization of EP^k-Express.

Theorem (B, Chen)

Let $\phi \in \mathrm{EP}_\sigma$*. Then,* $\phi \in \mathrm{EP}^k_\sigma$-Express *if and only if* $\mathrm{tw}(\mathrm{core}(\mathbf{C}[\tau])) < k$*, for all implicants* τ *in an irredundant disjunctive form of* ϕ*.*

Proof (Sketch).

Combine the combinatorial characterization of k-expressibility in PP-logic and the following combinatorial characterization of equivalence in EP-logic: If $\phi' = \bigvee_{i \in [m]} \phi'_i$ and $\phi'' = \bigvee_{j \in [n]} \phi''_j$ are irredundant disjunctive forms of $\phi \in \mathrm{EP}$, then there exists a bijection $\pi \colon [m] \to [n]$ such that, for all $i \in [m]$, $\mathbf{C}[\phi'_i] \leftrightarrow \mathbf{C}[\phi''_{\pi(i)}]$. □

Existential Positive Logic | Classification

A complexity classification of EP^k-Express.

Theorem (B, Chen)

EP^k_σ-Express *is* Π^p_2*-complete for all* $k \geq 3$ *and all* $\sigma \supseteq \{U_n \mid n \in \mathbb{N}\} \cup \{E\}$*,*
and for all $k \geq 6$ *and all* $\sigma \supseteq \{E\}$*.*

Existential Positive Logic | Classification

A complexity classification of EP^k-EXPRESS.

Theorem (B, Chen)

EP^k_σ-EXPRESS *is* Π^p_2*-complete for all* $k \geq 3$ *and all* $\sigma \supseteq \{U_n \mid n \in \mathbb{N}\} \cup \{E\}$*, and for all* $k \geq 6$ *and all* $\sigma \supseteq \{E\}$*.*

Proof (Sketch).

The upper bound follows from the characterization. A reduction from a Π^p_2-complete quantified version of the graph k-colorability problem gives the lower bound for all $k \geq 3$ (extra work required if $\sigma = \{E\}$). □

Existential Positive Logic | Reduction

Let $\mathbf{K}_k = ([k], [k]^2 \setminus \{(i,i) \mid i \in [k]\})$.

Existential Positive Logic | Reduction

Let $\mathbf{K}_k = ([k], [k]^2 \setminus \{(i,i) \mid i \in [k]\})$.

Reduction from the following Π_2^p-complete problem:

Problem Π_2-k-COLORABILITY

Instance $\psi = \forall y_1 \ldots \forall y_m \exists x_1 \ldots \exists x_n F[\mathbf{G}]$, where $\mathbf{G}$ is a graph and $G = \{y_1, \ldots, y_m, x_1, \ldots, x_n\} \cap [k] = \emptyset$.

Question $\mathbf{K}_k \models \psi$?

Note,
$\mathbf{K}_k \models \psi$
$\iff \mathbf{K}_k, f \models \exists x_1 \ldots \exists x_n F[\mathbf{G}]$ for all $f\colon \{y_1, \ldots, y_m\} \to [k]$
$\iff$ each partial k-coloring $f\colon \{y_1, \ldots, y_m\} \to [k]$ extends to a k-coloring of $\mathbf{G}$.

Existential Positive Logic | Reduction

$\psi = \forall y_1 \ldots \forall y_m \exists x_1 \ldots \exists x_n F[\mathbf{G}]$.

Reduction maps ψ to χ in $\mathrm{EP}_{\{E, U_1, \ldots, U_k, U_{y_1}, \ldots, U_{y_m}\}}$ defined as follows:

Existential Positive Logic | Reduction

$\psi = \forall y_1 \ldots \forall y_m \exists x_1 \ldots \exists x_n F[\mathbf{G}]$.

Reduction maps ψ to χ in $\mathrm{EP}_{\{E, U_1, \ldots, U_k, U_{y_1}, \ldots, U_{y_m}\}}$ defined as follows:

- $\mathrm{matrix}(\chi) = F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}]$;
- $\chi = \exists 1 \ldots \exists k \exists y_1 \ldots \exists y_m \exists x_1 \ldots \exists x_n \mathrm{matrix}(\chi)$,

where

Existential Positive Logic | Reduction

$\psi = \forall y_1 \ldots \forall y_m \exists x_1 \ldots \exists x_n F[\mathbf{G}]$.

Reduction maps ψ to χ in $\mathrm{EP}_{\{E, U_1, \ldots, U_k, U_{y_1}, \ldots, U_{y_m}\}}$ defined as follows:

- $\mathrm{matrix}(\chi) = F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}]$;
- $\chi = \exists 1 \ldots \exists k \exists y_1 \ldots \exists y_m \exists x_1 \ldots \exists x_n \mathrm{matrix}(\chi)$,

where

- $\mathbf{K}_k^k = \mathbf{C}[F[\mathbf{K}_k] \wedge U_1 1 \wedge \cdots \wedge U_k k]$;
- $\mathbf{L}_{y_i \mapsto j} = \mathbf{C}[U_{y_i} j]$;
- $\mathbf{M}_{y_i \mapsto j} = \mathbf{C}[\bigwedge_{c \in [k], c \neq j} (E y_i c \wedge E c y_i)]$.

Existential Positive Logic | Reduction

$\psi = \forall y_1 \ldots \forall y_m \exists x_1 \ldots \exists x_n F[\mathbf{G}]$.

Reduction maps ψ to χ in $\mathrm{EP}_{\{E, U_1, \ldots, U_k, U_{y_1}, \ldots, U_{y_m}\}}$ defined as follows:

- $\mathrm{matrix}(\chi) = F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}]$;
- $\chi = \exists 1 \ldots \exists k \exists y_1 \ldots \exists y_m \exists x_1 \ldots \exists x_n \mathrm{matrix}(\chi)$,

where

- $\mathbf{K}_k^k = \mathbf{C}[F[\mathbf{K}_k] \wedge U_1 1 \wedge \cdots \wedge U_k k]$;
- $\mathbf{L}_{y_i \mapsto j} = \mathbf{C}[U_{y_i} j]$;
- $\mathbf{M}_{y_i \mapsto j} = \mathbf{C}[\bigwedge_{c \in [k], c \neq j} (E y_i c \wedge E c y_i)]$.

Claim

$\mathbf{K}_k \models \psi \iff \chi \in \mathrm{EP}^k$-EXPRESS.

Existential Positive Logic | Reduction

$$\text{matrix}(\chi) = F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}]$$

Existential Positive Logic | Reduction

By distributing $\wedge$ over $\vee$,

$$\begin{aligned} \mathrm{matrix}(\chi) &= F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}] \\ &\equiv \bigvee_{f\colon \{y_1, \ldots, y_m\} \to [k]} F[\mathbf{G} \cup \mathbf{K}_k^k \cup \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})] \end{aligned}$$

Existential Positive Logic | Reduction

By distributing $\wedge$ over $\vee$,

$$\begin{aligned} \mathrm{matrix}(\chi) &= F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}] \\ &\equiv \bigvee_{f\colon \{y_1, \ldots, y_m\} \to [k]} F[\mathbf{G} \cup \mathbf{K}_k^k \cup \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})] \\ &= \bigvee_{f\colon \{y_1, \ldots, y_m\} \to [k]} F[\mathbf{H}_f]. \end{aligned}$$

Existential Positive Logic | Reduction

By distributing $\wedge$ over $\vee$,

$$\begin{aligned}\mathrm{matrix}(\chi) &= F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}] \\ &\equiv \bigvee_{f\colon \{y_1,\ldots,y_m\} \to [k]} F[\mathbf{G} \cup \mathbf{K}_k^k \cup \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})] \\ &= \bigvee_{f\colon \{y_1,\ldots,y_m\} \to [k]} F[\mathbf{H}_f].\end{aligned}$$

By the caracterization, suffices to show the following:

1. The disjunctive form $\bigvee_{f\colon \{y_1,\ldots,y_m\} \to [k]} Q[\mathbf{H}_f]$ of χ is irredundant.

Existential Positive Logic | Reduction

By distributing $\wedge$ over $\vee$,

$$\begin{aligned}\mathrm{matrix}(\chi) &= F[\mathbf{G} \cup \mathbf{K}_k^k] \wedge \bigwedge_{i \in [m]} \bigvee_{j \in [k]} F[\mathbf{L}_{y_i \mapsto j} \cup \mathbf{M}_{y_i \mapsto j}] \\ &\equiv \bigvee_{f\colon \{y_1,\dots,y_m\} \to [k]} F[\mathbf{G} \cup \mathbf{K}_k^k \cup \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})] \\ &= \bigvee_{f\colon \{y_1,\dots,y_m\} \to [k]} F[\mathbf{H}_f].\end{aligned}$$

By the caracterization, suffices to show the following:

1. The disjunctive form $\bigvee_{f\colon \{y_1,\dots,y_m\} \to [k]} Q[\mathbf{H}_f]$ of χ is irredundant.
2. $\mathbf{K}_k, f \models \exists x_1 \dots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$,
   for all $f\colon \{y_1, \dots, y_m\} \to [k]$.

Existential Positive Logic | Idea Item 1

The disjunctive form $\bigvee_{f\colon \{y_1,\ldots,y_m\}\to[k]} Q[\mathbf{H}_f]$ of χ is irredundant.

Example ($k = 3$, $m = 2$)

$f(y_1) = f'(y_1) = f'(y_2) = 2, f(y_2) = 1.$

$\mathbf{H}_f$ $\mathbf{H}_{f'}$

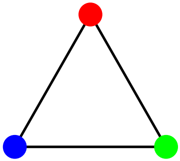
Existential Positive Logic | *Idea Item* 1

The disjunctive form $\bigvee_{f:\ \{y_1,\ldots,y_m\}\to[k]} Q[\mathbf{H}_f]$ of χ is irredundant.

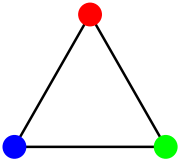
Example ($k = 3$, $m = 2$)

$f(y_1) = f'(y_1) = f'(y_2) = 2, f(y_2) = 1.$

$U_1^{\mathbf{H}_f} = U_1^{\mathbf{H}_{f'}} = \{\bullet\}, U_2^{\mathbf{H}_f} = U_2^{\mathbf{H}_{f'}} = \{\bullet\}, U_3^{\mathbf{H}_f} = U_3^{\mathbf{H}_{f'}} = \{\bullet\}.$



$\mathbf{H}_f$



$\mathbf{H}_{f'}$

Existential Positive Logic | *Idea Item* 1

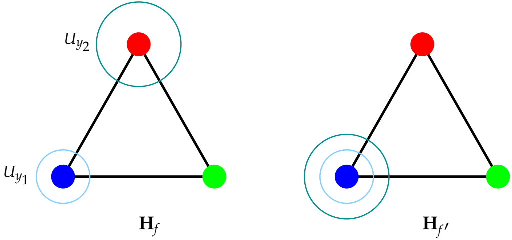
The disjunctive form $\bigvee_{f\colon \{y_1,\ldots,y_m\}\to[k]} Q[\mathbf{H}_f]$ of χ is irredundant.

Example ($k = 3$, $m = 2$)

$f(y_1) = f'(y_1) = f'(y_2) = 2, f(y_2) = 1.$

$U_1^{\mathbf{H}_f} = U_1^{\mathbf{H}_{f'}} = \{\bullet\}, U_2^{\mathbf{H}_f} = U_2^{\mathbf{H}_{f'}} = \{\bullet\}, U_3^{\mathbf{H}_f} = U_3^{\mathbf{H}_{f'}} = \{\bullet\}.$

$U_{y_1}^{\mathbf{H}_f} = U_{y_1}^{\mathbf{H}_{f'}} = U_{y_2}^{\mathbf{H}_{f'}} = \{\bullet\}, U_{y_2}^{\mathbf{H}_f} = \{\bullet\}.$

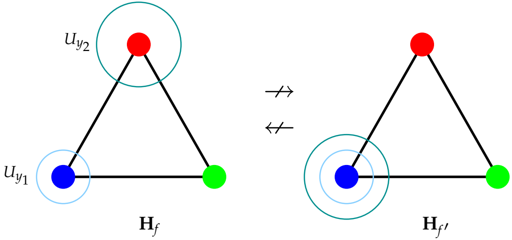


Existential Positive Logic | *Idea Item* 1

The disjunctive form $\bigvee_{f\colon \{y_1,\dots,y_m\}\to[k]} Q[\mathbf{H}_f]$ of χ is irredundant.

Example ($k = 3$, $m = 2$)

$f(y_1) = f'(y_1) = f'(y_2) = 2, f(y_2) = 1.$
$U_1^{\mathbf{H}_f} = U_1^{\mathbf{H}_{f'}} = \{\textcolor{red}{\bullet}\}, U_2^{\mathbf{H}_f} = U_2^{\mathbf{H}_{f'}} = \{\textcolor{blue}{\bullet}\}, U_3^{\mathbf{H}_f} = U_3^{\mathbf{H}_{f'}} = \{\textcolor{green}{\bullet}\}.$
$U_{y_1}^{\mathbf{H}_f} = U_{y_1}^{\mathbf{H}_{f'}} = U_{y_2}^{\mathbf{H}_{f'}} = \{\textcolor{blue}{\bullet}\}, U_{y_2}^{\mathbf{H}_f} = \{\textcolor{red}{\bullet}\}.$
$\mathbf{H}_f \not\to \mathbf{H}_{f'}$ and $\mathbf{H}_{f'} \not\to \mathbf{H}_f$, ie, $Q[\mathbf{H}_f] \not\models Q[\mathbf{H}_{f'}]$ and $Q[\mathbf{H}_{f'}] \not\models Q[\mathbf{H}_f]$.



Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k.$

Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.

$(\Longrightarrow)$ $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k.$

$(\Longrightarrow)$ $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

$(\Longleftarrow)$ Assume $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.
Then $\mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left (Picture 1).

Primitive Positive Logic | *Picture* 1

$\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k \Longrightarrow \mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\mathrm{core}(\mathbf{H}_f)$ is k-colorable.

Primitive Positive Logic | *Picture* 1

$\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k \Longrightarrow \mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\mathrm{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

Primitive Positive Logic | Picture 1

$\text{tw}(\text{core}(\mathbf{H}_f)) < k \Longrightarrow \text{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\text{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

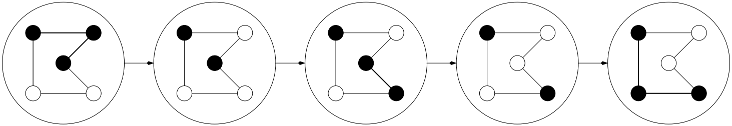
Is  3-colorable?

Primitive Positive Logic | *Picture* 1

$\text{tw}(\text{core}(\mathbf{H}_f)) < k \Longrightarrow \text{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\text{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

$\Downarrow$



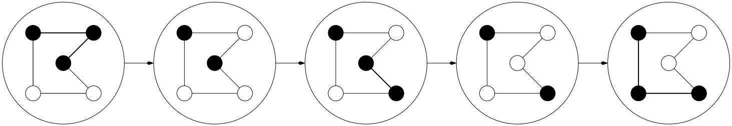
Is  3-colorable?

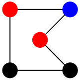
Primitive Positive Logic | *Picture* 1

$\text{tw}(\text{core}(\mathbf{H}_f)) < k \implies \text{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\text{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

$\Downarrow$





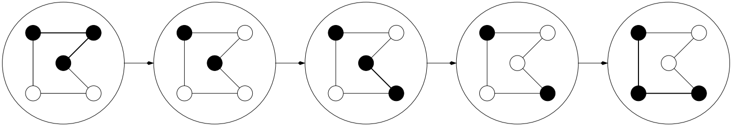
Is  3-colorable?

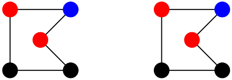
Primitive Positive Logic | *Picture* 1

$\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k \implies \mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\mathrm{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

$\Downarrow$





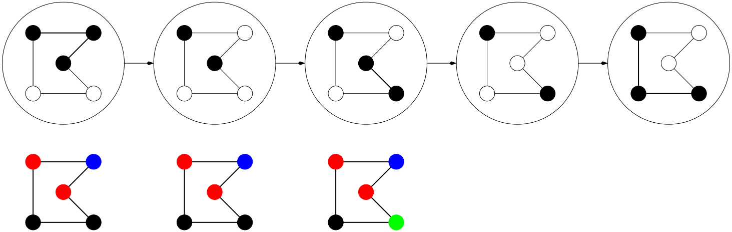
Is  3-colorable?

Primitive Positive Logic | *Picture* 1

$\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k \Longrightarrow \mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\mathrm{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

$\Downarrow$



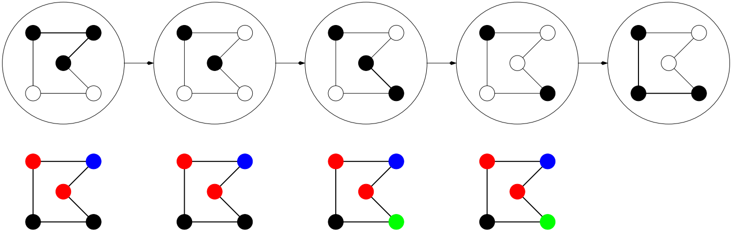
Is  3-colorable?

Primitive Positive Logic | *Picture* 1

$\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k \implies \mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\mathrm{core}(\mathbf{H}_f)$ is k-colorable.

tw() < 3

$\Downarrow$



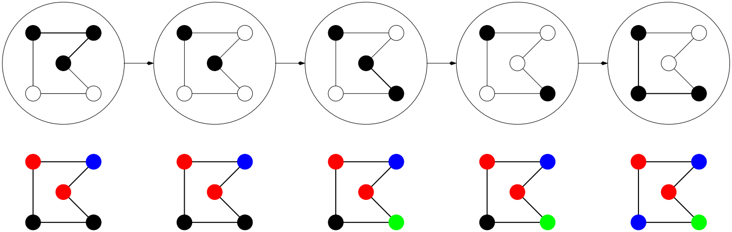
Is  3-colorable?

Primitive Positive Logic | *Picture* 1

$\text{tw}(\text{core}(\mathbf{H}_f)) < k \Longrightarrow \text{core}(\mathbf{H}_f) \to \mathbf{K}_k$, ie, $\text{core}(\mathbf{H}_f)$ is k-colorable.

tw () < 3

$\Downarrow$



$\Downarrow$

 is 3-colorable

Existential Positive Logic | *Sketch Item* 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k.$

($\Longrightarrow$) $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

($\Longleftarrow$) Assume $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.
Then $\mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left (Picture 1).

Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k.$

$(\Longrightarrow)$ $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

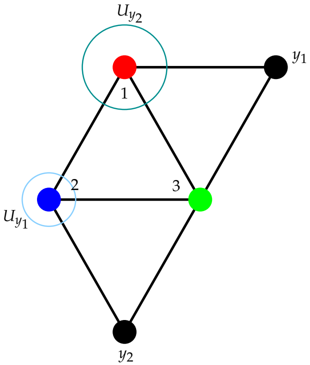
$(\Longleftarrow)$ Assume $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.
Then $\mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left (Picture 1).
We have $\mathbf{H}_f \to \mathrm{core}(\mathbf{H}_f)$, taking $\{E\}$-reducts.
Hence $\mathbf{H}_f \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left, say via map h.
Since $\mathbf{K}_k, h \models \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})$,
we have that h extends f (Picture 2).

Existential Positive Logic | *Picture* 2

$\mathbf{H}_f \to \mathbf{K}_k$ via $h \Longrightarrow h$ extends f.

Example ($k = 3$, $m = 2$)

$f(y_1) = 2$ and $f(y_2) = 1$. Thus $U_{y_1}^{\mathbf{H}_f} = \{2\}$, $U_{y_2}^{\mathbf{H}_f} = \{1\}$,
$E^{\mathbf{H}_f} \supseteq \{(y_1, 1), (y_1, 3), (1, y_1), (3, y_1), (y_2, 2), (y_2, 3), (2, y_2), (3, y_2)\}$.

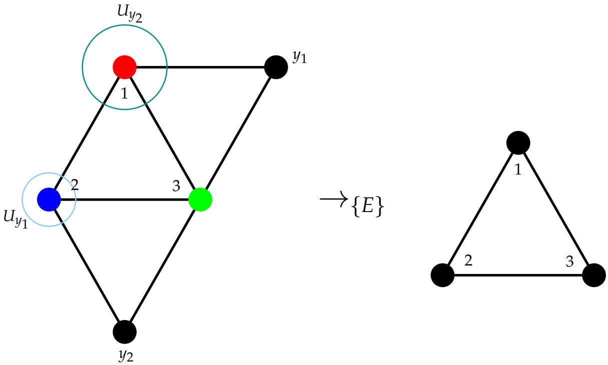


Existential Positive Logic | Picture 2

$\mathbf{H}_f \to \mathbf{K}_k$ via $h \Longrightarrow h$ extends f.

Example ($k = 3$, $m = 2$)

$f(y_1) = 2$ and $f(y_2) = 1$. Thus $U^{\mathbf{H}_f}_{y_1} = \{2\}$, $U^{\mathbf{H}_f}_{y_2} = \{1\}$,
$E^{\mathbf{H}_f} \supseteq \{(y_1, 1), (y_1, 3), (1, y_1), (3, y_1), (y_2, 2), (y_2, 3), (2, y_2), (3, y_2)\}$.
Assume $\mathbf{H}_f \to \mathbf{K}_k$ via h.

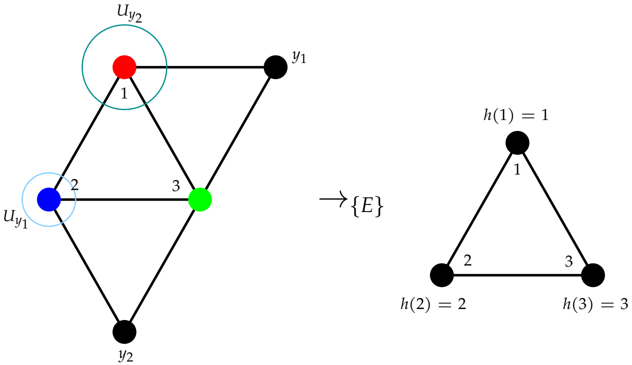


Existential Positive Logic | Picture 2

$\mathbf{H}_f \to \mathbf{K}_k$ via $h \Longrightarrow h$ extends f.

Example ($k = 3$, $m = 2$)

$f(y_1) = 2$ and $f(y_2) = 1$. Thus $U_{y_1}^{\mathbf{H}_f} = \{2\}$, $U_{y_2}^{\mathbf{H}_f} = \{1\}$,
$E^{\mathbf{H}_f} \supseteq \{(y_1, 1), (y_1, 3), (1, y_1), (3, y_1), (y_2, 2), (y_2, 3), (2, y_2), (3, y_2)\}$.
Assume $\mathbf{H}_f \to \mathbf{K}_k$ via h. Wlog, $h(i) = i$ for all $i \in [3]$.

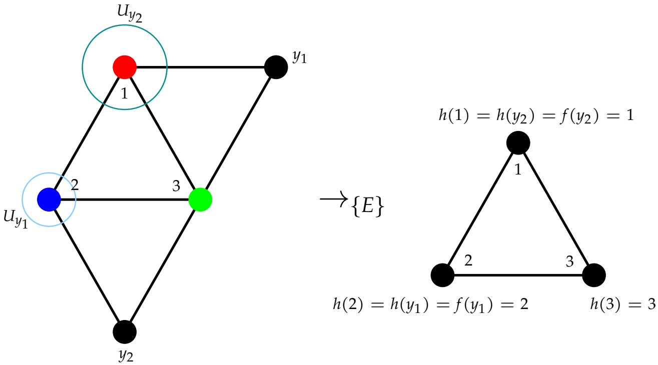


Existential Positive Logic | *Picture* 2

$\mathbf{H}_f \to \mathbf{K}_k$ via $h \Longrightarrow h$ extends f.

Example ($k = 3$, $m = 2$)

$f(y_1) = 2$ and $f(y_2) = 1$. Thus $U_{y_1}^{\mathbf{H}_f} = \{2\}$, $U_{y_2}^{\mathbf{H}_f} = \{1\}$,
$E^{\mathbf{H}_f} \supseteq \{(y_1, 1), (y_1, 3), (1, y_1), (3, y_1), (y_2, 2), (y_2, 3), (2, y_2), (3, y_2)\}$.
Assume $\mathbf{H}_f \to \mathbf{K}_k$ via h. Wlog, $h(i) = i$ for all $i \in [3]$.
Then, $h(y_j) = f(y_j)$ for all $j \in [2]$, ie, h extends f.



Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.

($\implies$) $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

($\impliedby$) Assume $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.
Then $\mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left (Picture 1).
We have $\mathbf{H}_f \to \mathrm{core}(\mathbf{H}_f)$, taking $\{E\}$-reducts.
Hence $\mathbf{H}_f \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left, say via map h.
Since $\mathbf{K}_k, h \models \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})$,
we have that h extends f (Picture 2).

Existential Positive Logic | Sketch Item 2

$\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}] \iff \mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.

($\Longrightarrow$) $\mathbf{K}_k, f \models \exists x_1 \ldots x_n F[\mathbf{G}]$, say $\mathbf{K}_k, h \models F[\mathbf{G}]$,
implies $\mathbf{H}_f \to \mathbf{K}_k^k$ via homomorphism acting as h on G and identically on $[k]$.
Thus, $\mathbf{H}_f \leftrightarrow \mathbf{K}_k^k$. Clearly $\mathbf{K}_k^k$ is a core.
Hence $\mathrm{core}(\mathbf{H}_f) = \mathbf{K}_k^k$.
Thus $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) = \mathrm{tw}(\mathbf{K}_k^k) = k - 1 < k$.

($\Longleftarrow$) Assume $\mathrm{tw}(\mathrm{core}(\mathbf{H}_f)) < k$.
Then $\mathrm{core}(\mathbf{H}_f) \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left (Picture 1).
We have $\mathbf{H}_f \to \mathrm{core}(\mathbf{H}_f)$, taking $\{E\}$-reducts.
Hence $\mathbf{H}_f \to \mathbf{K}_k$, taking the $\{E\}$-reduct on the left, say via map h.
Since $\mathbf{K}_k, h \models \bigcup_{i \in [m]} (\mathbf{L}_{y_i \mapsto f(y_i)} \cup \mathbf{M}_{y_i \mapsto f(y_i)})$,
we have that h extends f (Picture 2).
Moreover, $\mathbf{K}_k, h \models F[\mathbf{G}]$.

Outline

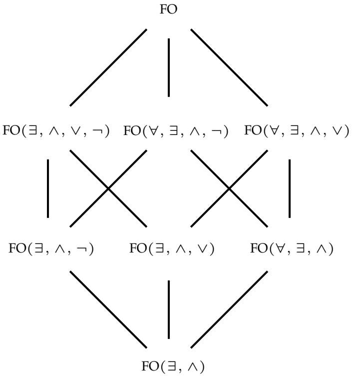
Research Motivation

Previous Work

Our Result

Other Results and Open Problems

First-Order Logic Fragments | Expressibility



S	FO(S)-EXPRESS
$\{\forall, \exists, \wedge, \vee, \neg\}$	undecidable [folklore]
$\{\forall, \exists, \wedge, \vee\}$	undecidable, $k \geq 3$ [B, Chen]
$\{\forall, \exists, \wedge, \neg\}$	open
$\{\exists, \wedge, \vee, \neg\}$	open
$\{\exists, \wedge, \neg\}$	open
$\{\forall, \exists, \wedge\}$	open
$\{\exists, \wedge, \vee\}$	Π_2^p-complete, $k \geq 3$ [B, Chen]
$\{\exists, \wedge\}$	NP-complete, $k \geq 2$ [Dalmau et al.]

FO(S) denotes equality-free relational FO-sentences in prefix negation form, using logical symbols in S.

First-Order Logic Fragments | Entailment and Equivalence

Understanding entailment/equivalence
helps in understanding expressibility.

As a byproduct, we obtained a (fairly complete) complexity classification of entailment/equivalence wrt:

- all existential fragments S of FO;
- all relational vocabularies σ;

thus refining known Π_2^p-completeness of $\mathrm{FO}_\sigma(\exists, \wedge, \neg)$ and $\mathrm{FO}_\sigma(\exists, \wedge, \vee)$.

σ	$\mathrm{FO}_\sigma(\exists, \wedge)$	$\mathrm{FO}_\sigma(\exists, \wedge, \neg)$	$\mathrm{FO}_\sigma(\exists, \wedge, \vee)$	$\mathrm{FO}_\sigma(\exists, \wedge, \vee, \neg)$
unary, $\lvert\sigma\rvert \leq 1$	P	P	P	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$
unary, finite, $\lvert\sigma\rvert > 1$	P	P	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$
unary infinite	P	P	Π_2^p-complete	Π_2^p-complete
$R \in \sigma$, $\mathrm{ar}(R) \geq 2$	NP-complete	Π_2^p-complete	Π_2^p-complete	Π_2^p-complete

First-Order Logic Fragments | Entailment and Equivalence

Understanding entailment/equivalence
helps in understanding expressibility.

As a byproduct, we obtained a (fairly complete) complexity classification of entailment/equivalence wrt:

- all existential fragments S of FO;
- all relational vocabularies σ;

thus refining known Π_2^p-completeness of $\mathrm{FO}_\sigma(\exists, \wedge, \neg)$ and $\mathrm{FO}_\sigma(\exists, \wedge, \vee)$.

σ	$\mathrm{FO}_\sigma(\exists, \wedge)$	$\mathrm{FO}_\sigma(\exists, \wedge, \neg)$	$\mathrm{FO}_\sigma(\exists, \wedge, \vee)$	$\mathrm{FO}_\sigma(\exists, \wedge, \vee, \neg)$
unary, $\lvert\sigma\rvert \leq 1$	P	P	P	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$
unary, finite, $\lvert\sigma\rvert > 1$	P	P	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$	coDP-hard, in $\mathrm{P}^{\mathrm{NP}[\mathrm{const}]}$
unary infinite	P	P	Π_2^p-complete	Π_2^p-complete
$R \in \sigma$, $\mathrm{ar}(R) \geq 2$	NP-complete	Π_2^p-complete	Π_2^p-complete	Π_2^p-complete

The complexity of $\mathrm{FO}_\sigma(\forall, \exists, \wedge)$ and $\mathrm{FO}_\sigma(\forall, \exists, \wedge, \neg)$ is open.

Literature

S. Bova and H. Chen.
The Complexity of Width Minimization for Existential Positive Logic.
Manuscript, 2013.

A. K. Chandra and P. M. Merlin.
Optimal Implementation of Conjunctive Queries in Relational Data Bases.
In *Proceedings of STOC'77*, pages 77–90, 1977.

H. Chen.
On the Complexity of Existential Positive Queries.
CoRR, abs/1206.3902, 2012.

V. Dalmau, P. G. Kolaitis, and M. Y. Vardi.
Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics.
In *Proceedings of CP'02*, 2002.

M. Grohe.
The Complexity of Homomorphism and Constraint Satisfaction Problems Seen from the Other Side.
Journal of the ACM, 54(1):1, 2007.

P. Hell and J. Nesetril.
The Core of a Graph.
Discrete Math., 109:117–126, 1992.

Y. Sagiv and M. Yannakakis.
Equivalences among Relational Expressions with the Union and Difference Operators.
Journal of the ACM, 27(4):633–655, 1980.

Thank you for your attention!