30 Years of Finite Model Theory in Finland

August 22, 2022

The Expressive Power of CSP Quantifiers

Lauri Hella

Faculty of Information and Communication Sciences,
Tampere University, Finland

Introduction

Cai, Fürer and Immerman 89 used a pebble game characterization for the infinitary k-variable logic with counting, $C^k_{\infty\omega}$ to prove that there are PTIME-computable properties that are not definable in $C^\omega_{\infty\omega} = \bigcup_{k\in\omega} C^k_{\infty\omega}$.

Introduction

Cai, Fürer and Immerman 89 used a pebble game characterization for the infinitary k-variable logic with counting, $C^k_{\infty\omega}$ to prove that there are PTIME-computable properties that are not definable in $C^\omega_{\infty\omega} = \bigcup_{k\in\omega} C^k_{\infty\omega}$.

In the LICS 92 paper "Logical hierarchies in PTIME", we introduced n-bijective k-pebble games that characterize equivalence with respect to $L^k_{\infty\omega}(\mathbf{Q}_n)$, the extension of k-variable logic with all n-ary quantifiers.

Using this game and a modification of the CFI construction, we proved an arity hierarchy theorem for PTIME: For every n there is a PTIME-computable quantifier of arity $n+1$ which is not definable in $L^\omega_{\infty\omega}(\mathbf{Q}_n)$.

In this talk we go through a new size hierarchy theorem for CSP quantifiers.

Here a generalized quantifier $Q_{\mathcal{K}}$ is a CSP quantifier if $\mathcal{K} = \mathsf{CSP}(\mathfrak{C})$ for some template $\mathfrak{C}$. The size of the quantifier $Q_{\mathsf{CSP}(\mathfrak{C})}$ is $|\mathrm{dom}(\mathfrak{C})|$.

In this talk we go through a new size hierarchy theorem for CSP quantifiers.

Here a generalized quantifier $Q_{\mathcal{K}}$ is a CSP quantifier if $\mathcal{K} = \mathrm{CSP}(\mathfrak{C})$ for some template $\mathfrak{C}$. The size of the quantifier $Q_{\mathrm{CSP}(\mathfrak{C})}$ is $|\mathrm{dom}(\mathfrak{C})|$.

We introduce CSP games that characterize equivalence with respect $L^k_{\infty\omega}(\mathbf{CSP}^+_n)$, where $\mathbf{CSP}^+_n$ is the union of $\mathbf{Q}_1$ and the class $\mathbf{CSP}_n$ of all CSP quantifiers of size at most n.

Using these games we prove that for every $n \geq 2$ there is a CSP quantifier of size $n + 1$ which is not definable in $L^\omega_{\infty\omega}(\mathbf{CSP}^+_n)$.

The proof is based on a new generalization of the CFI construction.

Constraint satisfaction problems

A homomorphism between two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a function $h : A \to B$ such that for every $R \in \tau$, and every $(a_1, \ldots, a_n) \in A^n$,

$$(a_1, \ldots, a_n) \in R^{\mathfrak{A}} \implies (h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}.$$

Constraint satisfaction problems

A homomorphism between two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a function $h : A \to B$ such that for every $R \in \tau$, and every $(a_1, \ldots, a_n) \in A^n$,

$$(a_1, \ldots, a_n) \in R^{\mathfrak{A}} \implies (h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}.$$

Every τ-structure $\mathfrak{C}$ gives rise to a constraint satisfaction problem:

- Given a τ-structure $\mathfrak{A}$, does there exist a homomorphism $h : \mathfrak{A} \to \mathfrak{C}$?

We denote the class of all positive instances $\mathfrak{A}$ of this problem by $\mathrm{CSP}(\mathfrak{C})$.

We classify template structures $\mathfrak{C}$ of CSP's by two numerical parameters:

- The arity of $\mathfrak{C}$ is $\mathrm{ar}(\mathfrak{C}) := \max\{\mathrm{ar}(R) \mid R \in \tau\}$.
- The size of $\mathfrak{C}$ is $\mathrm{sz}(\mathfrak{C}) := |\mathrm{dom}(\mathfrak{C})|$.

We classify template structures $\mathfrak{C}$ of CSP's by two numerical parameters:

- The arity of $\mathfrak{C}$ is $\mathrm{ar}(\mathfrak{C}) := \max\{\mathrm{ar}(R) \mid R \in \tau\}$.
- The size of $\mathfrak{C}$ is $\mathrm{sz}(\mathfrak{C}) := |\mathrm{dom}(\mathfrak{C})|$.

Example

(a) Clearly a graph G is n-colourable if and only if $G \in \mathrm{CSP}(\mathfrak{C}_{n\text{-COL}})$, where $\mathfrak{C}_{n\text{-COL}} := ([n], \{(i,j) \in [n]^2 \mid i \neq j\})$.

The arity and size of $\mathfrak{C}_{n\text{-COL}}$ are 2 and n, respectively.

We classify template structures $\mathfrak{C}$ of CSP's by two numerical parameters:

- The arity of $\mathfrak{C}$ is $\mathrm{ar}(\mathfrak{C}) := \max\{\mathrm{ar}(R) \mid R \in \tau\}$.
- The size of $\mathfrak{C}$ is $\mathrm{sz}(\mathfrak{C}) := |\mathrm{dom}(\mathfrak{C})|$.

Example

(a) Clearly a graph G is n-colourable if and only if $G \in \mathrm{CSP}(\mathfrak{C}_{n\text{-COL}})$, where $\mathfrak{C}_{n\text{-COL}} := ([n], \{(i,j) \in [n]^2 \mid i \neq j\})$.
The arity and size of $\mathfrak{C}_{n\text{-COL}}$ are 2 and n, respectively.

(b) The CFI structures constructed for the arity hierarchy theorem in the LICS 92 paper can be separated by $\mathrm{CSP}(\mathfrak{C}_{n\text{-CFI}})$, where $\mathfrak{C}_{n\text{-CFI}} = (\{0,1\}, R^{\mathrm{ev}})$ for

$$R^{\mathrm{ev}} := \{(b_1, \ldots, b_{n+1}) \mid b_1 + \cdots + b_{n+1} = 0 \mod 2\}.$$

The arity and size of $\mathfrak{C}_{n\text{-CFI}}$ are n and 2, respectively.

Generalized quantifiers

Let $\tau = \{R_1, \ldots, R_m\}$ be a relational vocabulary and let $\mathcal{K}$ be a class of finite τ-structures that is closed under isomorphisms.

The extension $L(Q_{\mathcal{K}})$ of a logic L by the quantifier $Q_{\mathcal{K}}$ is obtained by adding the following rules in the syntax and semantics of L:

Generalized quantifiers

Let $\tau = \{R_1, \ldots, R_m\}$ be a relational vocabulary and let $\mathcal{K}$ be a class of finite τ-structures that is closed under isomorphisms.

The extension $L(Q_\mathcal{K})$ of a logic L by the quantifier $Q_\mathcal{K}$ is obtained by adding the following rules in the syntax and semantics of L:

- If $\psi_1, \ldots, \psi_m$ are formulas and $\vec{y}_1, \ldots, \vec{y}_m$ are tuples of variables with $|\vec{y}_i| = \mathrm{ar}(R_i)$ for $i \in [m]$, then $\varphi = Q_\mathcal{K}\vec{y}_1, \ldots, \vec{y}_m\,(\psi_1, \ldots, \psi_m)$ is a formula.

Generalized quantifiers

Let $\tau = \{R_1, \dots, R_m\}$ be a relational vocabulary and let $\mathcal{K}$ be a class of finite τ-structures that is closed under isomorphisms.

The extension $L(Q_{\mathcal{K}})$ of a logic L by the quantifier $Q_{\mathcal{K}}$ is obtained by adding the following rules in the syntax and semantics of L:

- If $\psi_1, \dots, \psi_m$ are formulas and $\vec{y}_1, \dots, \vec{y}_m$ are tuples of variables with $|\vec{y}_i| = \mathrm{ar}(R_i)$ for $i \in [m]$, then $\varphi = Q_{\mathcal{K}}\vec{y}_1, \dots, \vec{y}_m\,(\psi_1, \dots, \psi_m)$ is a formula.
- $(\mathfrak{A}, \alpha) \models Q_{\mathcal{K}}\vec{y}_1, \dots, \vec{y}_m\,(\psi_1, \dots, \psi_m) \;\Leftrightarrow\; (A, \psi_1^{\mathfrak{A},\alpha,\vec{y}_1}, \dots, \psi_m^{\mathfrak{A},\alpha,\vec{y}_m}) \in \mathcal{K}$.

Here $\theta^{\mathfrak{A},\alpha,\vec{y}} := \{\vec{a} \in A^r \mid (\mathfrak{A}, \alpha[\vec{a}/\vec{y}]) \models \theta\}$ for a formula θ.

Generalized quantifiers

Let $\tau = \{R_1, \dots, R_m\}$ be a relational vocabulary and let $\mathcal{K}$ be a class of finite τ-structures that is closed under isomorphisms.

The extension $L(Q_\mathcal{K})$ of a logic L by the quantifier $Q_\mathcal{K}$ is obtained by adding the following rules in the syntax and semantics of L:

- If $\psi_1, \dots, \psi_m$ are formulas and $\vec{y}_1, \dots, \vec{y}_m$ are tuples of variables with $|\vec{y}_i| = \mathrm{ar}(R_i)$ for $i \in [m]$, then $\varphi = Q_\mathcal{K}\vec{y}_1, \dots, \vec{y}_m\,(\psi_1, \dots, \psi_m)$ is a formula.
- $(\mathfrak{A}, \alpha) \models Q_\mathcal{K}\vec{y}_1, \dots, \vec{y}_m\,(\psi_1, \dots, \psi_m) \;\Leftrightarrow\; (A, \psi_1^{\mathfrak{A},\alpha,\vec{y}_1}, \dots, \psi_m^{\mathfrak{A},\alpha,\vec{y}_m}) \in \mathcal{K}$.

Here $\theta^{\mathfrak{A},\alpha,\vec{y}} := \{\vec{a} \in A^r \mid (\mathfrak{A}, \alpha[\vec{a}/\vec{y}]) \models \theta\}$ for a formula θ.

The arity of $Q_\mathcal{K}$ is $\max\{\mathrm{ar}(R) \mid R \in \tau\}$. We denote the class of all quantifiers with arity at most m by $\mathbf{Q}_m$.

A quantifer $Q_{\mathcal{K}}$ is a CSP quantifier if $\mathcal{K} = \mathrm{CSP}(\mathfrak{C})$ for some template $\mathfrak{C}$. We denote the class of all CSP quantifiers $Q_{\mathrm{CSP}(\mathfrak{C})}$ such that $\mathrm{sz}(\mathfrak{C}) \leq n$ by $\mathbf{CSP}_n$. Furthermore, we write $\mathbf{CSP}_n^+ := \mathbf{CSP}_n \cup \mathbf{Q}_1$.

A quantifer $Q_{\mathcal{K}}$ is a CSP quantifier if $\mathcal{K} = \mathrm{CSP}(\mathfrak{C})$ for some template $\mathfrak{C}$. We denote the class of all CSP quantifiers $Q_{\mathrm{CSP}(\mathfrak{C})}$ such that $\mathrm{sz}(\mathfrak{C}) \leq n$ by $\mathbf{CSP}_n$. Furthermore, we write $\mathbf{CSP}_n^+ := \mathbf{CSP}_n \cup \mathbf{Q}_1$.

Example

(a) $\exists = Q_{\mathcal{K}_\exists}$, where $\mathcal{K}_\exists := \{(A, R) \mid R \subseteq A, R \neq \emptyset\}$ and $\forall = Q_{\mathcal{K}_\forall}$, where $\mathcal{K}_\forall := \{(A, R) \mid R = A\}$.

(b) Härtig quantifier: $I = Q_{\mathcal{K}_I}$, where $\mathcal{K}_I := \{(A, P, R) \mid |P| = |R|\}$.
If G is a graph, then $G \models \forall x \forall y\, I\, x, y(E(x, y), E(y, x))$ iff G is regular.

A quantifer $Q_{\mathcal{K}}$ is a CSP quantifier if $\mathcal{K} = \mathrm{CSP}(\mathfrak{C})$ for some template $\mathfrak{C}$. We denote the class of all CSP quantifiers $Q_{\mathrm{CSP}(\mathfrak{C})}$ such that $\mathrm{sz}(\mathfrak{C}) \leq n$ by $\mathbf{CSP}_n$. Furthermore, we write $\mathbf{CSP}_n^+ := \mathbf{CSP}_n \cup \mathbf{Q}_1$.

Example

(a) $\exists = Q_{\mathcal{K}_\exists}$, where $\mathcal{K}_\exists := \{(A, R) \mid R \subseteq A, R \neq \emptyset\}$ and $\forall = Q_{\mathcal{K}_\forall}$, where $\mathcal{K}_\forall := \{(A, R) \mid R = A\}$.

(b) Härtig quantifier: $I = Q_{\mathcal{K}_I}$, where $\mathcal{K}_I := \{(A, P, R) \mid |P| = |R|\}$.
If G is a graph, then $G \models \forall x \forall y\, I\, x, y(E(x, y), E(y, x))$ iff G is regular.

(c) $Q_{\mathfrak{C}_{n\text{-COL}}} \in \mathbf{Q}_2 \cap \mathbf{CSP}_n$ and $Q_{\mathfrak{C}_{n\text{-CFI}}} \in \mathbf{Q}_{n+1} \cap \mathbf{CSP}_2$ for all $n \geq 2$.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures of the same vocabulary, and α and β assignments on $\mathfrak{A}$ and $\mathfrak{B}$ such that $\mathrm{dom}(\alpha) = \mathrm{dom}(\beta)$.

We write $(\mathfrak{A}, \alpha) \equiv^{k}_{\infty\omega,n} (\mathfrak{B}, \beta)$ if the equivalence

$$(\mathfrak{A}, \alpha) \models \varphi \Leftrightarrow (\mathfrak{B}, \beta) \models \varphi$$

holds for all formulas $\varphi \in L^{k}_{\infty\omega}(\mathbf{CSP}^{+}_{n})$ with free variables in $\mathrm{dom}(\alpha)$.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures of the same vocabulary, and α and β assignments on $\mathfrak{A}$ and $\mathfrak{B}$ such that $\mathrm{dom}(\alpha) = \mathrm{dom}(\beta)$.

We write $(\mathfrak{A}, \alpha) \equiv^k_{\infty\omega,n} (\mathfrak{B}, \beta)$ if the equivalence

$$(\mathfrak{A}, \alpha) \models \varphi \Leftrightarrow (\mathfrak{B}, \beta) \models \varphi$$

holds for all formulas $\varphi \in L^k_{\infty\omega}(\mathbf{CSP}^+_n)$ with free variables in $\mathrm{dom}(\alpha)$.

Similarly we write $(\mathfrak{A}, \alpha) \equiv^k_n (\mathfrak{B}, \beta)$ if the equivalence above holds for all $\mathrm{FO}^k(\mathbf{CSP}^+_n)$-formulas φ. If $\alpha = \beta = \emptyset$, we write simply $\mathfrak{A} \equiv^k_{\infty\omega,n} \mathfrak{B}$ instead of $(\mathfrak{A}, \emptyset) \equiv^k_{\infty\omega,n} (\mathfrak{B}, \emptyset)$, and similarly for $\equiv^k_n$.

Pebble game for CSP quantifiers

Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures, and let α and β are assignments on $\mathfrak{A}$ and $\mathfrak{B}$ such that $\mathrm{dom}(\alpha) = \mathrm{dom}(\beta) \subseteq X_k := \{x_1, \ldots, x_k\}$. Furthermore, let $n, k \geq 1$ (n is a parameter for the size of templates and k for the number of pebbles).

Pebble game for CSP quantifiers

Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures, and let α and β are assignments on $\mathfrak{A}$ and $\mathfrak{B}$ such that $\mathrm{dom}(\alpha) = \mathrm{dom}(\beta) \subseteq X_k := \{x_1, \ldots, x_k\}$. Furthermore, let $n, k \geq 1$ (n is a parameter for the size of templates and k for the number of pebbles).

Definition

The game $\mathrm{CSPG}[\mathfrak{A}, \mathfrak{B}, n, k](\alpha, \beta)$ is played between S and D, and it has the following rules:

(1) If $\alpha \mapsto \beta \notin \mathrm{PI}(\mathfrak{A}, \mathfrak{B})$, then the game ends, and S wins.

(2) If (1) does not hold, there are three types of moves that S can choose to play: **bijection move**, **left/right CSP-quantifier move**.

(3) D wins the game if S does not win it in a finite number of rounds.

- **Bijection move:** - S starts by choosing a variable $y \in X_k$.
  - D answers by choosing a bijection $f : A \to B$.
  - S completes the round by choosing an element $a \in A$.
  - The players continue by playing $\mathrm{CSPG}(\alpha[a/y], \beta[f(a)/y])$.

- **Bijection move:** - S starts by choosing a variable $y \in X_k$.
  - D answers by choosing a bijection $f\colon A \to B$.
  - S completes the round by choosing an element $a \in A$.
  - The players continue by playing $\mathrm{CSPG}(\alpha[a/y], \beta[f(a)/y])$.
- **Left CSP-quantifier move:** - S starts by choosing $r \in [k]$ and an r-tuple $\vec{y} \in X_k^r$ of distinct variables and a colouring $g\colon A \to [n]$.
  - D chooses next a colouring $h\colon B \to [n]$.
  - S answers by choosing an r-tuple $\vec{b} \in B^r$.
  - D completes the round by choosing an r-tuple $\vec{a} \in A^r$ such that $g(a_j) = h(b_j)$ for all $j \in [r]$.
  - The players continue by playing $\mathrm{CSPG}(\alpha[\vec{a}/\vec{y}], \beta[\vec{b}/\vec{y}])$.

- **Bijection move:** - S starts by choosing a variable $y \in X_k$.
  - D answers by choosing a bijection $f\colon A \to B$.
  - S completes the round by choosing an element $a \in A$.
  - The players continue by playing $\mathrm{CSPG}(\alpha[a/y], \beta[f(a)/y])$.
- **Left CSP-quantifier move:** - S starts by choosing $r \in [k]$ and an r-tuple $\vec{y} \in X_k^r$ of distinct variables and a colouring $g\colon A \to [n]$.
  - D chooses next a colouring $h\colon B \to [n]$.
  - S answers by choosing an r-tuple $\vec{b} \in B^r$.
  - D completes the round by choosing an r-tuple $\vec{a} \in A^r$ such that $g(a_j) = h(b_j)$ for all $j \in [r]$.
  - The players continue by playing $\mathrm{CSPG}(\alpha[\vec{a}/\vec{y}], \beta[\vec{b}/\vec{y}])$.
- **Right CSP-quantifier move:** Switch the roles of $\mathfrak{A}$ and $\mathfrak{B}$.

Characterization theorem

The CSP game characterizes equivalence with respect to both of the logics $\mathrm{FO}^k(\mathbf{CSP}_n^+)$ and $L^k_{\infty\omega}(\mathbf{CSP}_n^+)$:

Theorem

The following conditions are equivalent:

- *D has a winning strategy in the game* $\mathrm{CSPG}[\mathfrak{A}, \mathfrak{B}, n, k](\alpha, \beta)$,
- $(\mathfrak{A}, \alpha) \equiv^k_{\infty\omega,n} (\mathfrak{B}, \beta)$,
- $(\mathfrak{A}, \alpha) \equiv^k_n (\mathfrak{B}, \beta)$.

Generalized CFI structures

Let $\tau_n = \{R_n\}$, where $\mathrm{ar}(R_n) = 3n$, and let $A_n = \{a_j^i \mid i \in [3], j \in [n+1]\}$.

We define gadget structures $\mathfrak{A}_n = (A_n, R_n)$ and $\tilde{\mathfrak{A}}_n = (A_n, \tilde{R}_n)$ such that switching a pair a_j^i and a_ℓ^i for any $i \in [3]$ and any $j \neq \ell \in [n+1]$ gives an isomorphism between $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$.

(Thus, switching two such pairs gives an automorphism of $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$.)

Generalized CFI structures

Let $\tau_n = \{R_n\}$, where $\mathrm{ar}(R_n) = 3n$, and let $A_n = \{a^i_j \mid i \in [3], j \in [n+1]\}$.

We define gadget structures $\mathfrak{A}_n = (A_n, R_n)$ and $\tilde{\mathfrak{A}}_n = (A_n, \tilde{R}_n)$ such that switching a pair a^i_j and a^i_ℓ for any $i \in [3]$ and any $j \neq \ell \in [n+1]$ gives an isomorphism between $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$.

(Thus, switching two such pairs gives an automorphism of $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$.)

Given an ordered 3-regular connected graph G, we define a τ_n-structure $\mathfrak{A}^{\mathrm{ev}}_n(G)$ by replacing the vertices of G by copies of $\mathfrak{A}_n$. The other CFI-structure $\mathfrak{A}^{\mathrm{od}}_n(G)$ is defined in the same way, except that for one vertex we use $\tilde{\mathfrak{A}}_n$.

Separating the CFI structures

Let $C_n = \{c_j \mid j \in [n+1]\}$ and let $h_n \colon A_n \to C_n$ be the projection $h_n(a^i_j) = c_j$. Defining $\mathcal{R}_n := \{h_n(\vec{a}) \mid \vec{a} \in R_n\}$ and $\mathfrak{C}_n := (C_n, \mathcal{R}_n)$, we see that h_n is a homomorphism $\mathfrak{A}_n \to \mathfrak{C}_n$.

Separating the CFI structures

Let $C_n = \{c_j \mid j \in [n+1]\}$ and let $h_n \colon A_n \to C_n$ be the projection $h_n(a^i_j) = c_j$. Defining $\mathcal{R}_n := \{h_n(\vec{a}) \mid \vec{a} \in R_n\}$ and $\mathfrak{C}_n := (C_n, \mathcal{R}_n)$, we see that h_n is a homomorphism $\mathfrak{A}_n \to \mathfrak{C}_n$.

Furthermore, the union of the copies of h_n on the copies of $\mathfrak{A}_n$ in $\mathfrak{A}^{\mathrm{ev}}_n(G)$ is a homomorphism $\mathfrak{A}^{\mathrm{ev}}_n(G) \to \mathfrak{C}_n$. Thus, $\mathfrak{A}^{\mathrm{ev}}_n(G) \in \mathrm{CSP}(\mathfrak{C}_n)$.

On the other hand, using a parity argument, we see that $\mathfrak{A}^{\mathrm{od}}_n(G) \notin \mathrm{CSP}(\mathfrak{C}_n)$.

Separating the CFI structures

Let $C_n = \{c_j \mid j \in [n+1]\}$ and let $h_n \colon A_n \to C_n$ be the projection $h_n(a^i_j) = c_j$. Defining $\mathcal{R}_n := \{h_n(\vec{a}) \mid \vec{a} \in R_n\}$ and $\mathfrak{C}_n := (C_n, \mathcal{R}_n)$, we see that h_n is a homomorphism $\mathfrak{A}_n \to \mathfrak{C}_n$.

Furthermore, the union of the copies of h_n on the copies of $\mathfrak{A}_n$ in $\mathfrak{A}^{\mathrm{ev}}_n(G)$ is a homomorphism $\mathfrak{A}^{\mathrm{ev}}_n(G) \to \mathfrak{C}_n$. Thus, $\mathfrak{A}^{\mathrm{ev}}_n(G) \in \mathrm{CSP}(\mathfrak{C}_n)$.

On the other hand, using a parity argument, we see that $\mathfrak{A}^{\mathrm{od}}_n(G) \notin \mathrm{CSP}(\mathfrak{C}_n)$.

Remark. $\mathrm{CSP}(\mathfrak{C}_n)$ is NP-complete, but $\mathfrak{A}^{\mathrm{ev}}_n(G)$ and $\mathfrak{A}^{\mathrm{od}}_n(G)$ can also be separated by PTIME properties.

Winning the CSP game on the CFI structures

Making use of the switching isomorphisms of the gadgets $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$, we prove that D has a winning strategy in $\mathrm{CSPG}[\mathfrak{A}_n^{\mathrm{ev}}(G), \mathfrak{A}_n^{\mathrm{od}}(G), n, k](\alpha, \beta)$ whenever she has one in BPG_1^k on the original CFI structures obtained from G.

Winning the CSP game on the CFI structures

Making use of the switching isomorphisms of the gadgets $\mathfrak{A}_n$ and $\tilde{\mathfrak{A}}_n$, we prove that D has a winning strategy in $\mathrm{CSPG}[\mathfrak{A}_n^{\mathrm{ev}}(G), \mathfrak{A}_n^{\mathrm{od}}(G), n, k](\alpha, \beta)$ whenever she has one in BPG_1^k on the original CFI structures obtained from G.

Theorem

$\mathrm{CSP}(\mathfrak{C}_n)$ *is not definable in* $L^{\omega}_{\infty\omega}(\mathbf{CSP}_n^+)$.

As a corollary, we get size hierarchy result for CSP quantifiers:

Corollary

For every $n \geq 2$ *there is a CSP quantifer of size* $n+1$ *which is not definable in* $L^{\omega}_{\infty\omega}(\mathbf{CSP}_n^+)$.

Open problems

1. The CSP-quantifiers $Q_{\mathfrak{C}_n}$ are NP-complete. Can they be replaced by some PTIME-computable CSP-quantifiers?

Open problems

1. The CSP-quantifiers $Q_{\mathfrak{C}_n}$ are NP-complete. Can they be replaced by some PTIME-computable CSP-quantifiers?
2. The arity of the CSP-quantifier $Q_{\mathfrak{C}_n}$ is $3n$. Are there templates $\mathfrak{D}_n$ such that $\mathrm{CSP}(\mathfrak{D}_n)$ is not definable in $L^{\omega}_{\infty\omega}(\mathbf{CSP}^{+}_{n})$ and $\mathrm{ar}(\mathfrak{D}_n) = r$ for some constant r?

Open problems

1. The CSP-quantifiers $Q_{\mathfrak{C}_n}$ are NP-complete. Can they be replaced by some PTIME-computable CSP-quantifiers?

2. The arity of the CSP-quantifier $Q_{\mathfrak{C}_n}$ is $3n$. Are there templates $\mathfrak{D}_n$ such that $\mathrm{CSP}(\mathfrak{D}_n)$ is not definable in $L^{\omega}_{\infty\omega}(\mathbf{CSP}^+_n)$ and $\mathrm{ar}(\mathfrak{D}_n) = r$ for some constant r?

3. It seems quite plausible that for any $\ell, n \geq 2$, equivalence with respect to the extension of $L^k_{\infty\omega}$ by all ℓth vectorizations of the quantifiers in $\mathbf{CSP}^+_n$ is just isomorphism for large enough k. Prove or disprove this!

Open problems

1. The CSP-quantifiers $Q_{\mathfrak{C}_n}$ are NP-complete. Can they be replaced by some PTIME-computable CSP-quantifiers?
2. The arity of the CSP-quantifier $Q_{\mathfrak{C}_n}$ is $3n$. Are there templates $\mathfrak{D}_n$ such that $\mathrm{CSP}(\mathfrak{D}_n)$ is not definable in $L^{\omega}_{\infty\omega}(\mathbf{CSP}^+_n)$ and $\mathrm{ar}(\mathfrak{D}_n) = r$ for some constant r?
3. It seems quite plausible that for any $\ell, n \geq 2$, equivalence with respect to the extension of $L^k_{\infty\omega}$ by all ℓth vectorizations of the quantifiers in $\mathbf{CSP}^+_n$ is just isomorphism for large enough k. Prove or disprove this!
4. What is the relationship between $L^{\omega}_{\infty\omega}(\mathbf{CSP}^+_n)$ and the extension $\mathrm{LA}^{\omega}(Q)$ of $L^{\omega}_{\infty\omega}$ with all linear algebraic operators (Dawar, Grädel and Pakusa 19)? Does equivalence with respect to one of these logics imply equivalence with respect to the other?

Happy 72nd birthday Phokion and Jouko!

Happy 72nd birthday Phokion and Jouko!

N.B: 72 is the smallest integer of the form $p^q \cdot q^p$ for distinct primes p, q.