Satisfiability with Agreement and Counting

Krasimir Georgiev

May 25, 2016

Contents

Glossary

$|A|$ the cardinality of A. 1
$\wp A$ the powerset of A. 1
$\wp^+ A$ the set of nonempty subsets of A. 1
$\wp^\kappa A$ the set of subsets of A of cardinality κ. 1
$A \times B$ the cartesian product of A and B. 1
$\operatorname{dom} R$ the domain of R. 1
$\operatorname{ran} R$ the range of R. 1
R^{-1} the inverse of R. 1
$R \restriction S$ the restriction of R to S. 1
$R[a]$ the R-successors of a. 1
$S \circ R$ the composition of S and R. 1
id_A the identity on A. 1
$f : A \to B$ a total function from A to B. 1
$f : A \hookrightarrow B$ an injective function from A into B. 1
$f : A \twoheadrightarrow B$ a surjective function from A onto B. 1
$f : A \leftrightarrow B$ a bijective function between A and B. 1
$f : A \rightsquigarrow B$ a partial function from A to B. 1
$f(a) \simeq b$ f is defined at a with value b. 1
$f(a) \simeq \perp$ f is not defined at a. 1
ch_S^A characteristic function. 1
$\|\mathrm{A}\|$ the length of A. 1
$\langle a, b, c\rangle$ a sequence. 1
ε the empty sequence. 1
$\mathrm{A} + \mathrm{B}$ the concatenation of A and B. 1
$\mathrm{A} - \mathrm{B}$ A without the elements of B. 1
$\mathbb{N}$ the natural numbers. 1
$\mathbb{N}^+$ the positive natural numbers. 1
$[n, m]$ the discrete interval between n and m. 1
log the base-2 logarithm. 2
$\mathrm{v} \prec \mathrm{w}$ lexicographically smaller. 2
$\mathbb{S}_n$ the set of permutations of $[1, n]$. 2
$\exp_a^e(x)$ tetration. 2
Ω an alphabet. 2
$w = w_1 w_2 \dots w_n$ a word. 2
Ω^* the set of words over Ω. 2
Ω^+ the set of nonempty words over Ω. 2
Ω^n the set of words of length n over Ω. 2
$\mathbb{B}$ the bits. 2
$\mathbb{B}^+$ the bitstrings. 2
$\|n\|$ the bitsize of n. 2
$\overline{n}$ the binary encoding of n. 2
$\underline{\mathrm{b}}$ the number encoded by b. 2
N_t the largest t-bit number. 2
$\mathbb{B}_t$ the t-bit numbers. 2
$\Omega_{\mathcal{C}}$ the symbol alphabet. 2
$\mathcal{V}$ the variable symbols. 3
$\boldsymbol{x}$ the first variable symbol. 3
$\boldsymbol{y}$ the second variable symbol. 3
$\boldsymbol{z}$ the third variable symbol. 3
Σ a predicate signature. 3
$\boldsymbol{p}_i$ a predicate symbol. 3
$\operatorname{ar} \boldsymbol{p}_i$ the arity of $\boldsymbol{p}_i$. 3
$\mathcal{A}t[\Sigma]$ the atomic formulas over Σ. 3
$\mathcal{L}it[\Sigma]$ the literals over Σ. 3
$\mathcal{C}[\Sigma]$ the first-order formulas with counting quantifiers over Σ. 3
$\mathcal{L}[\Sigma]$ the first-order formulas over Σ. 3
$\operatorname{vr} \varphi$ the variables occurring φ. 3
$\operatorname{fvr} \varphi$ the variables freely occurring φ. 3
$\mathcal{L}^v[\Sigma]$ the v-variable first-order formulas over Σ. 3
$\mathcal{C}^v[\Sigma]$ the v-variable first-order formulas with counting quantifiers over Σ. 3

$\operatorname{qr}\varphi$ the quantifier rank of φ. 3
$\mathcal{L}_r[\Sigma]$ the r-rank first-order formulas over Σ. 4
$\mathcal{C}_r[\Sigma]$ the r-rank first-order formulas with counting quantifiers over Σ. 4
$\mathcal{L}_r^v[\Sigma]$ the r-rank v-variable first-order formulas over Σ. 4
$\mathcal{C}_r^v[\Sigma]$ the r-rank v-variable first-order formulas with counting quantifiers over Σ. 4
$\mathfrak{A}$ a structure. 4
$\varphi^{\mathfrak{A}}$ interpretation of φ in $\mathfrak{A}$. 5
SAT$\mathcal{K}$ the satisfiable sentences of $\mathcal{K}$. 5
FINSAT$\mathcal{K}$ the finitely satisfiable sentences of $\mathcal{K}$. 6
$\varphi \equiv \psi$ logically equivalent formulas. 6
$\mathfrak{A} \equiv \mathfrak{B}$ elementary equivalent structures. 6
$\mathfrak{A} \equiv_r \mathfrak{B}$ r-rank equivalent structures. 6
$\mathfrak{A} \equiv^v \mathfrak{B}$ v-variable equivalent structures. 6
$\mathfrak{A} \equiv_r^v \mathfrak{B}$ r-rank v-variable equivalent structures. 6
$\mathfrak{p}$ parital isomorphism. 6
$G_r(\mathfrak{A}, \mathfrak{B})$ the r-round Ehrenfeucht-Fraïssé game. 6
$\Pi[\Sigma]$ the set of 1-types over Σ. 7
$\mathrm{T}[\Sigma]$ the set of 1-types over Σ. 7
τ^{-1} the inverse of the type τ. 7
$\operatorname{tp}_{\boldsymbol{x}}\tau$ the $\boldsymbol{x}$-type of τ. 7
$\operatorname{tp}_{\boldsymbol{y}}\tau$ the $\boldsymbol{y}$-type of τ. 7
$\operatorname{tp}^{\mathfrak{A}}[a]$ the 1-type of a in $\mathfrak{A}$. 7
$\pi^{\mathfrak{A}}$ the interpretation of the 1-type π in $\mathfrak{A}$. 7
$\operatorname{tp}^{\mathfrak{A}}[a, b]$ the 1-type of a in $\mathfrak{A}$. 7
$\tau^{\mathfrak{A}}$ the interpretation of the 2-type τ in $\mathfrak{A}$. 7
PTIME complexity class. 8
$A \leq_{\mathrm{m}}^{\mathrm{PTIME}} B$ A is polynomial-time reducible to B. 9
$A =_{\mathrm{m}}^{\mathrm{PTIME}} B$ A and B are polynomial-time equivalent. 9
B a bit setup. 11
$[\boldsymbol{u}\text{:data}]^{\mathfrak{A}}$ $\boldsymbol{u}$-data at $\mathfrak{A}$. 11
$[\boldsymbol{u}\text{:eq-}d](\boldsymbol{x})$ $\boldsymbol{u}$-data at $\boldsymbol{x}$ is d. 11
$[\boldsymbol{u}\text{:eq}](\boldsymbol{x}, \boldsymbol{y})$ $\boldsymbol{u}$-data equal at $\boldsymbol{x}$ and $\boldsymbol{y}$. 11
$[\boldsymbol{u}\text{:eq-01}](\boldsymbol{x}, \boldsymbol{y})$ $\boldsymbol{u}$-data at $\boldsymbol{x}$ and $\boldsymbol{y}$ is 0 and 1. 11
$[\boldsymbol{u}\text{:eq-10}](\boldsymbol{x}, \boldsymbol{y})$ $\boldsymbol{u}$-data at $\boldsymbol{x}$ and $\boldsymbol{y}$ is 1 and 0. 11
C a counter setup. 11
$[\text{C:data}]^{\mathfrak{A}}$ C-data at $\mathfrak{A}$. 12
$[\text{C:eq-}d](\boldsymbol{x})$ C-data at $\boldsymbol{x}$ is d. 12
$[\text{C:eq}](\boldsymbol{x}, \boldsymbol{y})$ C-data equal at $\boldsymbol{x}$ and $\boldsymbol{y}$. 12
$[\text{C:less}]d(\boldsymbol{x})$ C-data at $\boldsymbol{x}$ less than d. 12
$[\text{C:betw-}d\text{-}e](\boldsymbol{x})$ C-data at $\boldsymbol{x}$ between d and e. 12
$[\text{C:allbetw-}d\text{-}e]$ C-data between d and e. 12
$[\text{C:less}](\boldsymbol{x}, \boldsymbol{y})$ C-data at $\boldsymbol{x}$ less than C-data at $\boldsymbol{y}$. 13
$[\text{C:succ}](\boldsymbol{x}, \boldsymbol{y})$ C-data at $\boldsymbol{y}$ succeeds C-data at $\boldsymbol{x}$. 13
$[\mathrm{V}(p)\text{:data}]^{\mathfrak{A}}a$ the value of the p-th counter at a. 13
$[\text{V:data}]^{\mathfrak{A}}$ the V-data at a. 13
$[\text{V:eq-v}](\boldsymbol{x})$ the V-data at $\boldsymbol{x}$. 13
$[\mathrm{V}(pq)\text{:at-}i\text{-eq}](\boldsymbol{x})$ equal i-th bits at p and q at $\boldsymbol{x}$. 14
$[\mathrm{V}(pq)\text{:at-}i\text{-eq-01}](\boldsymbol{x})$ equal i-th bits at p and q are 0 and 1. 14
$[\mathrm{V}(pq)\text{:at-}i\text{-eq-10}](\boldsymbol{x})$ equal i-th bits at p and q are 1 and 0. 14
$[\mathrm{V}(pq)\text{:eq}](\boldsymbol{x})$ equal p and q V-data at $\boldsymbol{x}$. 14
$[\mathrm{V}(pq)\text{:less}](\boldsymbol{x})$ V-data at p less than at q. 14
$[\mathrm{V}(pq)\text{:succ}](\boldsymbol{x})$ V-data at q succeeds the data at p. 14
[P:alldiff] P-data at different positions is different. 15
[P:perm] P-data is a permutation. 15
$\mathscr{E}E$ the set of equivalence classes of E. 17
$[\boldsymbol{e}\text{:refl}]$ $\boldsymbol{e}$ is reflexive. 17
$[\boldsymbol{e}\text{:symm}]$ $\boldsymbol{e}$ is symmetric. 17
$[\boldsymbol{e}\text{:trans}]$ $\boldsymbol{e}$ is transitive. 17
$[\boldsymbol{e}\text{:equiv}]$ $\boldsymbol{e}$ is transitive. 17
$[\boldsymbol{d}, \boldsymbol{e}\text{:refine}]$ refinement. 18
$[\boldsymbol{d}, \boldsymbol{e}\text{:global}]$ global agreement. 18

$[\boldsymbol{d}, \boldsymbol{e}\textsf{:local}]$ local agreement. 18
$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:refine}]$ symbols in refinement. 19
$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:global}]$ symbols in global agreement. 19
$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:local}]$ symbols in local agreement. 19
$\Lambda_p^v e\mathrm{E}_{\mathsf{a}}$ logic notation. 21
[P:alleq] P-data equal everywhere. 22
[P:globperm] P-data is a global permutation. 22
$[\mathrm{L}\textsf{:eg-}i](\boldsymbol{x}, \boldsymbol{y})$ global refinement induced by levels. 23
gtr translation of global agreement to refinement. 23
$[\mathrm{E}\text{:chperm}]^{\mathfrak{A}}$ characteristic E-permutation in $\mathfrak{A}$. 25
[L:fixperm] fixed permutation condition. 26
[L:locperm] local agreement condition. 27
$[\mathrm{L}\textsf{:el-}i]$ local refinement induced by levels. 27
ltr translation of local agreement to refinement. 28
g granularity. 29
G granularity color setup. 29
$[\Gamma\textsf{:d}]$ finer equivalence granularity formula. 30
grtr granularity translation. 30
$[\Sigma\textsf{:cell}](\boldsymbol{x}, \boldsymbol{y})$ Σ-cell formula. 33
$\mathcal{O}$ organ-equivalence relation. 34
O sub-organ-equivalence relation. 35
$[\mathrm{D}\text{:Data}]^{\mathfrak{A}}$ D-Data. 39
$[\mathrm{D}\textsf{:Zero}](\boldsymbol{x})$ zero D-Data at $\boldsymbol{x}$. 39
$[\mathrm{D}\textsf{:Largest}](\boldsymbol{x})$ maximum D-Data at $\boldsymbol{x}$. 40
$\boldsymbol{m}_i$ message symbols. 47
$\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ classified signature. 47
(Π, T) type instance. 48
$\pi \sim \pi'$ connectable 1-types. 49

1 Introduction

1.1 Notation

The cardinal number $|A|$ is the *cardinality* of the set A. The set $\wp A$ is the *powerset* of A. The set $\wp^+ A = \wp A \setminus \{\varnothing\}$ is the *set of nonempty subsets* of A. If κ is a cardinal number, the set $\wp^\kappa A = \{S \in \wp A \mid |S| = \kappa\}$ is the *κ-powerset* of A. The *cartesian product* of A and B is $A \times B$. The sets A and B *properly intersect* if $A \cap B \neq \varnothing$, $A \setminus B \neq \varnothing$ and $B \setminus A \neq \varnothing$.

If R is a binary relation, its *domain* is $\operatorname{dom} R$ and its *range* is $\operatorname{ran} R$. The *inverse* of $R \subseteq A \times B$ is

$$R^{-1} = \{(b, a) \in B \times A \mid (a, b) \in R\} .$$

If S is a set and $R \subseteq A \times B$, the *restriction* of R to S is

$$R \restriction S = \{(a, b) \in R \mid a \in S\} .$$

If $R \subseteq A \times B$ is a binary relation and $a \in A$, the *R-successors* of a are

$$R[a] = \{b \in B \mid (a, b) \in R\} .$$

If $S \subseteq B \times C$ and $R \subseteq A \times B$ are two binary relations, their *composition* is

$$S \circ R = \{(a, c) \in A \times C \mid (\exists b \in B)(a, b) \in R \land (b, c) \in S\} .$$

A *function* is formally just a functional relation. The *identity function* on A is id_A. A *total* function from A to B is denoted $f : A \to B$. A *injective* function from A into B is denoted $f : A \hookrightarrow B$. A *surjective* function from A onto B is denoted $f : A \twoheadrightarrow B$. A *bijective* function between A and B is denoted $f : A \leftrightarrow B$. A *partial* function from A to B is denoted $f : A \rightsquigarrow B$. If $f : A \rightsquigarrow B$ is a partial function and $a \in A$, the notation $f(a) \simeq b$ means that f is defined at a and its value is b; the notation $f(a) \simeq \bot$ means that f is not defined at a. If $S \subseteq A$, the *characteristic function* of S in A is $\mathrm{ch}_S^A : A \to \{0, 1\}$.

A *sequence* is formally just a function with domain an ordinal number. If A is a sequence, its *length* $\|\mathrm{A}\|$ is just the domain of A. The sequence consisting of the elements a, b and c in that order is $\langle a, b, c\rangle$. The *empty sequence* is ε. A *finite sequence* is a sequence of finite length. If A and B are two sequences, their *concatenation* is $\mathrm{A} + \mathrm{B}$, and the sequence obtained from A by dropping all elements of B is $\mathrm{A} - \mathrm{B}$.

The set of *natural numbers* is $\mathbb{N} = \{0, 1, \dots\}$. The set of *positive natural numbers* is $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. If $n, m \in \mathbb{N}$ are natural numbers, the *discrete interval* $[n, m]$ between n

and m is

$$[n,m] = \begin{cases} \{n, n+1, \dots, m\} & \text{if } n \le m \\ \varnothing & \text{otherwise.} \end{cases}$$

The function $\log$ is the *base-2 logarithm*.

An n-vector $\mathrm{v} = (\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_n) \in \mathbb{N}^n$ is just a tuple of natural numbers. The n-vector v is *lexicographically smaller* than the n-vector w (written $\mathrm{v} \prec \mathrm{w}$) if there is a position $p \in [1,n]$ such that $\mathrm{v}_p < \mathrm{w}_p$ and $\mathrm{v}_q = \mathrm{w}_q$ for all $q \in [p+1, n]$.

The set of *n-permutations* of $[1,n]$ is $\mathbb{S}_n$. We think of an n-permutation ν as an n-vector $\nu = (\nu(1), \nu(2), \dots, \nu(n))$.

A function $f : \mathbb{N} \to \mathbb{N}$ is *polynomially bounded* if there is a polynomial p and a number $n_0 \in \mathbb{N}$ such that $f(n) \le p(n)$ for all $n \ge n_0$. The function f is *exponentially bounded* if there is a polynomial p and a number $n_0 \in \mathbb{N}$ such that $f(n) \le 2^{p(n)}$ for all $n \ge n_0$. We are going to use these terms implicitly with respect to quantities that depend on one another. For example, the cardinality of $\mathbb{S}_n$ is exponentially bounded by n.

Define the *tetration* operation $\exp_a^e(x)$ by $\exp_a^0(x) = x$ and $\exp_a^{e+1}(x) = a^{\exp_a^e(x)}$, so $\exp_a^e(x) = a^{a^{\cdot^{\cdot^{a^x}}}}$ is a tower of e exponentiations.

An *alphabet* Ω is just a nonempty set. The elements of Ω are *characters*. A *word* $w = w_1 w_2 \dots w_n$ is a finite sequence of characters. The set of words over Ω is Ω^*. The set of nonempty words over Ω is $\Omega^+ = \Omega^* \setminus \{\varepsilon\}$. If $n \in \mathbb{N}$, the set of words of length n over Ω is Ω^n.

The set of *bits* is $\mathbb{B} = \{0,1\}$. The set of *bitstrings* is $\mathbb{B}^+$. The bitstrings are read right-to-left, that is the bitstring $\mathrm{b} = 10$ has first character 0. If $t < u \in \mathbb{N}^+$, the t-bit bitstrings $\mathbb{B}^t$ are embedded into the u-bit bitstrings $\mathbb{B}^u$ by appending leading zeroes. If $n \in \mathbb{N}$, the *bitsize* $\|n\|$ of n is:

$$\|n\| = \begin{cases} 1 & \text{if } n = 0 \\ \lfloor \log n \rfloor + 1 & \text{otherwise.} \end{cases}$$

If $n \in \mathbb{N}$, the *binary encoding* of n is $\overline{n} \in \mathbb{B}^{\|n\|}$. If $\mathrm{b} \in \mathbb{B}^t$, the *number encoded by* b is $\underline{\mathrm{b}}$. The *largest t-bit number* is $N_t = 2^t - 1$. The set of *t-bit numbers* is $\mathbb{B}_t = [0, N_t]$.

1.2 Syntax

The *symbol alphabet* for the first-order logic with counting quantifiers is

$$\Omega_{\mathcal{C}} = \left\{ \neg, \wedge, \vee, \to, \leftrightarrow; \exists, \forall; =; (,,,); {}^{\le}, {}^{=}, {}^{\ge}, {}^{0}, {}^{1} \right\}.$$

The propositional connectives are listed in decreasing order of precedence. The *negation* $\neg$ is unary; the *disjunction* $\vee$, *conjunction* $\wedge$ and *equivalence* $\leftrightarrow$ are left-associative; the *implication* $\to$ is right-associative. The quantifiers bind as strong as the negation. Note that we consider logics with *formal equality* $=$.

A *counting quantifier* is a word over $\Omega_{\mathcal{C}}$ of the form $\exists^{\le \overline{m}}$ or $\exists^{= \overline{m}}$ or $\exists^{\ge \overline{m}}$, where $m \in \mathbb{N}$ and $\overline{m} \in \mathbb{B}^+$ is the binary encoding of m. Note that this encoding of the counting quantifiers is *succinct*. As we note in Remark 1, this succinct representation allows for exponentially small counting formulas compared to their pure first-order equivalents. We denote the counting quantifiers by $\exists^{\leq m}$, $\exists^{=m}$ and $\exists^{\geq m}$, that is, we omit the encoding notation for m.

The sequence $\mathcal{V} = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \rangle$ is a countable sequence of distinct *variable symbols*. We pay special attention to $\boldsymbol{x} = \boldsymbol{v}_1$, $\boldsymbol{y} = \boldsymbol{v}_2$ and $\boldsymbol{z} = \boldsymbol{v}_3$, the *first*, *second* and *third* variable symbol, respectively.

A *predicate signature* $\Sigma = \langle \boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s \rangle$ is a finite sequence of distinct *predicate symbols* $\boldsymbol{p}_i$ together with their *arities* $\operatorname{ar} \boldsymbol{p}_i \in \mathbb{N}^+$. A predicate signature is *unary* or *monadic* if all of its predicate symbols have arity 1. A predicate signature is *binary* if all of its predicate symbols have arity 1 or 2. For the purposes of this work we will not be considering constant and function symbols—constant symbols can be simulated by a fresh unary predicate symbol having the intended interpretation of being true at a unique element; presence of function symbols on the other hand leads quite easily to undecidable satisfiability problems. By convention $\Omega_{\mathcal{C}}$, $\mathcal{V}$ and Σ are disjoint.

Let Σ be a predicate signature. The set of *atomic formulas* $\mathcal{A}t[\Sigma] \subset (\Omega_{\mathcal{C}} \cup \mathcal{V} \cup \Sigma)^*$ over Σ is generated by the grammar:

$$\alpha ::= (x = y) \mid p(x_1, x_2, \ldots, x_n)$$

for $x, y \in \mathcal{V}$, $p \in \Sigma$, $n = \operatorname{ar} p$ and $x_1, x_2, \ldots, x_n \in \mathcal{V}$.

The set of *literals* $\mathcal{L}it[\Sigma] \subset (\Omega_{\mathcal{C}} \cup \mathcal{V} \cup \Sigma)^*$ over Σ is generated by the grammar:

$$\lambda ::= \alpha \mid (\neg\alpha).$$

The set of *first-order formulas with counting quantifiers* $\mathcal{C}[\Sigma] \subset (\Omega_{\mathcal{C}} \cup \mathcal{V} \cup \Sigma)^*$ over Σ is generated by the grammar:

$$\begin{aligned} \varphi ::= \alpha \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi) \mid (\exists x \varphi) \mid (\forall x \varphi) \mid \\ \mid (\exists^{\leq m} x \varphi) \mid (\exists^{=m} x \varphi) \mid (\exists^{\geq m} x \varphi) \end{aligned}$$

for $x \in \mathcal{V}$ and $m \in \mathbb{N}$.

The set of *first-order formulas* $\mathcal{L}[\Sigma] \subset \mathcal{C}[\Sigma]$ over Σ consists of the formulas that do not feature a counting quantifier.

The set of variables occurring in φ is $\operatorname{vr} \varphi \subset \mathcal{V}$. The set of variables freely occurring in φ is $\operatorname{fvr} \varphi \subset \mathcal{V}$. A formula φ is a *sentence* if $\operatorname{fvr} \varphi = \varnothing$. For $v \in \mathbb{N}$, a formula φ is a *v-variable formula* if $\operatorname{vr} \varphi \subseteq \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_v\}$. The set of v-variable first-order formulas over Σ is $\mathcal{L}^v[\Sigma]$. The set of v-variable first-order formulas with counting quantifiers over Σ is $\mathcal{C}^v[\Sigma]$.

If $\varphi \in \mathcal{C}[\Sigma]$, the *quantifier rank* $\operatorname{qr} \varphi \in \mathbb{N}$ of φ is defined as follows. If φ matches:

- $(x = y)$, then $\operatorname{qr} \varphi = 0$
- $p(x_1, x_2, \ldots, x_n)$, then $\operatorname{qr} \varphi = 0$
- $(\neg\psi)$, then $\operatorname{qr} \varphi = \operatorname{qr} \psi$

- $\psi_1 \oplus \psi_2$ for $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then $\operatorname{qr} \varphi = \max(\operatorname{qr} \psi_1, \operatorname{qr} \psi_2)$
- $(\exists x \psi)$ or $(\forall x \psi)$, then $\operatorname{qr} \varphi = 1 + \operatorname{qr} \psi$
- $(\exists^{\leq m} x \psi)$ or $(\exists^{=m} x \psi)$, then $\operatorname{qr} \varphi = m + 1 + \operatorname{qr} \psi$
- $(\exists^{\geq m} x \psi)$, then $\operatorname{qr} \varphi = m + \operatorname{qr} \psi$.

An r*-rank formula* is a formula having quantifier rank r. The set of r-rank first-order formulas over Σ is $\mathcal{L}_r[\Sigma]$. The set of r-rank first-order formulas with counting quantifiers over Σ is $\mathcal{C}_r[\Sigma]$. The set of r-rank v-variable first-order formulas over Σ is $\mathcal{L}_r^v[\Sigma]$. The set of r-rank v-variable first-order formulas with counting quantifiers over Σ is $\mathcal{C}_r^v[\Sigma]$.

If φ is a formula and $x_1, x_2, \ldots, x_n \in \mathcal{V}$ are distinct variables, we use the notation $\varphi(x_1, x_2, \ldots, x_n)$, a *focused formula*, to show that we are interested in the free occurrences of the variables x_i in φ. If $\varphi(x_1, x_2, \ldots, x_n)$ is a focused formula and $y_1, y_2, \ldots, y_n \in \mathcal{V}$, then $\varphi(y_1, y_2, \ldots, y_n)$ denotes the formula φ where all free occurrences of x_i are replaced by y_i. The notation $\varphi = \varphi(x_1, x_2, \ldots, x_n)$ means that $\operatorname{fvr} \varphi \subseteq \{x_1, x_2, \ldots, x_n\}$.

We will omit unnecessary brackets in formulas.

1.3 Semantics

If Σ is a predicate signature, a Σ*-structure* $\mathfrak{A}$ consists of a nonempty set A (the *domain* of $\mathfrak{A}$), together with a relation $p^{\mathfrak{A}} \subseteq A^{\operatorname{ar} p}$ (the *interpretation* of p at $\mathfrak{A}$) for every predicate symbol $p \in \Sigma$. A structure is *finite* if its domain is finite. We omit the standard definition of semantic notions. Seldom it will be useful to consider *structures with possibly empty domain*. We will be explicit when this is the case. If $\mathfrak{A}$ is a structure and $B \subseteq A$ there is a substructure $\mathfrak{B} \subseteq \mathfrak{A}$ with possibly empty domain B. We call it the substructure induced by B and denote it $(\mathfrak{A} \upharpoonright B)$.

Note that the interpretation of the counting quantifiers is clear: $\exists^{\leq m} x \varphi$ means that "at most m elements satisfy φ"; $\exists^{=m} x \varphi$ means that "exactly m elements satisfy φ"; $\exists^{\geq m} x \varphi$ means that "at least m elements satisfy φ".

The *standard translation* $\operatorname{st} : \mathcal{C}[\Sigma] \to \mathcal{L}[\Sigma]$ of first-order formulas with counting quantifiers to logically equivalent first-order formulas is defined as follows. If φ matches:

- $(x = y)$ or $p(x_1, x_2, \ldots, x_n)$, then $\operatorname{st} \varphi = \varphi$
- $(\neg \psi)$, then $\operatorname{st} \varphi = (\neg \operatorname{st} \psi)$
- $(\psi_1 \oplus \psi_2)$ for $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then $\operatorname{st} \varphi = (\operatorname{st} \psi_1 \oplus \operatorname{st} \psi_2)$
- $(\mathsf{Q} x \psi)$ for $\mathsf{Q} \in \{\exists, \forall\}$, then $\operatorname{st} \varphi = (\mathsf{Q} x \operatorname{st} \psi)$

- $(\exists^{\leq m} x \psi(x))$ or $(\exists^{=m} x \psi(x))$ or $(\exists^{\geq m} x \psi(x))$, then let

$$\theta_{\leq} = \forall y_1 \forall y_2 \dots \forall y_m \forall y_{m+1} \left(\bigwedge_{1 \leq i \leq m+1} \operatorname{st} \psi(y_i) \to \bigvee_{1 \leq i < j \leq m+1} y_i = y_j \right)$$

$$\theta_{\geq} = \exists y_1 \exists y_2 \dots \exists y_m \left(\bigwedge_{1 \leq i \leq m} \operatorname{st} \psi(y_i) \wedge \bigwedge_{1 \leq i < j \leq m} y_i \neq y_j \right)$$

where $y_1, y_2, \dots, y_{m+1}$ are distinct variable symbols not occurring in φ. The formula $\theta_{\leq}$ asserts that there are at most m distinct values satisfying ψ. The formula $\theta_{\geq}$ asserts that there are at least m distinct values satisfying ψ. If $\varphi = (\exists^{\leq m} x \psi(x))$, then $\operatorname{st} \varphi = \theta_{\leq}$. If $\varphi = (\exists^{=m} x \psi(x))$, then $\operatorname{st} \varphi = (\theta_{\leq} \wedge \theta_{\geq})$. If $\varphi = (\exists^{\geq m} x \psi(x))$, then $\operatorname{st} \varphi = \theta_{\geq}$.

Remark 1. *The translation of a first-order formula with counting quantifiers φ to a logically equivalent first-order formula $\psi = \operatorname{st} \varphi$ preserves quantifier rank. However, the resulting formula ψ may have exponentially larger length.*

A *predicate signature with intended interpretations* Σ is formally a predicate signature together with an *intended interpretation condition* $\mathcal{A}$, which is formally a class of Σ-structures. A Σ-structure $\mathfrak{A}$ is then just an element of $\mathcal{A}$. That is, when we speak about a predicate signature with intended interpretations, we are considering the logics strictly over the class of structures respecting the intended interpretation condition. The semantic concepts are relativised appropriately in this context. For example, if $\Sigma = \langle \boldsymbol{e} \rangle$ is a predicate signature consisting of the single binary predicate symbol $\boldsymbol{e}$, having intended interpretation as an equivalence, then the Σ-formula $\forall \boldsymbol{x} \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{x})$ is logically valid. From now on, we will use the term *predicate signature* as *predicate signature with possible intended interpretations*.

The predicate signature Σ' is an *enrichment* of the predicate signature Σ if Σ' contains all predicate symbols of Σ and respects their intended interpretation in Σ. A Σ'-structure $\mathfrak{A}'$ is an enrichment of the Σ-structure $\mathfrak{A}$ if they have the same domain and the same interpretation of the predicate symbols of Σ. The basic semantic significance of enrichment is that if $\varphi(x_1, x_2, \dots, x_n)$ is a Σ-formula and $a_1, a_2, \dots, a_n \in A$, then $\mathfrak{A} \vDash \varphi(a_1, a_2, \dots, a_n)$ iff $\mathfrak{A}' \vDash \varphi(a_1, a_2, \dots, a_n)$. If $\mathfrak{A}'$ is an enrichment of $\mathfrak{A}$ then $\mathfrak{A}$ is a *reduct*[1] of $\mathfrak{A}'$.

If $\varphi(x_1, x_2, \dots, x_n)$ is a focused formula, the interpretation of φ in $\mathfrak{A}$ is

$$\varphi^{\mathfrak{A}} = \{(a_1, a_2, \dots, a_n) \in A^n \mid \mathfrak{A} \vDash \varphi(a_1, a_2, \dots, a_n)\}.$$

If Σ is a predicate signature and φ is a Σ-sentence, then φ is *satisfiable* if there is a Σ-structure that is a model for φ; φ is *finitely satisfiable* if there is a *finite* Σ-structure that is a model for φ. If $\mathcal{K} \subseteq \mathcal{C}[\Sigma]$ is a family of formulas over the predicate signature Σ, the set of *satisfiable sentences* is $\mathrm{SAT}\mathcal{K} \subseteq \mathcal{K}$ and the set of *finitely satisfiable sentences* is

[1] or why not *empoverishment*?

$\text{FINSAT}\mathcal{K} \subseteq \mathcal{K}$. The family $\mathcal{K}$ has the *finite model property* if $\text{SAT}\mathcal{K} = \text{FINSAT}\mathcal{K}$. By the Löwenheim-Skolem theorem, every satisfiable sentence φ has a finite or countable model (assuming the intended interpretation condition of the predicate signature is first-order-definable). In this work the intended interpretation conditions of the predicate signatures will always be first-order-definable formula and we will silently assume that all structures are either finite or countable.

Two Σ-sentences φ and ψ are *logically equivalent* (written $\varphi \equiv \psi$) if they have the same models.

Two Σ-structures $\mathfrak{A}$ and $\mathfrak{B}$ are *elementary equivalent* (written $\mathfrak{A} \equiv \mathfrak{B}$) if they satisfy the same first-order sentences (hence also the same first-order sentences with counting quantifiers). The structures $\mathfrak{A}$ and $\mathfrak{B}$ are *r-rank equivalent* (written $\mathfrak{A} \equiv_r \mathfrak{B}$) if they satisfy the same r-rank first-order sentences. The structures $\mathfrak{A}$ and $\mathfrak{B}$ are *v-variable equivalent* (written $\mathfrak{A} \equiv^v \mathfrak{B}$) if they satisfy the same v-variable first-order sentences. The structures $\mathfrak{A}$ and $\mathfrak{B}$ are *r-rank v-variable equivalent* (written $\mathfrak{A} \equiv^v_r \mathfrak{B}$) if they satisfy the same r-rank v-variable first-order sentences.

1.4 Logic games

Logic games capture structure equivalence. Let Σ be a predicate signature and let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures. A *partial isomorphism* $\mathfrak{p} : A \rightsquigarrow B$ from $\mathfrak{A}$ to $\mathfrak{B}$ is a partial mapping that is an isomorphism between the induced substructures $(\mathfrak{A} \restriction \operatorname{dom} \mathfrak{p})$ and $(\mathfrak{B} \restriction \operatorname{ran} \mathfrak{p})$.

Let $r \in \mathbb{N}^+$. The *r-round Ehrenfeucht-Fraïssé game* $G_r(\mathfrak{A}, \mathfrak{B})$ is a two-player game, played with a pair of pebbles, one for each structure. The two players are Spoiler and Duplicator. Initially the pebbles are off the structures. During each round, Spoiler picks a pebble and places it on some element in its designated structure. Duplicator responds by picking the other pebble and placing it on some element in the other structure. Thus during round i, the players play a pair of elements $a_i \mapsto b_i \in A \times B$. Collect the sequences of played elements $\bar{a} = \langle a_1, a_2, \ldots, a_r \rangle$ and $\bar{b} = \langle b_1, b_2, \ldots, b_r \rangle$. Duplicator wins the match if the relation $\bar{a} \mapsto \bar{b} = \{a_1 \mapsto b_1, a_2 \mapsto b_2, \ldots, a_r \mapsto b_r\} \subseteq A \times B$, built from the pairs of elements in each round, is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. Ehrenfeucht's theorem says that Duplicator has a winning strategy for $G_r(\mathfrak{A}, \mathfrak{B})$ iff $\mathfrak{A} \equiv_r \mathfrak{B}$. Fraïssé's theorem gives a back-and-forth characterization of the winning strategy for Duplicator [1, ch. 2]:

Theorem 1. *Suppose that $(\mathfrak{I}_0, \mathfrak{I}_1, \ldots, \mathfrak{I}_r)$ is a sequence of nonempty sets of partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$ with the following properties:*

1. *For every $j < r$, $\mathfrak{p} \in \mathfrak{I}_{j+1}$ and $a \in A$, there is $\mathfrak{q} \in \mathfrak{I}_j$ such that $\mathfrak{p} \subseteq \mathfrak{q}$ and $a \in \operatorname{dom} \mathfrak{q}$.*
2. *For every $j < r$, $\mathfrak{p} \in \mathfrak{I}_{j+1}$ and $b \in B$, there is $\mathfrak{q} \in \mathfrak{I}_j$ such that $\mathfrak{p} \subseteq \mathfrak{q}$ and $b \in \operatorname{ran} \mathfrak{q}$.*

Then $\mathfrak{A} \equiv_r \mathfrak{B}$.

1.5 Types

Let $\Sigma = \langle \boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s \rangle$ be a predicate signature. A 1-*type* π over Σ is a maximal consistent set of literals featuring only the variable symbol $\boldsymbol{x}$. The set of 1-types over Σ is $\Pi[\Sigma]$. Note that consistency here is relativised by the intended interpretations of the predicate signature. For example if Σ contains the binary predicate symbol $\boldsymbol{e}$ with intended interpretation as an equivalence, then every 1-type over Σ includes the literal $\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{x})$. Also note that the cardinality of a 1-type over Σ is polynomially bounded by the length s of Σ and the cardinality of $\Pi[\Sigma]$ is exponentially bounded by s.

A 2-*type* τ over Σ is a maximal consistent set of literals featuring only the variable symbols $\boldsymbol{x}$ and $\boldsymbol{y}$ and including the literal $(\boldsymbol{x} \neq \boldsymbol{y})$. The set of 2-types over Σ is $\mathrm{T}[\Sigma]$. Again, consistency is relativised by the intended interpretation of the predicate signature. For example, if Σ contains the binary predicate symbol $\boldsymbol{e}$ with intended interpretation as an equivalence, then if $\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \in \tau$, then $\boldsymbol{e}(\boldsymbol{y}, \boldsymbol{x}) \in \tau$. Again, the cardinality of a 2-type over Σ is polynomially bounded by s and the cardinality of $\mathrm{T}[\Sigma]$ is exponentially bounded by s.

If $\tau \in \mathrm{T}[\Sigma]$, the *inverse* τ^{-1} of τ is the 2-type obtained from τ by swapping the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ in every literal. The $\boldsymbol{x}$-*type* of τ is the 1-type $\mathrm{tp}_{\boldsymbol{x}}\, \tau$ consisting of all the literals of τ featuring only the variable symbol $\boldsymbol{x}$. Similarly, the $\boldsymbol{y}$-*type* of τ is the 1-type $\mathrm{tp}_{\boldsymbol{y}}\, \tau$ consisting of all the literals of τ featuring only the variable symbol $\boldsymbol{y}$, that is replaced by $\boldsymbol{x}$. For instance we have the identity $\mathrm{tp}_{\boldsymbol{x}}\, \tau^{-1} = \mathrm{tp}_{\boldsymbol{y}}\, \tau$. We say that τ *connects* the 1-types $\mathrm{tp}_{\boldsymbol{x}}\, \tau$ and $\mathrm{tp}_{\boldsymbol{y}}\, \tau$.

If $\mathfrak{A}$ is a Σ-structure and $a \in A$, the 1-*type of* a *in* $\mathfrak{A}$ is

$$\mathrm{tp}^{\mathfrak{A}}[a] = \{\lambda(\boldsymbol{x}) \in \mathcal{L}it[\Sigma] \mid \mathfrak{A} \models \lambda(a)\}\,.$$

If $\mathrm{tp}^{\mathfrak{A}}[a] = \pi$, we say that the 1-type π is *realized* by a in $\mathfrak{A}$. The interpretation of the 1-type π in $\mathfrak{A}$ is the set of elements realizing π:

$$\pi^{\mathfrak{A}} = \left\{ a \in A \,\middle|\, \mathrm{tp}^{\mathfrak{A}}[a] = \pi \right\}.$$

If $a \neq b \in A$, the 2-*type of* (a, b) *in* $\mathfrak{A}$ is

$$\mathrm{tp}^{\mathfrak{A}}[a, b] = \{\lambda(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{L}it[\Sigma] \mid \mathfrak{A} \models \lambda(a, b)\}\,.$$

We do not define a 2-type in case $a = b$. If $\mathrm{tp}^{\mathfrak{A}}[a, b] = \tau$, we say that the 2-type τ is *realized* by (a, b) in $\mathfrak{A}$. The interpretation of the 2-type τ in $\mathfrak{A}$ is the set of pairs realizing τ:

$$\tau^{\mathfrak{A}} = \left\{ (a, b) \in A \times A \,\middle|\, a \neq b \wedge \mathrm{tp}^{\mathfrak{A}}[a, b] = \tau \right\}.$$

1.6 Normal forms

In two-variable logics, a common technique of reducing formula quantifier rank while preserving satisfiability is Skolemization [2]: Let φ be a $\mathcal{L}^2$-sentence. By replacing universally quantified subformulas $\forall x\psi$ by twofold existential negations $\neg\exists x\neg\psi$, without loss of generality assume that only existential quantifiers occur in φ. Consider a subformula ψ of φ that has the lowest possible nontrivial quantifier rank 1. Then $\psi = \psi(y) = \exists x\alpha(x, y)$, where the formula α is quantifier-free, $\{x, y\} = \{\boldsymbol{x}, \boldsymbol{y}\}$ and y may or may not necessarly occur freely in α. Introduce a new unary predicate symbol $\boldsymbol{u}_\psi$ with the intended interpretation $\forall y\boldsymbol{u}_\psi(y) \leftrightarrow \exists x\alpha(x, y)$ and let φ' be the formula obtained from φ by replacing the subformula ψ by $\boldsymbol{u}_\psi(y)$. The original formula φ is equisatisfiable with $\varphi_1 = (\forall y\boldsymbol{u}_\psi(y) \leftrightarrow \exists x\alpha(x, y)) \wedge \varphi'$ in a strinct sense, that is any model for φ can be $\boldsymbol{u}_\psi$-enriched into a model for φ_1 and any model for φ_1 is a model for φ. By repeating this process linearly many times, we can bring the formula to a form where the quantifier rank is at most 2 [3, 2]:

Theorem 2 (Scott)**.** *There is a polynomial-time reduction* sctr : $\mathcal{L}^2 \to \mathcal{L}^2$ *which reduces every sentence* φ *to a sentence* sctr φ *in* Scott normal form*:*

$$(\forall\boldsymbol{x}\forall\boldsymbol{y}\alpha_0(\boldsymbol{x},\boldsymbol{y}) \vee \boldsymbol{x} = \boldsymbol{y}) \wedge \bigwedge_{1\le i\le m} \forall\boldsymbol{x}\exists\boldsymbol{y}\alpha_i(\boldsymbol{x},\boldsymbol{y}) \wedge \boldsymbol{x} \neq \boldsymbol{y},$$

where the formulas α_i *are quantifier-free and use at most linearly many new unary predicate symbols. The sentences* φ *and* sctr φ *are satisfiable over the same domains. Moreover the length* sctr φ *is linear in the length of* φ*.*

A completely analogous normal form can be described for the two-variable fragment with counting quantifiers [4]:

Theorem 3 (Pratt-Hartmann)**.** *There is a polynomial-time reduction* prtr : $\mathcal{C}^2 \to \mathcal{C}^2$ *with reduces every sentence* φ *to a sentence* prtr φ *in the form:*

$$(\forall\boldsymbol{x}\forall\boldsymbol{y}\alpha_0(\boldsymbol{x},\boldsymbol{y}) \vee \boldsymbol{x} = \boldsymbol{y}) \wedge \bigwedge_{1\le i\le m} \forall\boldsymbol{x}\exists^{=M_i}\boldsymbol{y}\alpha_i(\boldsymbol{x},\boldsymbol{y}) \wedge \boldsymbol{x} \neq \boldsymbol{y},$$

where the formulas α_i *are quantifier-free and may use linearly many new unary and binary predicate symbols. Let* $M = \max\{M_1, M_2, \ldots, M_m\}$*. Then* φ *and* prtr φ *are satisfiable over the same domains of cardinality greater than* M*. Moreover the length* prtr φ *is linear in the length of* φ*.*

1.7 Complexity

We denote the complexity classes PTime $= \textsc{Time}[\mathrm{poly}(n)] = \bigcup_{c\in\mathbb{N}^+} \textsc{Time}[n^c]$, NPTime, PSpace, ExpTime and NExpTime. For $e \in \mathbb{N}^+$, the e-exponential deterministic and nondeterministic time classes are eExpTime $= \textsc{Time}[\exp_2^e(\mathrm{poly}(n))]$ and NeExpTime. The complexity class Elementary is the union of the complexity classes eExpTime for $e \in \mathbb{N}^+$.

The Grzegorczyk hierarchy $\mathcal{E}^i$ for $i \in \mathbb{N}$ orders the primitive recursive functions by means of the power of recursion needed. The *basic functions* are the zero function $\mathrm{zero}(n) = 0$, the successor function $\mathrm{succ}(n) = n + 1$ and the projection functions

$\mathrm{proj}_i^u(n_1, n_2, \ldots, n_u) = n_i$. If $u, v \in \mathbb{N}$, $f : \mathbb{N}^u \to \mathbb{N}$ and $g_1, g_2, \ldots, g_u : \mathbb{N}^v \to \mathbb{N}$ are functions, their *superposition* is the function $h : \mathbb{N}^v \to \mathbb{N}$ defined by $h(\bar{n}) = f(g_1(\bar{n}), g_2(\bar{n}), \ldots, g_u(\bar{n}))$ for $\bar{n} \in \mathbb{N}^v$. If $u \in \mathbb{N}$, $f : \mathbb{N}^u \to \mathbb{N}$ and $g : \mathbb{N}^{u+2} \to \mathbb{N}$, their *primitive recursion* is the function $h : \mathbb{N}^{u+1} \to \mathbb{N}$ defined by:

$$h(\bar{n}, 0) = f(\bar{n})$$
$$h(\bar{n}, i+1) = g(\bar{n}, i, h(\bar{n}, i))$$

for $\bar{n} \in \mathbb{N}^u$. For $i \in \mathbb{N}$, define the function E_i by $E_0(n) = n + 1$ and

$$E_{i+1}(n) = E_i^n(2) = \underbrace{E_i(E_i(\ldots E_i}_{n}(2))).$$

For $i \in \mathbb{N}$, the i-th level of the Grzegorczyk hierarchy $\mathcal{E}^i$ as the least set of functions containing the basic functions, the functions E_k for $k \in [0, i]$ and closed under superposition and *limited primitive recursion*, that is a primitive recursion $h : \mathbb{N}^{u+1}$ of the functions $f : \mathbb{N}^u \to \mathbb{N}$, $g : \mathbb{N}^{u+2} \to \mathbb{N}$, $f, g \in \mathcal{E}^i$, such that there is a function $b : \mathbb{N}^{u+1} \to \mathbb{N}$, $b \in \mathcal{E}^i$ bounding h: $h(\bar{n}) \leq b(\bar{n})$ for all $n \in \mathbb{N}^{u+1}$. A decision problem $A \subseteq \Omega^*$ is in some level of the Grzegorczyk hierarchy just in case its characteristic function occurs at that level. The primitive recursive functions are partitioned by the Grzegorczyk hierarchy. The complexity class Elementary coincides with the third level of the Grzegorczyk hierarchy $\mathcal{E}^3$.

If $A \subseteq \Omega_1^*$ and $B \subseteq \Omega_2^*$ are decision problems, the problem A is *many-one polynomial-time reducible* to B (written $A \leq_{\mathrm{m}}^{\mathrm{PTime}} B$) if there is a polynomial-time algorithm $f : \Omega_1^* \to \Omega_2^*$ such that $a \in A$ iff $f(a) \in B$. Similar reductions are defined analogously. The decision problems A and B are *many-one polynomial-time equivalent* (written $A =_{\mathrm{m}}^{\mathrm{PTime}} B$) if $A \leq_{\mathrm{m}}^{\mathrm{PTime}} B$ and $B \leq_{\mathrm{m}}^{\mathrm{PTime}} A$.

A decision problem is *hard* for a complexity class if any decision problem of that complexity class is polynomial-time reducible to it. A decision problem is *complete* for a complexity class if it is hard for that class and contained in that class.

We will need the following standard domino tiling problem [5, p. 403]: A *domino system* is a triple $D = (T, H, V)$, where $T = [1, k]$ is a finite set of *tiles* and $H, V \subseteq T \times T$ are *horizontal* and *vertical matching relations*. A *tiling* of $m \times m$ for a domino system D with *initial condition* $c^0 = \langle t_1^0, t_2^0, \ldots, t_n^0 \rangle$, where $n \leq m$, is a mapping $t : [1, m] \times [1, m] \to T$ such that:

- $(t(i, j), t(i+1, j)) \in H$ for all $i \in [1, m-1]$ and $j \in [1, m]$
- $(t(i, j), t(i, j+1)) \in V$ for all $i \in [1, m]$ and $j \in [1, m-1]$
- $t(i, 1) = t_i^0$ for all $i \in [1, n]$.

It is well-known [6, 7] that there exists a *"Turing-complete"* domino system D_0 for which:

- the problem asking whether there exists a tiling of $m \times m$ with initial condition c^0 of length n, where $m = n$, is NPTime-complete.

- the problem asking whether there exists a tiling of $m \times m$ with initial condition c^0 of length n, where $m = 2^n$, is NExpTime-complete.
- the problem asking whether there exists a tiling of $m \times m$ with initial condition c^0 of length n, where $m = 2^{2^n}$, is N2ExpTime-complete.
- the argument extends to arbitrary exponential towers: the problem asking whether there exists a tiling of $m \times m$ with initial condition c^0 of length n, where $m = \exp_2^e(n)$ is NeExpTime-complete.

2 Counter setups

2.1 Bits

A *bit setup* $\mathrm{B} = \langle \boldsymbol{u} \rangle$ is a predicate signature consisting of a single unary predicate symbol $\boldsymbol{u}$.

Definition 1. *Let* $\mathfrak{A}$ *be a* B*-structure. Define the function* $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} : A \to \mathbb{B}$ *by:*

$$[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} a = \begin{cases} 1 & \text{if } \mathfrak{A} \models \boldsymbol{u}(a) \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2. *Let* $d \in \mathbb{B}$*. Define the quantifier-free* $\mathcal{L}^1[\mathrm{B}]$*-formula* $[\boldsymbol{u}\text{:}\mathsf{eq}\text{-}d](\boldsymbol{x})$ *by:*

$$[\boldsymbol{u}\text{:}\mathsf{eq}\text{-}d](\boldsymbol{x}) = \begin{cases} \boldsymbol{u}(\boldsymbol{x}) & \text{if } d = 1 \\ \neg\boldsymbol{u}(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

If $\mathfrak{A}$ is a B-structure, $a \in A$ and $d \in \mathbb{B}$, then $\mathfrak{A} \models [\boldsymbol{u}\text{:}\mathsf{eq}\text{-}d](a)$ iff $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} a = d$.

Definition 3. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{B}]$*-formulas* $[\boldsymbol{u}\text{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y})$*,* $[\boldsymbol{u}\text{:}\mathsf{eq}\text{-}01](\boldsymbol{x}, \boldsymbol{y})$ *and* $[\boldsymbol{u}\text{:}\mathsf{eq}\text{-}10](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$\begin{aligned} [\boldsymbol{u}\text{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{u}(\boldsymbol{x}) \leftrightarrow \boldsymbol{u}(\boldsymbol{y}) \\ [\boldsymbol{u}\text{:}\mathsf{eq}\text{-}01](\boldsymbol{x}, \boldsymbol{y}) &= \neg\boldsymbol{u}(\boldsymbol{x}) \wedge \boldsymbol{u}(\boldsymbol{y}) \\ [\boldsymbol{u}\text{:}\mathsf{eq}\text{-}10](\boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{u}(\boldsymbol{x}) \wedge \neg\boldsymbol{u}(\boldsymbol{y}). \end{aligned}$$

If $\mathfrak{A}$ is a B-structure and $a, b \in A$, then:

- $\mathfrak{A} \models [\boldsymbol{u}\text{:}\mathsf{eq}](a, b)$ iff $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} a = [\boldsymbol{u}\text{:data}]^{\mathfrak{A}} b$
- $\mathfrak{A} \models [\boldsymbol{u}\text{:}\mathsf{eq}\text{-}01](a, b)$ iff $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} a = 0$ and $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} b = 1$
- $\mathfrak{A} \models [\boldsymbol{u}\text{:}\mathsf{eq}\text{-}10](a, b)$ iff $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} a = 1$ and $[\boldsymbol{u}\text{:data}]^{\mathfrak{A}} b = 0$.

2.2 Counters

A *t-bit counter setup* for $t \in \mathbb{N}^+$ is a predicate signature $\mathrm{C} = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_t \rangle$ consisting of t distinct unary predicate symbols $\boldsymbol{u}_i$.

Definition 4. *Let* $\mathfrak{A}$ *be a* C*-structure. Define the function* $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} : A \to \mathbb{B}_t$ *by:*

$$[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = \sum_{1 \le i \le t} 2^{i-1} [\boldsymbol{u}_i\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a.$$

Definition 5. *Let* $d \in \mathbb{B}_t$ *be a* t*-bit number. Define the quantifier-free* $\mathcal{L}^1[\mathrm{C}]$*-formula* $[\mathrm{C}\mathbf{:}\mathsf{eq}\text{-}d](\boldsymbol{x})$ *by:*

$$[\mathrm{C}\mathbf{:}\mathsf{eq}\text{-}d](\boldsymbol{x}) = \bigwedge_{1 \le i \le t} [\boldsymbol{u}_i\mathbf{:}\mathsf{eq}\text{-}\overline{d}_i](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a C-structure, $a \in A$ and $d \in \mathbb{B}_t$, then $\mathfrak{A} \vDash [\mathrm{C}\mathbf{:}\mathsf{eq}\text{-}d](a)$ iff $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = d$.

If A is a nonempty set and $\mathrm{data} : A \to \mathbb{B}_t$ is any function, there is a C-structure $\mathfrak{A}$ over A such that $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} = \mathrm{data}$.

Definition 6. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{C}]$*-formula* $[\mathrm{C}\mathbf{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{C}\mathbf{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}) = \bigwedge_{1 \le i \le t} [\boldsymbol{u}_i\mathbf{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}).$$

If $\mathfrak{A}$ is a C-structure and $a, b \in A$, then $\mathfrak{A} \vDash [\mathrm{C}\mathbf{:}\mathsf{eq}](a, b)$ iff $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = [\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} b$.

Definition 7. *Let* $d \in \mathbb{B}_t$*. Define the quantifier-free* $\mathcal{L}^1[\mathrm{C}]$*-formula* $[\mathrm{C}\mathbf{:}\mathsf{less}]d(\boldsymbol{x})$ *by:*

$$[\mathrm{C}\mathbf{:}\mathsf{less}\text{-}d](\boldsymbol{x}) = \bigvee_{1 \le j \le t} \neg \boldsymbol{u}_j(\boldsymbol{x}) \wedge \neg [\boldsymbol{u}_j\mathbf{:}\mathsf{eq}\text{-}\overline{d}_j](\boldsymbol{x}) \wedge \bigwedge_{j < k \le t} [\boldsymbol{u}_k\mathbf{:}\mathsf{eq}\text{-}\overline{d}_k](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a C-structure, $a \in A$ and $d \in \mathbb{B}_t$, then $\mathfrak{A} \vDash [\mathrm{C}\mathbf{:}\mathsf{less}\text{-}d](a)$ iff $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a < d$.

Definition 8. *Let* $d \le e \in \mathbb{B}_t$*. Define the quantifier-free* $\mathcal{L}^1[\mathrm{C}]$*-formula* $[\mathrm{C}\mathbf{:}\mathsf{betw}\text{-}d\text{-}e](\boldsymbol{x})$ *by:*

$$[\mathrm{C}\mathbf{:}\mathsf{betw}\text{-}d\text{-}e](\boldsymbol{x}) = \neg [\mathrm{C}\mathbf{:}\mathsf{less}\text{-}d](\boldsymbol{x}) \wedge ([\mathrm{C}\mathbf{:}\mathsf{less}\text{-}e](\boldsymbol{x}) \vee [\mathrm{C}\mathbf{:}\mathsf{eq}\text{-}e](\boldsymbol{x})).$$

If $\mathfrak{A}$ is a C-structure, $a \in A$ and $d \le e \in \mathbb{B}_t$, then

$$\mathfrak{A} \vDash [\mathrm{C}\mathbf{:}\mathsf{betw}\text{-}d\text{-}e](a) \text{ iff } d \le [\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a \le e.$$

Definition 9. *Let* $d \le e \in \mathbb{B}_t$*. Define the* $\mathcal{L}^1[\mathrm{C}]$*-sentence* $[\mathrm{C}\mathbf{:}\mathsf{allbetw}\text{-}d\text{-}e]$ *by:*

$$[\mathrm{C}\mathbf{:}\mathsf{allbetw}\text{-}d\text{-}e] = \forall \boldsymbol{x} [\mathrm{C}\mathbf{:}\mathsf{betw}\text{-}d\text{-}e](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a C-structure and $d \le e \in \mathbb{B}_t$, then $\mathfrak{A} \vDash [\mathrm{C}\mathbf{:}\mathsf{betw}\text{-}d\text{-}e]$ iff $d \le [\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a \le e$ for all $a \in A$.

The bitstring $\mathrm{a} \in \mathbb{B}^t$ encodes a number less than the number encoded by the bitstring $\mathrm{b} \in \mathbb{B}^t$, if they differ and at least position where they are different $j \in [1, t]$ the bitstring a has value 0 and the bitstring b has value 1, that is, iff there is a position $j \in [1, t]$ such that the following three conditions hold:

$$\mathrm{a}_j = 0 \text{ and } \mathrm{b}_j = 1 \tag{Less1}$$

$$\mathrm{a}_k = \mathrm{b}_k \text{ for all } k \in [j+1, t]. \tag{Less2}$$

Definition 10. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{C}]$*-formula* $[\mathrm{C}{:}\mathsf{less}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{C}{:}\mathsf{less}](\boldsymbol{x}, \boldsymbol{y}) = \bigvee_{1 \le j \le t} [\boldsymbol{u}_j{:}\mathsf{eq}\text{-}01](\boldsymbol{x}, \boldsymbol{y}) \wedge \bigwedge_{j < k \le t} [\boldsymbol{u}_k{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}).$$

If $\mathfrak{A}$ is a C-structure and $a, b \in A$, then $\mathfrak{A} \models [\mathrm{C}{:}\mathsf{less}](a, b)$ iff $[\mathrm{C}{:}\mathrm{data}]^{\mathfrak{A}} a < [\mathrm{C}{:}\mathrm{data}]^{\mathfrak{A}} b$.

The bitstring $\mathrm{b} \in \mathbb{B}^t$ encodes the successor of the number encoded by the bitstring a if there is a position $j \in [1, t]$ such that the following four conditions hold:

$$\mathrm{a}_j = 0 \text{ and } \mathrm{b}_j = 1 \tag{Succ1}$$
$$\mathrm{a}_i = 1 \text{ for all } i \in [1, j-1] \tag{Succ2}$$
$$\mathrm{b}_i = 0 \text{ for all } i \in [1, j-1] \tag{Succ3}$$
$$\mathrm{a}_k = \mathrm{b}_k \text{ for all } k \in [j+1, t]. \tag{Succ4}$$

Definition 11. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{C}]$*-formula* $[\mathrm{C}{:}\mathsf{succ}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{C}{:}\mathsf{succ}](\boldsymbol{x}, \boldsymbol{y}) = \bigvee_{1 \le j \le t} \bigwedge_{1 \le i < j} [\boldsymbol{u}_i{:}\mathsf{eq}\text{-}10](\boldsymbol{x}, \boldsymbol{y}) \wedge [\boldsymbol{u}_j{:}\mathsf{eq}\text{-}01](\boldsymbol{x}, \boldsymbol{y}) \wedge \bigwedge_{j < k \le t} [\boldsymbol{u}_k{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}).$$

If $\mathfrak{A}$ is a C-structure and $a, b \in A$, then:

$$\mathfrak{A} \models [\mathrm{C}{:}\mathsf{succ}](a, b) \text{ iff } [\mathrm{C}{:}\mathrm{data}]^{\mathfrak{A}} b = 1 + [\mathrm{C}{:}\mathrm{data}]^{\mathfrak{A}} a.$$

2.3 Vectors

Let $n, t \in \mathbb{N}^+$. Recall the set of *n-dimensional t-bit vectors* is $\mathbb{B}_t^n$. An *n-dimensional t-bit vector setup* is a predicate signature $\mathrm{V} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{nt} \rangle$ of (nt) distinct unary predicate symbols. The *counter setup* $\mathrm{V}(p)$ of V at position $p \in [1, n]$ is $\mathrm{V}(p) = \langle \boldsymbol{u}_{p1}, \boldsymbol{u}_{p2}, \ldots, \boldsymbol{u}_{pt} \rangle$.

Definition 12. *Let* $\mathfrak{A}$ *be a* V*-structure and* $a \in A$*. We refer to* $[\mathrm{V}(p){:}\mathrm{data}]^{\mathfrak{A}} a$ *as the* value of the p-th counter at a. *Define the function* $[\mathrm{V}{:}\mathrm{data}]^{\mathfrak{A}} : A \to \mathbb{B}_t^n$ *by:*

$$[\mathrm{V}{:}\mathrm{data}]^{\mathfrak{A}} a = \left([\mathrm{V}(1){:}\mathrm{data}]^{\mathfrak{A}} a, [\mathrm{V}(2){:}\mathrm{data}]^{\mathfrak{A}} a, \ldots, [\mathrm{V}(n){:}\mathrm{data}]^{\mathfrak{A}} a \right).$$

Definition 13. *Let* $\mathrm{v} = (d_1, d_2, \ldots, d_n) \in \mathbb{B}_t^n$ *be an n-dimensional t-bit vector. Define the quantifier-free* $\mathcal{L}^1[\mathrm{V}]$*-formula* $[\mathrm{V}{:}\mathsf{eq}\text{-v}](\boldsymbol{x})$ *by:*

$$[\mathrm{V}{:}\mathsf{eq}\text{-v}](\boldsymbol{x}) = \bigwedge_{1 \le p \le n} [\mathrm{V}(p){:}\mathsf{eq}\text{-}d_p](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a V-structure, $a \in A$ and $\mathrm{v} \in \mathbb{B}_t^n$, then $\mathfrak{A} \models [\mathrm{V}{:}\mathsf{eq}\text{-v}](a)$ iff $[\mathrm{V}{:}\mathrm{data}]^{\mathfrak{A}} a = \mathrm{v}$.

If $\mathfrak{A}$ is a nonempty set and data : $A \to \mathbb{B}_t^n$ is any function, then there is a V-structure $\mathfrak{A}$ over A such that $[\mathrm{V}{:}\mathrm{data}]^{\mathfrak{A}} = \mathrm{data}$.

Definition 14. *Let $p, q \in [1, n]$ and let $i \in [1, t]$. Define the quantifier-free $\mathcal{L}^1[\mathrm{V}]$-formulas $[\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}](\boldsymbol{x})$, $[\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}01](\boldsymbol{x})$ and $[\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}10](\boldsymbol{x})$ by:*

$$\begin{aligned}[\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}](\boldsymbol{x}) &= \boldsymbol{u}_{pi}(\boldsymbol{x}) \leftrightarrow \boldsymbol{u}_{qi}(\boldsymbol{x})\\ [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}01](\boldsymbol{x}) &= \neg \boldsymbol{u}_{pi}(\boldsymbol{x}) \wedge \boldsymbol{u}_{qi}(\boldsymbol{x})\\ [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}10](\boldsymbol{x}) &= \boldsymbol{u}_{pi}(\boldsymbol{x}) \wedge \neg \boldsymbol{u}_{qi}(\boldsymbol{x}).\end{aligned}$$

If $\mathfrak{A}$ is a V-structure and $a \in A$, then:

- $\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}](a)$ iff $[\boldsymbol{u}_{pi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = [\boldsymbol{u}_{qi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}$, that is the values of the i-th bit at positions p and q at a are equal
- $\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}01](a)$ iff $[\boldsymbol{u}_{pi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = 0$ and $[\boldsymbol{u}_{qi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = 1$, that is the i-th bit at position p at a is 0 and the i-th bit at position q at a is 1
- $\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}10](a)$ iff $[\boldsymbol{u}_{pi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = 1$ and $[\boldsymbol{u}_{qi}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = 0$, that is the i-th bit at position p at a is 1 and the i-th bit at position q at a is 0.

Definition 15. *Let $p, q \in [1, n]$. Define the quantifier-free $\mathcal{L}^1[\mathrm{V}]$-formula $[\mathrm{V}(pq)\mathbf{:}\mathsf{eq}](\boldsymbol{x})$ by:*

$$[\mathrm{V}(pq)\mathbf{:}\mathsf{eq}](\boldsymbol{x}) = \bigwedge_{1\le i\le t} [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a V-structure and $a \in A$, then:

$$\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{eq}](a) \text{ iff } [\mathrm{V}(p)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = [\mathrm{V}(q)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a.$$

Definition 16. *Let $p, q \in [1, n]$. Define the quantifier-free $\mathcal{L}^1[\mathrm{V}]$-formula $[\mathrm{V}(pq)\mathbf{:}\mathsf{less}](\boldsymbol{x})$ by:*

$$[\mathrm{V}(pq)\mathbf{:}\mathsf{less}](\boldsymbol{x}) = \bigvee_{1\le j\le t} [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}j\text{-}\mathsf{eq}\text{-}01](\boldsymbol{x}) \wedge \bigwedge_{j<k\le t} [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}k\text{-}\mathsf{eq}](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a V-structure and $a \in A$, then:

$$\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{less}](a) \text{ iff } [\mathrm{V}(p)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a < [\mathrm{V}(q)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a.$$

Definition 17. *Let $p, q \in [1, n]$. Define the quantifier-free $\mathcal{L}^1[\mathrm{V}]$-formula*

$$\begin{aligned}[\mathrm{V}(pq)\mathbf{:}\mathsf{succ}](\boldsymbol{x}) = \bigvee_{1\le j\le t} \bigwedge_{1\le i<j} [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}i\text{-}\mathsf{eq}\text{-}10](\boldsymbol{x}) \wedge [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}j\text{-}\mathsf{eq}\text{-}01](\boldsymbol{x}) \wedge\\ \bigwedge_{j<k\le t} [\mathrm{V}(pq)\mathbf{:}\mathsf{at}\text{-}k\text{-}\mathsf{eq}](\boldsymbol{x}).\end{aligned}$$

If $\mathfrak{A}$ is a V-structure and $a \in A$, then:

$$\mathfrak{A} \models [\mathrm{V}(pq)\mathbf{:}\mathsf{succ}](a) \text{ iff } [\mathrm{V}(q)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = 1 + [\mathrm{V}(p)\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a.$$

2.4 Permutations

Let $n \in \mathbb{N}^+$. An *n-permutation setup* $\mathrm{P} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{nt} \rangle$ is just an n-dimensional t-bit vector setup, where $t = \|n\|$ is the bitsize of n. Recall that the set $\mathbb{S}_n$ of permutations of $[1, n]$ is a subset of $\mathbb{B}_t^n$.

Definition 18. *Define the quantifier-free $\mathcal{L}^1[\mathrm{P}]$-sentence* $[\mathrm{P}\mathbf{:}\mathsf{alldiff}]$ *by:*

$$[\mathrm{P}\mathbf{:}\mathsf{alldiff}] = \forall \boldsymbol{x} \bigwedge_{1 \le p < q \le n} \neg [\mathrm{P}(pq)\mathbf{:}\mathsf{eq}](\boldsymbol{x}).$$

If $\mathfrak{A}$ is a P-structure then $\mathfrak{A} \models [\mathrm{P}\mathbf{:}\mathsf{alldiff}]$ iff $[\mathrm{P}(p)\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a \neq [\mathrm{P}(q)\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a$ for all $a \in A$ and $p \neq q \in [1, n]$.

Definition 19. *Define the quantifier-free $\mathcal{L}^1[\mathrm{P}]$-sentence* $[\mathrm{P}\mathbf{:}\mathsf{perm}]$ *by:*

$$[\mathrm{P}\mathbf{:}\mathsf{perm}] = [\mathrm{P}\mathbf{:}\mathsf{betw}\text{-}1\text{-}n] \wedge [\mathrm{P}\mathbf{:}\mathsf{alldiff}].$$

If $\mathfrak{A}$ is a P-structure then $\mathfrak{A} \models [\mathrm{P}\mathbf{:}\mathsf{perm}]$ iff $[\mathrm{P}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a \in \mathbb{S}_n$ for all $a \in A$.

If A is a nonempty set and $\mathrm{data} : A \to \mathbb{S}_n$ is any function, then there is a P-structure $\mathfrak{A} \models [\mathrm{P}\mathbf{:}\mathsf{perm}]$ over A such that $[\mathrm{P}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} = \mathrm{data}$.

3 Equivalence relations

An *equivalence relation* $E \subseteq A \times A$ on A is a relation that is reflexive, symmetric and transitive. The set of *equivalence classes* of E is $\mathscr{E}E = \{E[a] \mid a \in A\}$.

Let $\mathrm{E} = \langle \boldsymbol{e} \rangle$ be a predicate signature consisting of a single binary predicate symbol $\boldsymbol{e}$. Define the $\mathcal{L}^2[\mathrm{E}]$-sentence $[\boldsymbol{e}{:}\mathsf{refl}]$ by:

$$[\boldsymbol{e}{:}\mathsf{refl}] = \forall \boldsymbol{x} \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{x}).$$

Define the $\mathcal{L}^2[\mathrm{E}]$-sentence $[\boldsymbol{e}{:}\mathsf{symm}]$ by:

$$[\boldsymbol{e}{:}\mathsf{symm}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \left(\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \to \boldsymbol{e}(\boldsymbol{y}, \boldsymbol{x}) \right).$$

Define the $\mathcal{L}^3[\mathrm{E}]$-sentence $[\boldsymbol{e}{:}\mathsf{trans}]$ by:

$$[\boldsymbol{e}{:}\mathsf{trans}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \left(\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \wedge \boldsymbol{e}(\boldsymbol{y}, \boldsymbol{z}) \to \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{z}) \right).$$

Define the $\mathcal{L}^3[\mathrm{E}]$-sentence $[\boldsymbol{e}{:}\mathsf{equiv}]$ by:

$$[\boldsymbol{e}{:}\mathsf{equiv}] = [\boldsymbol{e}{:}\mathsf{refl}] \wedge [\boldsymbol{e}{:}\mathsf{symm}] \wedge [\boldsymbol{e}{:}\mathsf{trans}].$$

Let $\mathfrak{A}$ be an E-structure and let $E = \boldsymbol{e}^{\mathfrak{A}}$. Then E is reflexive iff $\mathfrak{A} \models [\boldsymbol{e}{:}\mathsf{refl}]$; E is symmetric iff $\mathfrak{A} \models [\boldsymbol{e}{:}\mathsf{symm}]$; E is transitive iff $\mathfrak{A} \models [\boldsymbol{e}{:}\mathsf{trans}]$; E is an equivalence on A iff $\mathfrak{A} \models [\boldsymbol{e}{:}\mathsf{equiv}]$. It can be shown that transitivity and equivalence cannot be defined in the two-variable fragment with counting $\mathcal{C}^2[\mathrm{E}]$.

3.1 Two equivalence relations in agreement

Definition 20. *Let $\langle D, E \rangle$ be a sequence of two equivalence relations on A. The relation D is* finer *than the relation E if every equivalence class of D is a subset of some equivalence class of E. Equivalently, $D \subseteq E$. Equivalently,*

$$(\forall a \in A)(\forall b \in A) \left(D(a, b) \to E(a, b) \right).$$

If D is finer than E, then E is coarser *than D. The sequence $\langle D, E \rangle$ is a sequence of equivalence relations on A in* refinement *if D is finer E.*

The sequence $\langle D, E \rangle$ is a sequence of equivalence relations in global agreement *if either D is finer than E or E is finer than D.*

The sequence $\langle D, E \rangle$ is a sequence of equivalence relations in local agreement *if for every $a \in A$, either $D[a] \subseteq E[a]$ or $E[a] \subseteq D[a]$. Equivalently, no two equivalence classes $E[a]$ and $D[b]$ properly intersect. Equivalently,*

$$(\forall a \in A) \left((\forall b \in A) \left(D(a, b) \to E(a, b) \right) \vee (\forall b \in A) \left(E(a, b) \to D(a, b) \right) \right).$$

Let $\mathrm{E} = \langle \boldsymbol{d}, \boldsymbol{e} \rangle$ be a predicate signature consisting of the two binary predicate symbols $\boldsymbol{d}$ and $\boldsymbol{e}$. Let $\mathfrak{A}$ is an E-structure and suppose that $\boldsymbol{d}$ and $\boldsymbol{e}$ are interpreted in $\mathfrak{A}$ as equivalence relations on A. Let $D = \boldsymbol{d}^{\mathfrak{A}}$ and $E = \boldsymbol{e}^{\mathfrak{A}}$ be the interpretations of the two symbols.

Definition 21. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{refine}]$ *by:*

$$[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{refine}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \left(\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \right).$$

Then $\langle D, E \rangle$ is in refinement iff $\mathfrak{A} \vDash [\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{refine}]$.

Definition 22. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{global}]$ *by:*

$$[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{global}] = [\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{refine}] \vee [\boldsymbol{e}, \boldsymbol{d}{:}\mathsf{refine}].$$

Then $\langle D, E \rangle$ is in global agreement iff $\mathfrak{A} \vDash [\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{global}]$.

Definition 23. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{local}]$ *by:*

$$[\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{local}] = \forall \boldsymbol{x} \left(\forall \boldsymbol{y} \left(\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \right) \vee \forall \boldsymbol{y} \left(\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \right) \right).$$

Then $\langle D, E \rangle$ is in global agreement iff $\mathfrak{A} \vDash [\boldsymbol{d}, \boldsymbol{e}{:}\mathsf{local}]$.

Lemma 1. *If* $\langle D, E \rangle$ *is a sequence two equivalence relations on* A*, then it is in local agreement iff* $L = D \cup E$ *is an equivalence relation on* A*.*

Proof. The union of two equivalence relations on A is a reflexive and symmetric relation.

First suppose that D and E are in local agreement. We claim that L is transitive. Let $a, b, c \in A$ be such that $(a, b) \in L$ and $(b, c) \in L$. Since D and E are in local agreement, without loss of generality $D[b] \subseteq E[b]$. Since $(a, b) \in L$, either $a \in D[b] \subseteq E[b]$ or $a \in E[b]$. Similarly $c \in E[b]$. Therefore $(a, c) \in E \subseteq L$.

Next suppose that L is an equivalence relation, let $b \in A$ and assume towards a contradiction that $D[b] \nsubseteq E[b]$ and $E[b] \nsubseteq D[b]$. There is some $a \in D[b] \setminus E[b]$ and $c \in E[b] \setminus D[b]$. Then $(a, b) \in D \subseteq L$ and $(b, c) \in E \subseteq L$, hence $(a, c) \in L$. Without loss of generality $(a, c) \in E$. Since $c \in E[b]$, we have $a \in E[b]$ — a contradiction. $\square$

3.2 Many equivalence relations in agreement

Let e be a positive natural number.

Definition 24. *Let* $\langle E_1, E_2, \ldots, E_e \rangle$ *be a sequence of equivalence relations on* A*.*

The sequence is in refinement *if* $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_e$*.*

The sequence is in global agreement *if the equivalence relations form a chain under inclusion, that is for all* $i, j \in [1, e]$*, either* $E_i \subseteq E_j$ *or* $E_j \subseteq E_i$*. Equivalently, there is a (not necessarily unique) permutation* $\nu \in \mathbb{S}_e$ *such that* $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(e)}$*.*

The sequence is in local agreement *if for every element* $a \in A$ *the equivalence classes* $E_1[a], E_2[a], \ldots, E_e[a]$ *form a chain under inclusion. Equivalently, no two equivalence classes* $E_i[a]$ *and* $E_j[b]$ *properly intersect.*

Let $\mathrm{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ be a predicate signature consisting of e binary predicate symbols. Let $\mathfrak{A}$ be an E-structure and suppose that the symbols $\boldsymbol{e}_i$ are interpreted as equivalence relations on A. Let $E_i = \boldsymbol{e}_i^{\mathfrak{A}}$ for $i \in [1, e]$.

Definition 25. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:refine}]$ *by:*

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:refine}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \bigwedge_{1 \le i < e} (\boldsymbol{e}_i(\boldsymbol{x}, \boldsymbol{y}) \to \boldsymbol{e}_{i+1}(\boldsymbol{x}, \boldsymbol{y}))\,.$$

Then $\langle E_1, E_2, \ldots, E_e \rangle$ is in refinement iff $\mathfrak{A} \vDash [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:refine}]$.

Definition 26. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:global}]$ *by:*

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:global}] = \bigvee_{\nu \in \mathbb{S}_e} [\boldsymbol{e}_{\nu(1)}, \boldsymbol{e}_{\nu(2)}, \ldots, \boldsymbol{e}_{\nu(e)}\textsf{:refine}].$$

Then $\langle E_1, E_2, \ldots, E_e \rangle$ is in global agreement iff $\mathfrak{A} \vDash [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:global}]$. Note that the length of the formula $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:global}]$ grows exponentially as e grows.

Definition 27. *Define the* $\mathcal{L}^2[\mathrm{E}]$*-sentence* $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:local}]$ *by:*

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:local}] = \forall \boldsymbol{x} \bigvee_{\nu \in \mathbb{S}_e} \forall \boldsymbol{y} \bigwedge_{1 \le i < e} (\boldsymbol{e}_{\nu(i)}(\boldsymbol{x}, \boldsymbol{y}) \to \boldsymbol{e}_{\nu(i+1)}(\boldsymbol{x}, \boldsymbol{y})).$$

Then $\langle E_1, E_2, \ldots, E_e \rangle$ is in local agreement iff $\mathfrak{A} \vDash [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:local}]$. Note that the length of the formula $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\textsf{:local}]$ grows exponentially as e grows.

Let $E = \langle E_1, E_2, \ldots, E_e \rangle$ be a sequence of equivalence relations on A.

Theorem 4. *The sequence E is in local agreement iff the union $\cup S$ of any nonempty subsequence $S \subseteq E$ is an equivalence relation on A.*

Proof. First suppose that the equivalence relations E_i are in local agreement. We show that the union $\cup S$ of arbitrary nonempty subsequence $S = \{E_{i(1)}, E_{i(2)}, \ldots, E_{i(s)}\}$, where $1 \le i(1) < i(2) < \cdots < i(s) \le e$, is an equivalence relation by induction on s, the length of S. If $s = 1$ this claim is trivial. Suppose $s > 1$. By the induction hypothesis, $D = \cup\{E_{i(1)}, E_{i(2)}, \ldots, E_{i(s-1)}\}$ is an equivalence relation on A. We claim that D and $E_{i(s)}$ are in local agreement. Indeed, let $a \in A$ be arbitrary and consider $D[a] = E_{i(1)}[a] \cup E_{i(2)}[a] \cup \cdots \cup E_{i(s-1)}[a]$ and $E_{i(s)}[a]$. Since all equivalences E_k are in local agreement, either $E_{i(s)}[a] \subseteq E_{i(j)}[a]$ for some $j \in [1, s-1]$, or $E_{i(j)}[a] \subseteq E_{i(s)}[a]$ for all $j \in [1, s-1]$. In the first case $E_{i(s)}[a] \subseteq D[a]$; in the second case $D[a] \subseteq E_{i(s)}[a]$. Thus D and $E_{i(s)}$ are in local agreement. By Lemma 1, $\cup S = D \cup E_{i(s)}$ is an equivalence relation on A.

Next suppose that the equivalences are not in local agreement. There is an element $a \in A$ such that $\{E_i[a] \mid i \in [1, e]\}$ is not a chain. There are $i, j \in [1, e]$ such that $E_i[a] \nsubseteq E_j[a]$ and $E_j[a] \nsubseteq E_i[a]$. Thus E_i and E_j are not in local agreement. By Lemma 1, the union $E_i \cup E_j$ is not an equivalence relation on A. $\square$

Suppose that the sequence $E = \langle E_1, E_2, \ldots, E_e \rangle$ is in local agreement.

Definition 28. *An* index set *is an element* $\mathrm{I} \in \wp^{+}[1,e]$*. Define* $(E \restriction \cdot) : \wp^{+}[1,e] \to \wp^{+}E$ *by:*

$$(E \restriction \mathrm{I}) = \{E_i \mid i \in \mathrm{I}\}.$$

That is, $(E \restriction \mathrm{I})$ *just collects the equivalences having indices from* I*.*

The level sequence $L = \langle L_1, L_2, \ldots, L_e\rangle$ *of the sequence* E *is defined as follows. For* $k \in [1,e]$*:*

$$L_k = \cap \left\{ \cup(E \restriction \mathrm{I}) \;\middle|\; \mathrm{I} \in \wp^{k}[1,e] \right\}.$$

Remark 2. *All* L_k *are equivalence relations on* A*.*

Proof. Let $k \in [1,e]$ and let $\mathrm{K} \in \wp^{k}[1,e]$ be any k-index set. By Theorem 4, $\cup(E \restriction \mathrm{K})$ is an equivalence relation on A. Since intersection of equivalence relations on A is again an equivalence relation on A, the level $L_k = \cap \left\{ \cup(E \restriction \mathrm{K}) \;\middle|\; \mathrm{K} \in \wp^{k}[1,e] \right\}$ is an equivalence relation on A. ☐

Remark 3. *The level sequence* $L = \langle L_1, L_2, \ldots, L_e\rangle$ *is a sequence of equivalence relations on* A *in refinement.*

Proof. Let $i < j \in [1,e]$. Let $\mathrm{J} \in \wp^{j}[1,e]$ be any j-index set. We claim that $L_i \subseteq \cup(E \restriction \mathrm{J})$. Indeed, choose some i-index set $\mathrm{I} \subset \mathrm{J}$. By the definition of L_i we have $L_i \subseteq \cup(E \restriction \mathrm{I}) \subseteq \cup(E \restriction \mathrm{J})$. Hence $L_i \subseteq \cap\{\cup(E \restriction \mathrm{J}) \mid \mathrm{J} \in \wp^{j}[1,e]\} = L_j$. ☐

Let $a \in A$. Since the sequence $E = \langle E_1, E_2, \ldots, E_e\rangle$ is in local agreement, there is a permutation $\nu \in \mathbb{S}_e$ such that:

$$E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \cdots \subseteq E_{\nu(e)}[a]. \tag{3.1}$$

Lemma 2. *If* $\nu \in \mathbb{S}_e$ *is a permutation satisfying eq. (3.1), then* $L_{\nu^{-1}(i)}[a] = E_i[a]$ *for all* $i \in [1,e]$*.*

Proof. Let $k = \nu^{-1}(i)$, so $\nu(k) = i$. We claim that $L_k[a] = E_i[a]$. First, consider the k-index set $\mathrm{K} = \{\nu(1), \nu(2), \ldots, \nu(k)\}$. By the definition of L_k, followed by eq. (3.1), we have $L_k[a] \subseteq \cup(E \restriction \mathrm{K})[a] = E_{\nu(k)}[a] = E_i[a]$. Next, let $\mathrm{K} \subseteq \wp^{k}[1,e]$ be any k-index set. By the pigeonhole principle, there is some $k' \geq k$ such that $k' \in \mathrm{K}$. By eq. (3.1) we have:

$$E_i[a] = E_{\nu(k)}[a] \subseteq E_{\nu(k')}[a] \subseteq \cup(E \restriction \mathrm{K})[a].$$

Hence $E_i[a] \subseteq \cap \left\{ \cup(E \restriction \mathrm{K})[a] \;\middle|\; \mathrm{K} \in \wp^{k}[1,e] \right\} = L_k[a]$. ☐

4 Reductions

We restrict our attention to binary predicate signatures, consisting of unary and binary predicate symbols only. To denote various logics with builtin equivalence symbols, we use the notation

$$\Lambda_p^v e \mathrm{E}_{\mathsf{a}}$$

where:

- $\Lambda \in \{\mathcal{L}, \mathcal{C}\}$ is the *ground logic*
- v, if given, bounds the number of variables
- e, if given, bounds the number of builtin equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{global}, \mathsf{local}\}$, if given, gives the agreement condition between the builtin equivalence symbols
- p, the *signature power*, specifies constraints on the signature:
  - if $p = 0$, the signature consists of only constantly many unary predicate symbols in addition to the builtin equivalence symbols
  - if $p = 1$, the signature consists of unboundedly many unary predicate symbols in addition to the builtin equivalence symbols
  - if p is not given, the signature consists of unboundedly many unary and binary predicate symbols in addition to the builtin equivalence symbols. This is the commonly investigated fragment with respect to satisfiability of the two-variable logics with or without counting quantifiers.

For example $\mathcal{L}_1$ is the monadic first-order logic, featuring only unary predicate symbols. $\mathcal{L}_0 1\mathrm{E}$ is the first-order logic of a single equivalence relation. $\mathcal{C}^2$ is the two-variable logic with counting quantifiers, featuring unary and binary predicate symbols. $\mathcal{L}^2 2\mathrm{E}$ is the two-variable logic, featuring unary, binary predicate symbols and two builtin equivalence symbols. $\mathcal{C}_1^2 2\mathrm{E}_{\mathsf{local}}$ is the two-variable logic with counting quantifiers, featuring unary predicate symbols and two builtin equivalence symbols in local agreement. $\mathcal{L}_1 \mathrm{E}_{\mathsf{global}}$ is the monadic first-order logic featuring many equivalence symbols in global agreement.

When we working with a concrete logic, for example $\mathcal{C}_2^2 2\mathrm{E}_{\mathsf{local}}$, we implicitly assume an appropriate generic predicate signature Σ for it. In this case, there are two builtin equivalence symbols $\boldsymbol{d}$ and $\boldsymbol{e}$ in Σ and in addition Σ contains arbitrary many unary and binary predicate symbols. The *intended interpretation* of the builtin equivalence symbols is fixed by an appropriate condition θ. In this case:

$$\theta = [\boldsymbol{d}\mathsf{:equiv}] \wedge [\boldsymbol{e}\mathsf{:equiv}] \wedge [\boldsymbol{d}, \boldsymbol{e}\mathsf{:local}].$$

Note that the interpretation condition might in general be a first-order formula outside the logic in interest, as in this case, since for instance $[\boldsymbol{d}\mathsf{:equiv}]$ uses the variables $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{z}$ and the logic $\mathcal{C}^2_2 2\mathrm{E}_{\mathsf{local}}$ is a two-variable logic. Recall that when talking about semantics, we include the intended interpretation condition in the definition of Σ-structures.

4.1 Global agreement to refinement

In this section we demonstrate how (finite) satisfiability in logics featuring builtin equivalence symbols in global agreement reduces to (finite) satisfiability in logics featuring builtin equivalence symbols in refinement. Our strategy is to encode the permutation of the builtin equivalence symbols in global agreement that turns them in refinement into a permutation setup.

Fix an arbitrary ground logic $\Lambda \in \{\mathcal{L}, \mathcal{C}\}$ and think of Σ as a predicate signature for the logics $\Lambda e\mathrm{E}_{\mathsf{global}}$ and $\Lambda e\mathrm{E}_{\mathsf{refine}}$. The e builtin equivalence symbols of Σ are $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$.

Let φ be a $\Lambda[\Sigma]$-sentence. The class of $\Lambda e\mathrm{E}_{\mathsf{refine}}$-structures satisfying φ coincides with the class of $\Lambda e\mathrm{E}_{\mathsf{global}}$-structures satisfying:

$$\varphi \wedge [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\mathsf{:refine}].$$

Hence:

$$(\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{refine}} \leq_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{global}}.$$

Since the length of the formula $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\mathsf{:refine}]$ grows polynomially as e grows:

$$(\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{refine}} \leq_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{global}}.$$

Consider the opposite direction. Let $\mathrm{P} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{et} \rangle$ be an e-permutation setup (where $t = \|e\|$).

Definition 29. *Define the* $\mathcal{L}^2[\mathrm{P}]$*-sentence* $[\mathrm{P}\mathsf{:alleq}]$ *by:*

$$[\mathrm{P}\mathsf{:alleq}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \bigwedge_{1 \leq i \leq e} [\mathrm{P}\mathsf{:eq}\text{-}i](\boldsymbol{x}, \boldsymbol{y}).$$

If $\mathfrak{A}$ is a P-structure, then $\mathfrak{A} \models [\mathrm{P}\mathsf{:alleq}]$ iff $[\mathrm{P}\mathsf{:}\mathrm{data}]^{\mathfrak{A}} a = [\mathrm{P}\mathsf{:}\mathrm{data}]^{\mathfrak{A}} b$ for all $a, b \in A$. If A is a nonempty set and $\mathrm{v} \in \mathbb{B}^e_t$ is any e-dimensional t-vector, there is a P-structure $\mathfrak{A}$ over A such that $\mathfrak{A} \models [\mathrm{P}\mathsf{:alleq}]$ and $[\mathrm{P}\mathsf{:}\mathrm{data}]^{\mathfrak{A}} a = \mathrm{v}$ for all $a \in A$.

Definition 30. *Define the* $\mathcal{L}^2[\mathrm{P}]$*-sentence* $[\mathrm{P}\mathsf{:globperm}]$ *by:*

$$[\mathrm{P}\mathsf{:globperm}] = [\mathrm{P}\mathsf{:perm}] \wedge [\mathrm{P}\mathsf{:alleq}].$$

If $\mathfrak{A}$ be a P-structure then $\mathfrak{A} \vDash [\mathrm{P}\mathbf{:}\mathsf{globperm}]$ iff there is a permutation $\nu \in \mathbb{S}_e$ such that $[\mathrm{P}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = \nu$ for all $a \in A$.

If A be a nonempty set and $\nu \in \mathbb{S}_e$ is any permutation, there is a P-structure $\mathfrak{A}$ over A such that $\mathfrak{A} \vDash [\mathrm{P}\mathbf{:}\mathsf{globperm}]$ and $[\mathrm{P}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = \nu$ for all $a \in A$.

Let $\mathrm{L} = \langle \boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e \rangle + \mathrm{P}$ be a predicate signature consisting of the binary predicate symbols $\boldsymbol{l}_k$ in addition to the symbols from P.

Definition 31. *For $i \in [1, e]$, define the quantifier-free $\mathcal{L}^2[\mathrm{L}]$-formula $[\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i](\boldsymbol{x}, \boldsymbol{y})$ by:*

$$[\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i](\boldsymbol{x}, \boldsymbol{y}) = \bigwedge_{1 \le k \le e} \left([\mathrm{P}\mathbf{:}\mathsf{eq}\text{-}k\text{-}i](\boldsymbol{x}) \to \boldsymbol{l}_k(\boldsymbol{x}, \boldsymbol{y}) \right).$$

Remark 4. *Let $\mathfrak{A}$ be an* L*-structure and suppose that $\mathfrak{A} \vDash [\mathrm{P}\mathbf{:}\mathsf{globperm}]$ and that the binary symbols $\boldsymbol{l}_k$ are interpreted as equivalence relations on A in refinement. Recall that there is a permutation $\nu \in \mathbb{S}_e$ such that $[\mathrm{P}\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = \nu$ for all $a \in A$. Then for all $i \in [1, e]$:*

$$[\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i]^{\mathfrak{A}} = \boldsymbol{l}^{\mathfrak{A}}_{\nu^{-1}(i)}.$$

In particular, $\left\langle [\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}1]^{\mathfrak{A}}, [\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}2]^{\mathfrak{A}}, \ldots, [\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}e]^{\mathfrak{A}} \right\rangle$ is a sequence of equivalence relations on A in global agreement.

Proof. Let $k = \nu^{-1}(i)$, so $\nu(k) = i$ and $[\mathrm{P}(k)\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = i$. Since ν is a permutation, for every $k' \in [1, e]$:

$$\mathfrak{A} \vDash [\mathrm{P}\mathbf{:}\mathsf{eq}\text{-}k'\text{-}i](a) \text{ iff } [\mathrm{P}(k')\mathbf{:}\mathrm{data}]^{\mathfrak{A}} a = i \text{ iff } k' = k. \tag{4.1}$$

Let $a, b \in A$. First suppose that $\mathfrak{A} \vDash [\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i](a, b)$. By eq. (4.1) we have $\mathfrak{A} \vDash [\mathrm{P}\mathbf{:}\mathsf{eq}\text{-}k\text{-}i](a)$, hence $\mathfrak{A} \vDash \boldsymbol{l}_k(a, b)$.

Now suppose that $\mathfrak{A} \vDash \neg[\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i](a, b)$. There is some $k' \in [1, e]$ such that:

$$\mathfrak{A} \vDash \neg \left([\mathrm{P}\mathbf{:}\mathsf{eq}\text{-}k'\text{-}i](a) \to \boldsymbol{l}_{k'}(a, b) \right) \equiv [\mathrm{P}\mathbf{:}\mathsf{eq}\text{-}k'\text{-}i](a) \wedge \neg \boldsymbol{l}_{k'}(a, b).$$

By eq. (4.1) we have $k' = k$, hence $\mathfrak{A} \vDash \neg \boldsymbol{l}_k(a, b)$. □

Let $\mathrm{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ be a predicate signature consisting of the binary predicate symbols $\boldsymbol{e}_i$. Let Σ be a predicate signature enriching E and not containing any symbols from L. Let $\Sigma' = \Sigma \cup \mathrm{L}$ and $\mathrm{L}' = \Sigma' - \mathrm{E}$.

Definition 32. *Define the syntactic operation $\mathrm{gtr} : \Lambda[\Sigma] \to \Lambda[\mathrm{L}']$ by:*

$$\mathrm{gtr}\, \varphi = \varphi' \wedge [\mathrm{P}\mathbf{:}\mathsf{globperm}],$$

where φ' is obtained from φ by replacing all occurrences of a subformula of the form $\boldsymbol{e}_i(x, y)$ by the formula $[\mathrm{L}\mathbf{:}\mathsf{eg}\text{-}i](x, y)$, where x and y are (not necessarily distinct) variables and $i \in [1, e]$.

Remark 5. *Let φ be a $\Lambda[\Sigma]$-formula and let $\mathfrak{A}$ be a Σ-structure. Suppose that $\mathfrak{A} \vDash \varphi$ and that the symbols $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$ are interpreted in $\mathfrak{A}$ as equivalence relations on A in global agreement. Then there is a Σ'-enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \operatorname{gtr}\varphi$ and that the symbols $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e$ are interpreted in $\mathfrak{A}'$ as equivalence relations on A in refinement.*

Proof. There is a permutation $\nu \in \mathbb{S}_e$ such that $\boldsymbol{e}^{\mathfrak{A}}_{\nu(1)} \subseteq \boldsymbol{e}^{\mathfrak{A}}_{\nu(2)} \subseteq \cdots \subseteq \boldsymbol{e}^{\mathfrak{A}}_{\nu(e)}$. Consider an enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ to a Σ'-structure where $\boldsymbol{l}^{\mathfrak{A}'}_k = \boldsymbol{e}^{\mathfrak{A}}_{\nu(k)}$, so the interpretations of $\boldsymbol{l}_k$ in $\mathfrak{A}'$ are equivalence relations on A in refinement. We can interpret the unary predicate symbols from permutation setup P in $\mathfrak{A}'$ so that $\mathfrak{A}' \vDash [\mathrm{P}{:}\mathsf{globperm}]$ and $[\mathrm{P}{:}\mathrm{data}]^{\mathfrak{A}} a = \nu$ for all $a \in A$. By Remark 4, for every $i \in [1, e]$:

$$[\mathrm{L}{:}\mathsf{eg}\text{-}i]^{\mathfrak{A}'} = \boldsymbol{l}^{\mathfrak{A}'}_{\nu^{-1}(i)} = \boldsymbol{e}^{\mathfrak{A}'}_{\nu(\nu^{-1}(i))} = \boldsymbol{e}^{\mathfrak{A}'}_i = \boldsymbol{e}^{\mathfrak{A}}_i.$$

Hence $\mathfrak{A}' \vDash \forall \boldsymbol{x} \forall \boldsymbol{y}\, (\boldsymbol{e}_i(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\mathrm{L}{:}\mathsf{eg}\text{-}i](\boldsymbol{x}, \boldsymbol{y}))$. Since $\mathfrak{A}' \vDash \varphi$ we have $\mathfrak{A}' \vDash \operatorname{gtr}\varphi$. □

Remark 6. *Let φ be a $\Lambda[\Sigma]$-formula and let $\mathfrak{A}$ be an L'-structure. Suppose that $\mathfrak{A} \vDash \operatorname{gtr}\varphi$ and that the symbols $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e$ are interpreted in $\mathfrak{A}$ as equivalence relations on A in refinement. Then there is a Σ'-enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \varphi$ and that the symbols $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$ are interpreted as equivalence relations on A in global agreement in $\mathfrak{A}'$.*

Proof. Consider an enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ to a Σ'-structure where $\boldsymbol{e}^{\mathfrak{A}'}_i = [\mathrm{L}{:}\mathsf{eg}\text{-}i]^{\mathfrak{A}}$. By Remark 4, $\langle \boldsymbol{e}^{\mathfrak{A}'}_1, \boldsymbol{e}^{\mathfrak{A}'}_2, \ldots, \boldsymbol{e}^{\mathfrak{A}'}_e \rangle$ is a sequence of equivalence relations on A in global agreement. For every $i \in [1, e]$ we have $\mathfrak{A}' \vDash (\forall \boldsymbol{x} \forall \boldsymbol{y} \boldsymbol{e}_i(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\mathrm{L}{:}\mathsf{eg}\text{-}i](\boldsymbol{x}, \boldsymbol{y}))$ by definition. Since $\mathfrak{A}' \vDash \operatorname{gtr}\varphi$ we have $\mathfrak{A}' \vDash \varphi$. □

The last two remarks show that a $\Lambda e\mathrm{E}_{\mathsf{global}}$-formula φ has essentially the same models as the $\Lambda e\mathrm{E}_{\mathsf{refine}}$-formula $\operatorname{gtr}\varphi$, so we have shown:

Proposition 1. *The logic $\Lambda e\mathrm{E}_{\mathsf{global}}$ has the finite model property iff the logic $\Lambda e\mathrm{E}_{\mathsf{refine}}$ has the finite model property. The corresponding satisfiability problems are polynomial-time equivalent:* $(\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{global}} =^{\mathrm{PTIME}}_{\mathrm{m}} (\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{refine}}$.

Since the relative size of $\operatorname{gtr}\varphi$ with respect to φ grows polynomially as e grows, we have shown:

Proposition 2. *The logic $\Lambda\mathrm{E}_{\mathsf{global}}$ has the finite model property iff the logic $\Lambda\mathrm{E}_{\mathsf{refine}}$ has the finite model property. The corresponding satisfiability problems are polynomial-time equivalent:* $(\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{global}} =^{\mathrm{PTIME}}_{\mathrm{m}} (\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{refine}}$.

The reduction is two-variable first-order and uses additional (et) unary predicate symbols for the permutation setup P, so it is also valid for the two-variable fragments $\Lambda^2_0 e\mathrm{E}_{\mathsf{a}}$, $\Lambda^2_1 e\mathrm{E}_{\mathsf{a}}$ and $\Lambda^2_1\mathrm{E}_{\mathsf{a}}$ for $\mathsf{a} \in \{\mathsf{global}, \mathsf{refine}\}$.

4.2 Local agreement to refinement

In this section demonstrate how (finite) satisfiability in logics featuring builtin equivalence symbols in local agreement reduces to (finite) satisfiability in logics featuring builtin equivalence symbols in refinement. Our strategy is to start with the level equivalences which form a refinement, and to encode a permutation specifying the local chain structure for every element in the structure.

Fix an arbitrary ground logic $\Lambda \in \{\mathcal{L}, \mathcal{C}\}$ and think of Σ as a predicate signature for the logics $\Lambda e\mathrm{E}_{\mathsf{local}}$ and $\Lambda e\mathrm{E}_{\mathsf{refine}}$. The e builtin equivalence symbols of Σ are $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$.

Let φ be a $\Lambda[\Sigma]$-sentence. The class of $\Lambda e\mathrm{E}_{\mathsf{refine}}$-structures satisfying φ coincides with the class of $\Lambda e\mathrm{E}_{\mathsf{local}}$-structures satisfying

$$\varphi \wedge [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\text{:}\mathsf{refine}].$$

Hence:

$$(\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{refine}} \leq_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{local}}.$$

Since the size of the formula $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e\text{:}\mathsf{refine}]$ grows polynomially as e grows, we have:

$$(\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{refine}} \leq_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{local}}.$$

Consider the opposite direction. Let $\mathrm{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ be a predicate signature consisting of the binary predicate symbols $\boldsymbol{e}_i$ (not necessarily interpreted as equivalences). Let $\mathfrak{A}$ be an E-structure and suppose that the symbols $\boldsymbol{e}_i$ are interpreted in $\mathfrak{A}$ as equivalence relations on A in local agreement. Let $E_i = \boldsymbol{e}_i^{\mathfrak{A}}$ for $i \in [1, e]$. Recall that for every $a \in A$ there is a permutation $\nu \in \mathbb{S}_e$ satisfying eq. (3.1):

$$E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \cdots \subseteq E_{\nu(e)}[a]. \tag{4.2}$$

Definition 33. *The* characteristic E-permutation *of a in $\mathfrak{A}$ is the lexicographically smallest permutation $\nu \in \mathbb{S}_e$ satisfying eq. (4.2). Define the function $[\mathrm{E}\text{:}\mathrm{chperm}]^{\mathfrak{A}} : A \to \mathbb{S}_e$ so that $[\mathrm{E}\text{:}\mathrm{chperm}]^{\mathfrak{A}}a$ is the characteristic* E*-permutation of a in $\mathfrak{A}$.*

Remark 7. *Let $a \in A$, $\nu = [\mathrm{E}\text{:}\mathrm{chperm}]^{\mathfrak{A}}a$ and $i < j \in [1, e]$. Suppose that $E_{\nu(i)}[a] = E_{\nu(j)}[a]$. Then $\nu(i) < \nu(j)$.*

Proof. Suppose not. For some $i < j \in [1, e]$ we have $\nu(i) \geq \nu(j)$. Since ν is a permutation and $i \neq j$, we have $\nu(i) > \nu(j)$. Since $E_{\nu(i)}[a] = E_{\nu(j)}[a]$, by eq. (4.2) we have $E_{\nu(k)} = E_{\nu(i)}$ for all $k \in [i, j]$. Consider the permutation $\mu \in \mathbb{S}_e$ defined by:

$$\mu(k) = \begin{cases} \nu(j) & \text{if } k = i \\ \nu(i) & \text{if } k = j \\ \nu(k) & \text{otherwise.} \end{cases}$$

Clearly, μ is a permutation satisfying eq. (4.2) that is lexicographically smaller than ν — a contradiction. $\square$

Remark 8. *Let $a, b \in A$ and let $\alpha = [\text{E:chperm}]^{\mathfrak{A}} a$ and $\beta = [\text{E:chperm}]^{\mathfrak{A}} b$. Let $i \in [1, e]$ and suppose that $(a, b) \in E_i$. Then $\alpha^{-1}(i) = \beta^{-1}(i)$.*

Proof. Suppose not, so $\alpha^{-1}(i) \neq \beta^{-1}(i)$. Let $p = \alpha^{-1}(i)$ and $q = \beta^{-1}(i)$. Without loss of generality, suppose that $p < q$. Thus p is the position of i in the permutation α and $q > p$ is the position of i in the permutation β. By the pigeonhole principle, there is $k \in [1, e]$ that occurs after i in α and before j in β: $p < \alpha^{-1}(k)$ and $\beta^{-1}(k) < q$. Since β is the characteristic E-permutation of b in $\mathfrak{A}$, by eq. (4.2) we have $E_k[b] \subseteq E_i[b]$. Since $(a, b) \in E_i$, we have $E_k[b] \subseteq E_i[a]$. Since $E_k[b] \subseteq E_i[a]$ are equivalence classes, $E_k[a] \subseteq E_i[a]$. Since k occurs after i in α, which is the characteristic E-permutation of a in $\mathfrak{A}$, by eq. (4.2) we have $E_k[a] = E_i[a]$. By Remark 7, $i < k$. By the contrapositive of Remark 7, $E_k[b] = E_i[b]$ is impossible. Since k occurs before i in β, by eq. (4.2) we have $E_k[b] \subset E_i[b]$. Hence

$$E_k[b] \subset E_i[b] = E_i[a] = E_k[a]$$

— a contradiction — since the equivalence classes $E_k[b]$ and $E_k[a]$ are either equal or disjoint. □

Let $L = \langle L_1, L_2, \ldots, L_e \rangle \subseteq A \times A$ be the levels of $E = \langle E_1, E_2, \ldots, E_e \rangle$. Recall that by Remark 3, the levels are equivalence relations on A in refinement.

Remark 9. *Let $a \in A$, $\alpha = [\text{E:chperm}]^{\mathfrak{A}} a$ and let $k \in [1, e]$. Then $L_k[a] = E_{\alpha(k)}[a]$.*

Proof. Since α satisfies eq. (4.2), by Lemma 2:

$$L_k[a] = L_{\alpha^{-1}(\alpha(k))}[a] = E_{\alpha(k)}[a].$$

□

Remark 10. *Let $a, b \in A$, $\alpha = [\text{E:chperm}]^{\mathfrak{A}} a$, $\beta = [\text{E:chperm}]^{\mathfrak{A}} b$ and $k \in [1, e]$. Suppose that $(a, b) \in L_k$. Then $\alpha(k) = \beta(k)$. That is, the elements connected at level k agree at position k in their characteristic permutations.*

Proof. By Remark 9, $L_k[a] = E_{\alpha(k)}[a]$, thus $(a, b) \in E_{\alpha(k)}$. By Remark 7,

$$k = \alpha^{-1}(\alpha(k)) = \beta^{-1}(\alpha(k)).$$

Hence $\beta(k) = \alpha(k)$. □

Let $\mathrm{P} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{et} \rangle$ be an e-permutation setup. Let $\mathrm{L} = \langle \boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e \rangle + \mathrm{P}$ be a predicate signature containing the binary predicate symbols $\boldsymbol{l}_k$ (not necessarily interpreted as equivalence relations) together with the symbols from P.

Definition 34. *Define the $\mathcal{L}^2[\mathrm{L}]$-sentence* [L:fixperm] *by:*

$$[\mathsf{L{:}fixperm}] = \forall \boldsymbol{x} \forall \boldsymbol{y} \bigwedge_{1 \leq k \leq e} (\boldsymbol{l}_k(\boldsymbol{x}, \boldsymbol{y}) \to [\mathrm{P}(k){:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}))\,.$$

Definition 35. *Define the* $\mathcal{L}^2[\mathrm{L}]$*-sentence* [L:locperm] *by:*

$$[\mathsf{L{:}locperm}] = [\mathsf{P{:}perm}] \wedge [\mathsf{L{:}fixperm}].$$

Remark 11. *Let* $\mathfrak{A}$ *be an* L*-structure and suppose that* $\mathfrak{A} \models [\mathsf{L{:}locperm}]$. *Let* $a, b \in A$, $k \in [1, e]$ *and suppose that* $\mathfrak{A} \models \boldsymbol{l}_k(a, b)$. *Let* $\alpha = [\mathrm{P{:}data}]^{\mathfrak{A}} a$ *and* $\beta = [\mathrm{P{:}data}]^{\mathfrak{A}} b$ *be the* e*-permutations at* a *and* b*,* encoded by the permutation setup P. *Then* $\alpha(k) = \beta(k)$.

Proof. Since $\mathfrak{A} \models [\mathsf{L{:}fixperm}]$ and $\mathfrak{A} \models \boldsymbol{l}_k(a, b)$, we have $\mathfrak{A} \models [\mathsf{P{:}eq}\text{-}k](a, b)$, which means $\alpha(k) = \beta(k)$. ☐

Definition 36. *For* $i \in [1, e]$*, define the quantifier-free* $\mathcal{L}^2[\mathrm{L}]$*-formula* [L:el-$i$] *by:*

$$[\mathsf{L{:}el}\text{-}i](\boldsymbol{x}, \boldsymbol{y}) = \bigwedge_{1 \le k \le n} \left([\mathsf{L{:}eq}\text{-}k\text{-}i](\boldsymbol{x}) \to \boldsymbol{l}_k(\boldsymbol{x}, \boldsymbol{y})\right).$$

Remark 12. *Let* $\mathfrak{A}$ *be an* L*-structure and suppose that* $\mathfrak{A} \models [\mathsf{L{:}locperm}]$ *and that the binary symbols* $\boldsymbol{l}_k$ *are interpreted in* $\mathfrak{A}$ *as equivalence relations on* A *in refinement. Define* $\nu : A \to \mathbb{S}_e$ *by* $\nu(a) = [\mathrm{P{:}data}]^{\mathfrak{A}} a$ *for* $a \in A$. *Let* $a \in A$ *be arbitrary. Then for all* $i \in [1, e]$*:*

$$[\mathsf{L{:}el}\text{-}i]^{\mathfrak{A}}[a] = \boldsymbol{l}^{\mathfrak{A}}_{\nu(a)^{-1}(i)}[a].$$

Proof. Let $E_i = [\mathsf{L{:}el}\text{-}i]^{\mathfrak{A}}$ and $L_i = \boldsymbol{l}_i^{\mathfrak{A}}$ for every $i \in [1, e]$. Let $i \in [1, e]$ be arbitrary. Let $\alpha = \nu(a)$ and $k = \alpha^{-1}(i)$, so $\alpha = [\mathrm{P{:}data}]^{\mathfrak{A}} a$ and $\alpha(k) = i$. We have to show that $E_i[a] = L_k[a]$. Since α is a permutation, for every $k' \in [1, e]$ we have:

$$\mathfrak{A} \models [\mathsf{P{:}eq}\text{-}k'\text{-}i](a) \text{ iff } \alpha(k') = i \text{ iff } k' = k. \tag{4.3}$$

First, suppose $b \in E_i[a]$. Then $\mathfrak{A} \models [\mathsf{L{:}el}\text{-}i](a, b)$ and by eq. (4.3) we have $\mathfrak{A} \models \boldsymbol{l}_k(a, b)$, hence $b \in L_k[a]$.

Next, suppose $b \notin E_i[a]$. Then $\mathfrak{A} \models \neg[\mathsf{L{:}el}\text{-}i](a, b)$, so there is some $k' \in [1, e]$ such that $\mathfrak{A} \models \neg([\mathsf{L{:}eq}\text{-}k'\text{-}i](a) \to \boldsymbol{l}_{k'}(a, b)) \equiv [\mathsf{L{:}eq}\text{-}k'\text{-}i](a) \wedge \neg\boldsymbol{l}_{k'}(a, b)$. By eq. (4.3) we have $k' = k$. Hence $\mathfrak{A} \models \neg\boldsymbol{l}_k(a, b)$, so $b \notin L_k[a]$. ☐

Remark 13. *Let* $\mathfrak{A}$ *and* ν *are declared as in Remark 12. Then the sequence of interpretations* $\langle [\mathsf{L{:}el}\text{-}1]^{\mathfrak{A}}, [\mathsf{L{:}el}\text{-}2]^{\mathfrak{A}}, \ldots, [\mathsf{L{:}el}\text{-}e]^{\mathfrak{A}} \rangle$ *is a sequence of equivalence relations on* A *in local agreement.*

Proof. Let $E_i = [\mathsf{L{:}el}\text{-}i]^{\mathfrak{A}}$ and $L_i = \boldsymbol{l}_i^{\mathfrak{A}}$ for every $i \in [1, e]$. Let $i \in [1, e]$ be arbitrary. We check that E_i is reflexive, symmetric and transitive.

- For reflexivity, let $a \in A$. By Remark 12, $E_i[a] = L_k[a]$ for $k = \nu(a)^{-1}(i)$. But $L_k[a]$ is an equivalence class, hence $a \in L_k[a]$, so $(a, a) \in E_i$.
- For symmetry, let $a, b \in A$ and $(a, b) \in E_i$. Let $k = \nu(a)^{-1}(i)$ so that $i = \nu(k)$. By Remark 12, $E_i[a] = L_k[a]$. Thus $\mathfrak{A} \models \boldsymbol{l}_k(a, b)$ and by Remark 11, $i = \nu(a)(k) = \nu(b)(k)$. By Remark 12:

  $$E_i[b] = [\mathsf{L{:}el}\text{-}i]^{\mathfrak{A}}[b] = \boldsymbol{l}^{\mathfrak{A}}_{\nu(b)^{-1}(i)}[b] = L_k[b] = L_k[a].$$

  Since $a \in L_k[a] = E_i[b]$, we have $(b, a) \in E_i$.

- For transitivity, continue the argument for symmetry. Let $c \in E_i[b]$. Then $c \in E_i[b] = L_k[a] = E_i[a]$, thus $(a, c) \in E_i$.

By Remark 12, since the relations L_k are in refinement, we have that $E_1, E_2, \ldots, E_e$ are in local agreement. $\square$

Let $\mathrm{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ be a predicate signature consisting of binary predicate symbols. Let Σ be a predicate signature enriching E and not containing any symbols from L. Let $\Sigma' = \Sigma + \mathrm{L}$ and $\mathrm{L}' = \Sigma' - \mathrm{E}$.

Definition 37. *Define the syntactic operation* ltr $: \Lambda[\Sigma] \to \Lambda[\mathrm{L}']$ *by:*

$$\operatorname{ltr} \varphi = \varphi' \wedge [\mathrm{L}\textbf{:}\mathsf{locperm}],$$

where φ' is obtained from φ by replacing all occurrences of a subformula of the form $\boldsymbol{e}_i(x, y)$ by the formula $[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i](x, y)$, where x and y are (not necessarily distinct) variable symbols and $i \in [1, e]$.

Remark 14. *Let φ be a $\Lambda[\Sigma]$-formula and let $\mathfrak{A}$ be a Σ-structure. Suppose that $\mathfrak{A} \vDash \varphi$ and that the symbols $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$ are interpreted in $\mathfrak{A}$ as equivalence relations on A in local agreement. Then there is a Σ'-enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \operatorname{ltr} \varphi$ and that the symbols $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e$ are interpreted in $\mathfrak{A}'$ as equivalence relations on A in refinement.*

Proof. Since the binary symbols $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$ are interpreted as equivalence relations on A in local agreement in $\mathfrak{A}$, we may define the levels $L_1, L_2, \ldots, L_e \subseteq A \times A$ and the characteristic E-permutation mapping $\nu = [\mathrm{E}\textbf{:}\mathsf{chperm}]^{\mathfrak{A}} : A \to \mathbb{S}_e$. Consider an enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ where $\boldsymbol{l}_i^{\mathfrak{A}'} = L_i$. By Remark 3, L_i are equivalences on A in refinement. We interpret the unary symbols from the permutation setup P so that $[\mathrm{P}\textbf{:}\mathrm{data}]^{\mathfrak{A}'} a = \nu(a)$ for all $a \in A$. By Remark 10, $\mathfrak{A}' \vDash [\mathrm{L}\textbf{:}\mathsf{fixperm}]$. By Remark 12, followed by Lemma 2, for every $i \in [1, e]$ and $a \in A$ we have:

$$[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i]^{\mathfrak{A}'}[a] = \boldsymbol{l}^{\mathfrak{A}'}_{\nu(a)^{-1}(i)}[a] = \boldsymbol{e}^{\mathfrak{A}'}_{\nu(a)(\nu(a)^{-1}(i))}[a] = \boldsymbol{e}_i^{\mathfrak{A}'}[a].$$

By Remark 13, the interpretations $[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i]^{\mathfrak{A}'}$ are equivalence relations. Since the interpretation of the formula $[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i]$ has the same classes as the interpretation of the symbol $\boldsymbol{e}_i$, we have $\mathfrak{A}' \vDash \forall \boldsymbol{x} \forall \boldsymbol{y}\, (\boldsymbol{e}_i(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\mathrm{L}\textbf{:}\mathsf{el}\text{-}i](\boldsymbol{x}, \boldsymbol{y}))$. Since $\mathfrak{A}' \vDash \varphi$ we have $\mathfrak{A}' \vDash \operatorname{ltr} \varphi$. $\square$

Remark 15. *Let φ be a $\Lambda[\Sigma]$-formula and let $\mathfrak{A}$ be an L'-structure. Suppose that $\mathfrak{A} \vDash \operatorname{ltr} \varphi$ and that the symbols $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_e$ are interpreted as equivalence relations on A in refinement in $\mathfrak{A}$. Then there is a Σ'-enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \varphi$ and that the binary symbols $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e$ are interpreted as equivalence relations on A in global agreement in $\mathfrak{A}'$.*

Proof. Consider an enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ to a Σ'-structure where $\boldsymbol{e}_i^{\mathfrak{A}'} = [\mathrm{L}\textbf{:}\mathsf{el}\text{-}i]^{\mathfrak{A}}$. By Remark 13, $\boldsymbol{e}_i^{\mathfrak{A}'}$ are equivalence relations on A in local agreement. For every $i \in [1, e]$ we have $\mathfrak{A}' \vDash \forall \boldsymbol{x} \forall \boldsymbol{y}\, (\boldsymbol{e}_i(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\mathrm{L}\textbf{:}\mathsf{el}\text{-}i](\boldsymbol{x}, \boldsymbol{y}))$ by definition. Since $\mathfrak{A}' \vDash \operatorname{ltr} \varphi$ we have $\mathfrak{A}' \vDash \varphi$. $\square$

The last two remarks show that a $\Lambda e\mathrm{E}_{\mathsf{local}}$-formula φ has essentially the same models as the $\Lambda e\mathrm{E}_{\mathsf{refine}}$-formula $\operatorname{ltr}\varphi$, so we have shown:

Proposition 3. *The logic $\Lambda e\mathrm{E}_{\mathsf{local}}$ has the finite model property iff the logic $\Lambda e\mathrm{E}_{\mathsf{refine}}$ has the finite model property. The corresponding satisfiability problems are polynomial-time equivalent:* $(\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{local}} =_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda e\mathrm{E}_{\mathsf{refine}}$.

Since the relative size of $\operatorname{ltr}\varphi$ with respect to φ grows polynomially as e grows, we have shown:

Proposition 4. *The logic $\Lambda\mathrm{E}_{\mathsf{local}}$ has the finite model property iff the logic $\Lambda\mathrm{E}_{\mathsf{refine}}$ has the finite model property. The corresponding satisfiability problems are polynomial-time equivalent:* $(\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{local}} =_{\mathrm{m}}^{\mathrm{PTIME}} (\mathrm{FIN})\mathrm{SAT}\Lambda\mathrm{E}_{\mathsf{refine}}$.

The reduction is two-variable first-order and uses additional (et) unary predicate symbols for the permutation setup P, so it is also valid for the two-variable fragments $\Lambda^2_0 e\mathrm{E}_{\mathsf{a}}$, $\Lambda^2_1 e\mathrm{E}_{\mathsf{a}}$ and $\Lambda^2_1\mathrm{E}_{\mathsf{a}}$ for $\mathsf{a} \in \{\mathsf{local}, \mathsf{refine}\}$ respectively.

4.3 Granularity

In this section we demonstrate how to replace the finest equivalence from a sequence of equivalences in refinement with a counter setup. This works if the structures are granular, that is, if the finest equivalence doesn't have many classes within a single bigger equivalence class.

Definition 38. *Let $\langle D, E\rangle$ be a sequence of two equivalence relations on A in refinement. Let $g \in \mathbb{N}^+$. The sequence is g-*granular *if every E-equivalence class includes at most g D-equivalence classes.*

Definition 39. *Let $g \in \mathbb{N}^+$ and let $\langle D, E\rangle$ be g-granular. The function $c : A \to [1, g]$ is a g-*granular coloring *for the sequence, if two E-equivalent elements have the same color iff they are D-equivalent. That is, for every $(a, b) \in E$ we have $c(a) = c(b)$ iff $(a, b) \in D$.*

Remark 16. *Let $g \in \mathbb{N}^+$ and let $\langle D, E\rangle \subseteq A \times A$ be g-granular. Then there is a g-granular coloring for the sequence.*

Proof. Let X be an E-class. Since $D \subseteq E$ is g-granular, the set $S = \{D[a] \mid a \in X\}$ has cardinality at most g. Let $\imath : S \hookrightarrow [1, g]$ be any injective function. Define the color c on X as $c(a) = \imath(D[a])$. □

Remark 17. *Let $E \subseteq A \times A$ be an equivalence relation on A, $g \in \mathbb{N}^+$ and $c : A \to [1, g]$. Then there is an equivalence relation $D \subseteq E$ on A such that $\langle D, E\rangle$ is g-granular, having c as a g-granular coloring.*

Proof. Take $D = \{(a, b) \in E \mid c(a) = c(b)\}$. □

Definition 40. *Let $g \in \mathbb{N}^+$ and let $t = \|g\|$ be the bitsize of g. A g-*color setup *$\mathrm{G} = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_t\rangle$ is just a t-bit counter setup.*

Let $\Lambda \in \{\mathcal{L}, \mathcal{C}\}$ be a ground logic, $g \in \mathbb{N}^+$ and $\mathrm{G} = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_t \rangle$ be a g-color setup. Let Σ be a predicate signature containing the binary symbols $\boldsymbol{d}$ and $\boldsymbol{e}$ and not containing any symbols from G. Let $\Sigma' = \Sigma + \mathrm{G}$ and $\Gamma = \Sigma' - \{\boldsymbol{d}\}$.

Definition 41. *Define the quantifier-free $\mathcal{L}^2[\Gamma]$-formula $[\Gamma\mathsf{:d}](\boldsymbol{x}, \boldsymbol{y})$ by:*

$$[\Gamma\mathsf{:d}](\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \wedge [\mathrm{G}\mathsf{:eq}](\boldsymbol{x}, \boldsymbol{y}).$$

Definition 42. *Define the syntactic operation* grtr $: \Lambda[\Sigma] \to \Lambda[\Gamma]$ *by:*

$$\operatorname{grtr} \varphi = \varphi' \wedge [\mathrm{G}\mathsf{:betw\text{-}1\text{-}}g],$$

where φ' is obtained from the formula φ by replacing all subformulas of the form $\boldsymbol{d}(x, y)$ by $[\Gamma\mathsf{:d}](x, y)$, where x and y are (not necessarily distinct) variable symbols.

Lemma 3. *Let $\mathfrak{A}$ be a Σ-structure and suppose that the sequence of symbols $\langle \boldsymbol{d}, \boldsymbol{e} \rangle$ is interpreted in $\mathfrak{A}$ as a g-granular sequence $\langle D, E \rangle$. Suppose that $\mathfrak{A} \vDash \varphi$. Then there is a Σ'-enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \operatorname{grtr} \varphi$.*

Proof. By Remark 16, there exists a g-granular coloring $c : A \to [1, g]$. We interpret the unary symbols in G so that $[\mathrm{G}\mathsf{:data}]^{\mathfrak{A}} = c$. Since $\mathfrak{A}'$ is an enrichment of $\mathfrak{A}$, we have $\mathfrak{A}' \vDash \varphi$. Let $a, b \in A$. Then $\mathfrak{A}' \vDash [\Gamma\mathsf{:d}](a, b)$ is equivalent to:

$$\mathfrak{A}' \vDash \boldsymbol{e}(a, b) \text{ and } \mathfrak{A}' \vDash [\mathrm{G}\mathsf{:eq}](a, b),$$

which is equivalent to:

$$(a, b) \in E \text{ and } [\mathrm{G}\mathsf{:data}]^{\mathfrak{A}'} a = [\mathrm{G}\mathsf{:data}]^{\mathfrak{A}'} b,$$

which, since $[\mathrm{G}\mathsf{:data}]^{\mathfrak{A}'} = c$ is a g-granular coloring, is equivalent to:

$$(a, b) \in D.$$

Hence $\mathfrak{A}' \vDash \forall \boldsymbol{x} \forall \boldsymbol{y} \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\Gamma\mathsf{:d}](\boldsymbol{x}, \boldsymbol{y})$ and since $\mathfrak{A}' \vDash \varphi$, we have $\mathfrak{A}' \vDash \operatorname{grtr} \varphi$. □

Lemma 4. *Let $\mathfrak{A}$ be a Γ-structure and suppose that the binary symbol $\boldsymbol{e}$ is interpreted in $\mathfrak{A}$ as an equivalence relation on A. Suppose that $\mathfrak{A} \vDash \operatorname{grtr} \varphi$. Then there is a Σ'-structure $\mathfrak{A}'$ enriching $\mathfrak{A}$ such that $\mathfrak{A}' \vDash \varphi$ and the sequence of binary symbols $\langle \boldsymbol{d}, \boldsymbol{e} \rangle$ is interpreted in $\mathfrak{A}'$ as a g-granular sequence $\langle D, E \rangle$.*

Proof. Since $\mathfrak{A} \vDash [\mathrm{G}\mathsf{:betw\text{-}1\text{-}}g]$, we have $[\mathrm{G}\mathsf{:data}]^{\mathfrak{A}} a \in [1, g]$ for all $a \in A$. Define $c : A \to [1, g]$ by $c(a) = [\mathrm{C}\mathsf{:data}]^{\mathfrak{A}} a$. By Remark 16, we can find $D \subseteq E$ such that the sequence $\langle D, E \rangle$ is g-granular, having c as a g-granular coloring. Consider the Σ'-structure $\mathfrak{A}'$, where $\boldsymbol{d}^{\mathfrak{A}'} = D$. Since $\mathfrak{A}'$ is an enrichment of $\mathfrak{A} \vDash \operatorname{grtr} \varphi$, we have $\mathfrak{A}' \vDash \varphi$. Let $a, b \in A$. Then $\mathfrak{A}' \vDash [\Gamma\mathsf{:d}](a, b)$ is equivalent to:

$$\mathfrak{A}' \vDash \boldsymbol{e}(a, b) \text{ and } \mathfrak{A}' \vDash [\mathrm{G}\mathsf{:eq}](a, b),$$

which is equivalent to:

$$(a, b) \in E \text{ and } c(a) = c(b),$$

which, since c is a g-granular coloring, is equivalent to:

$$(a, b) \in D.$$

Hence $\mathfrak{A}' \models \forall \boldsymbol{x} \forall \boldsymbol{y} \, (\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \leftrightarrow [\Gamma\mathbf{:}\mathsf{d}](\boldsymbol{x}, \boldsymbol{y}))$ and since $\mathfrak{A}' \models \operatorname{grtr} \varphi$, we have $\mathfrak{A}' \models \varphi$. $\square$

5 Monadic logics

In this chapter we investigate questions about (finite) satisfiability of first-order sentences featuring unary predicate symbols and builtin equivalence symbols in refinement. Our strategy is to extract small substructures of structures and analyse them using Ehrenfeucht-Fraïssé games. It is known that:

- The monadic first-order logic $\mathcal{L}_1$ has the finite model property and its (finite) satisfiability problem is NExpTime-complete [8]
- The first-order logic of a single equivalence relation $\mathcal{L}_0 1\mathrm{E}$ has the finite model property and its (finite) satisfiability problem is PSpace-complete [6]
- The first-order logic of two equivalence relations $\mathcal{L}_0 2\mathrm{E}$ lacks the finite model property and both the satisfiability and finite satisfiability problems are undecidable [9].

Let $\mathrm{U}(u) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_u \rangle$ be an unary predicate signature consisting of the unary predicate symbols $\boldsymbol{u}_i$. Let $\mathrm{E}(e) = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_e \rangle$ be a binary predicate signature consisting of the builtin equivalence symbols $\boldsymbol{e}_j$ in refinement. Let $\Sigma(u, e) = \mathrm{U}(u) + \mathrm{E}(e)$, so $\Sigma(u, e)$ is a generic predicate signature for the monadic first-order logic $\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$.

5.1 Cells

Let $u, e \in \mathbb{N}$, $e \geq 1$ and $\Sigma = \Sigma(u, e) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_u, \boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_e \rangle$ be a predicate signature. Abbreviate the finest equivalence symbol $\boldsymbol{d} = \boldsymbol{e}_1$.

Definition 43. *Define the quantifier-free $\mathcal{L}^2[\Sigma]$-formula $[\Sigma{:}\mathsf{cell}](\boldsymbol{x}, \boldsymbol{y})$ by:*

$$[\Sigma{:}\mathsf{cell}](\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \wedge \bigwedge_{1 \leq i \leq u} (\boldsymbol{u}_i(\boldsymbol{x}) \leftrightarrow \boldsymbol{u}_i(\boldsymbol{y})).$$

If $\mathfrak{A}$ is a Σ-structure and $D = \boldsymbol{d}^{\mathfrak{A}}$, then the interpretation $C = [\Sigma{:}\mathsf{cell}]^{\mathfrak{A}} \subseteq A \times A$ is an equivalence relation on A that refines D. The cells *of $\mathfrak{A}$ are the equivalence classes of C. That is, a cell is a maximal set of D-equivalent elements satisfying the same $\boldsymbol{u}$-predicates.*

Remark 18. *Let $\mathfrak{A}$ be a Σ-structure, $r \in \mathbb{N}$, $\bar{a} = a_1 a_2 \dots a_r \in A^r$, $\bar{b} = b_1 b_2 \dots b_r \in A^r$ and a_i and b_i are in the same $\mathfrak{A}$-cell for all $i \in [1, r]$. Suppose that $a_i = a_j$ iff $b_i = b_j$ for all $i, j \in [1, r]$. Then $\bar{a} \mapsto \bar{b}$ is a partial isomorphism.*

Proof. Direct consequence of the fact that the cell equivalence relation refines the finest equivalence relation D and that the elements in the same cell satisfy the same $\boldsymbol{u}$-predicates. The equality condition ensures that the mapping is a bijection. $\square$

Lemma 5. *Let $\mathfrak{A}$ be a Σ-structure and $r \in \mathbb{N}^+$. There is $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and every $\mathfrak{B}$-cell has cardinality at most r.*

Proof. Let $C \subseteq A \times A$ be the $\mathfrak{A}$-cell equivalence relation. Execute the following process: for every $\mathfrak{A}$-cell, if it has cardinality less than r, select all elements from that cell; otherwise select r distinct elements from that cell. Let $B \subseteq A$ be the set of selected elements and let $\mathfrak{B} = \mathfrak{A} \upharpoonright B$. By construction, every $\mathfrak{B}$-cell has cardinality at most r. We claim that $\mathfrak{A} \equiv_r \mathfrak{B}$. Let $h = C \cap (A \times B)$ relates elements from A with elements from B in the same cell. Note that for all $a \in A$:

$$|h[a]| = \min(|C[a]|, r). \tag{5.1}$$

Consider the set $\mathfrak{I}$ of partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ that have cardinality at most r and that are included in h. This set is nonempty since it contains the empty partial isomorphism. We claim that the sequence $\mathfrak{I}_0 = \mathfrak{I}_1 = \cdots = \mathfrak{I}_r = \mathfrak{I}$ satisfies the back-and-forth conditions of Theorem 1. Let $i \in [1, r]$ and let

$$\mathfrak{p} = \bar{a} \mapsto \bar{b} = a_1 a_2 \ldots a_r \mapsto b_1 b_2 \ldots b_r \in \mathfrak{I}$$

be any partial isomorphism. Without loss of generality, suppose $i = 1$.

1. For the forth condition, let $a \in A$. We have to find some $b \in B$ such that $\bar{a}_i^a \mapsto \bar{b}_i^b \in \mathfrak{I}$. If $a = a_k$ for some $k \in [2, r]$, then $b = b_k$ is appropriate.

   Suppose that $a \neq a_k$ for all $k \in [2, r]$. Let $S \subseteq C[a]$ be the set of $\bar{a}$-elements in the same $\mathfrak{A}$-cell as a:
   $$S = \{a_k \in C[a] \mid k \in [2, r]\}.$$
   Note that $|S| \leq r - 1$ and $|C[a]| \geq |S| + 1$. By eq. (5.1), $|h[a]| \geq |S| + 1$. Hence there is an element $b \in h[a]$ that is distinct from b_k for all $k \in [2, r]$.

2. For the back condition, let $b \in B$. We have to find some $a \in A$ such that $\bar{a}_i^a \mapsto \bar{b}_i^b \in \mathfrak{I}$. If $b = b_k$ for some $k \in [2, r]$, then $a = a_k$ is appropriate.

   Suppose that $b \neq b_k$ for all $k \in [2, r]$. Since $b \in h[b]$, $a = b$ is appropriate.

By Theorem 1, $\mathfrak{A} \equiv_r \mathfrak{B}$. □

5.2 Organs

Let $u, e \in \mathbb{N}$, $e \geq 2$ and $\Sigma = \Sigma(u, e) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ be a predicate signature. Abbreviate the finest two equivalence symbols $\boldsymbol{d} = \boldsymbol{e}_1$ and $\boldsymbol{e} = \boldsymbol{e}_2$.

Definition 44. *Let $\mathfrak{A}$ be a Σ-structure and let $D = \boldsymbol{d}^{\mathfrak{A}}$ and $E = \boldsymbol{e}^{\mathfrak{A}}$. Recall that the set of D-classes is $\mathscr{E}D$. Two D-classes $X, Y \in \mathscr{E}D$ are* organ-equivalent *if $X \times Y \subseteq E$ and the induced substructures $(\mathfrak{A} \upharpoonright X)$ and $(\mathfrak{A} \upharpoonright Y)$ are isomorphic. The organ-equivalence relation is $\mathcal{O} \subseteq \mathscr{E}D \times \mathscr{E}D$. Since D refines E, organ-equivalence is an equivalence* *relation on $\mathscr{E}D$. An* organ *is an organ-equivalence-class. That is, an organ is a maximal set of isomorphic D-classes, included in the same E-class.*

For any two organ-equivalent D-classes $(X, Y) \in \mathcal{O}$, fix an isomorphism

$$\mathfrak{h}_{XY} : (\mathfrak{A} \upharpoonright X) \leftrightarrow (\mathfrak{A} \upharpoonright Y)$$

consistently, so that $\mathfrak{h}_{XX} = \mathrm{id}_X$, $\mathfrak{h}_{YX} = \mathfrak{h}_{XY}^{-1}$ and if $(Y, Z) \in \mathcal{O}$ then $\mathfrak{h}_{XZ} = \mathfrak{h}_{YZ} \circ \mathfrak{h}_{XY}$. Two elements $a, b \in A$ are sub-organ-equivalent *if $(D[a], D[b]) \in \mathcal{O}$ and $\mathfrak{h}_{D[a]D[b]}(a) = b$. Since the isomorphisms $\mathfrak{h}_{XY}$ are chosen consistently, sub-organ-equivalence $O \subseteq A \times A$ is an equivalence relation on A that refines E.*

Remark 19. *Let $\mathfrak{A}$ be a Σ-structure, $r \in \mathbb{N}$, $\bar{a} = a_1 a_2 \dots a_r \in A^r$, $\bar{b} = b_1 b_2 \dots b_r \in A^r$, a_i and b_i are sub-organ-equivalent for all $i \in [1, r]$. Suppose that $\mathfrak{A} \vDash \boldsymbol{d}(a_i, a_j)$ iff $\mathfrak{A} \vDash \boldsymbol{d}(b_i, b_j)$ for all $i, j \in [1, r]$. Then $\bar{a} \mapsto \bar{b}$ is a partial isomorphism.*

Proof. The condition about the finest equivalence symbol $\boldsymbol{d}$ ensures that the interpretation of $\boldsymbol{d}$ is preserved. Since sub-organ-equivalence relates isomorphic elements, the interpretation of the unary symbols and the formal equality is preserved. Since the sub-organ-equivalence $O \subseteq A \times A$ refines the second finest equivalence relation E, the interpretation of all remaining equivalence symbols $\boldsymbol{e}_j$ is preserved. □

Lemma 6. *Let $\mathfrak{A}$ be a Σ-structure and $r \in \mathbb{N}^+$. There is $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and every $\mathfrak{B}$-organ has cardinality at most r.*

Proof. Let $D = \boldsymbol{d}^{\mathfrak{A}}$, $E = \boldsymbol{e}^{\mathfrak{A}}$ and let $\mathcal{A} = \mathscr{E}D$ be the set of D-classes. Let $\mathcal{O} \subseteq \mathcal{A} \times \mathcal{A}$ be the $\mathfrak{A}$-organ-equivalence relation on $\mathcal{A}$. Execute the folloing process: for every $\mathfrak{A}$-organ, if it has cardinality at most r, select all D-classes from that organ; otherwise select r distinct D-classes from that organ (note that these will be isomorphic). Let $\mathcal{B} \subseteq \mathcal{A}$ be the set of selected D-classes. Let $B = \cup\mathcal{B} \subseteq A$ be the set of elements in the selected classes and let $\mathfrak{B} = (\mathfrak{A} \upharpoonright B)$. By construction, every $\mathfrak{B}$-organ has cardinality at most r. We claim that $\mathfrak{A} \equiv_r \mathfrak{B}$. Let $\mathcal{H} = \mathcal{O} \cap (\mathcal{A} \times \mathcal{B})$ relates the D-classes with the isomorphic D-classes from $\mathcal{B}$ in the same organ. Let h relates the elements of A with their isomorphic elements from B. Note that for all elements $a \in A$:

$$|h[a]| = \min(|\mathcal{O}[D[a]]|, r). \tag{5.2}$$

Consider the set $\mathfrak{I}$ of partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ that have cardinality at most r and that are included in h. This set is nonempty since it contains the empty partial isomorphism. We claim that the sequence $\mathfrak{I}_0 = \mathfrak{I}_1 = \dots = \mathfrak{I}_r = \mathfrak{I}$ satisfies the back-and-forth conditions of Theorem 1. Let $i \in [1, r]$ and let

$$\mathfrak{p} = \bar{a} \mapsto \bar{b} = a_1 a_2 \dots a_r \mapsto b_1 b_2 \dots b_r \in \mathfrak{I}$$

be any partial isomorphism. Without loss of generality, suppose $i = 1$.

1. For the forth condition, let $a \in A$. We have to find some $b \in B$ such that $\bar{a}_i^a \mapsto \bar{b}_i^b \in \mathfrak{I}$. If $a \in D[a_k]$ for some $k \in [2, r]$, then $b = \mathfrak{h}_{D[a_k]D[b_k]}(a)$ is appropriate.

   Suppose $a \notin D[a_k]$ for all $k \in [2, r]$. Let $\mathcal{S} \subseteq \mathcal{O}[D[a]]$ be the set of D-classes of $\bar{a}$-elements in the same $\mathfrak{A}$-organ as $D[a]$:

$$\mathcal{S} = \{D[a_k] \in \mathcal{O}[D[a]] \mid k \in [2, r]\}\,.$$

   Note that $|\mathcal{S}| \le r - 1$ and $|\mathcal{O}[D[a]]| \ge |\mathcal{S}| + 1$. By eq. (5.2), $|h[a]| \ge |\mathcal{S}| + 1$. Hence there is some $b \in h[a]$ such that $b \notin D[b_k]$ for all $k \in [2, r]$. This b is appropriate.

2. For the back condition, let $b \in B$. We have to find some $a \in A$ such that $\bar{a}_i^a \mapsto \bar{b}_i^b \in \mathfrak{I}$. If $b \in D[b_k]$ for some $k \in [2, r]$, then $a = \mathfrak{h}_{D[b_k]D[a_k]}(b)$ is appropriate.

   Suppose that $b \notin D[b_k]$ for all $k \in [2, r]$. Since $b \in h[b]$, $a = b$ is appropriate.

By Theorem 1, $\mathfrak{A} \equiv_r \mathfrak{B}$. □

5.3 Satisfiability

In this section we will employ the results on cells and organs to bound the size of a small substructure of a general structure.

Remark 20. *Let $u, e \in \mathbb{N}$, $e \ge 2$ and consider the predicate signature $\Sigma = \Sigma(u, e) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$. Abbreviate $\boldsymbol{d} = \boldsymbol{e}_1$ and $\boldsymbol{e} = \boldsymbol{e}_2$. Let $r \in \mathbb{N}^+$. There is $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and $\langle \boldsymbol{d}^{\mathfrak{B}}, \boldsymbol{e}^{\mathfrak{B}} \rangle$ is g-granular for $g = g(u, r) = r.((r + 1)^{2^u} - 1)$. Furthermore, this $\mathfrak{B}$ has the property that every $\mathfrak{B}$-cell has cardinality at most r.*

Proof. By Lemma 5, there is $\mathfrak{B}' \subseteq \mathfrak{A}$ such that $\mathfrak{B}' \equiv_r \mathfrak{A}$ and every $\mathfrak{B}'$-cell has cardinality at most r. By Lemma 6, there is $\mathfrak{B} \subseteq \mathfrak{B}'$ such that $\mathfrak{B} \equiv_r \mathfrak{B}'$ and the $\mathfrak{B}$-organs have cardinality at most r. Let $D = \boldsymbol{d}^{\mathfrak{B}}$ and $E = \boldsymbol{e}^{\mathfrak{B}}$. Since every D-class includes at most 2^u cells and is nonempty and every cell has cardinality at most r, there are at most $\big((r + 1)^{2^u} - 1\big)$ nonisomorphic D-classes in $\mathfrak{B}$. Since every E-class includes at most r isomorphic D-classes, we get that $\langle D, E \rangle$ is g-granular. □

Corollary 1. *Let $u, e \in \mathbb{N}$, $e \ge 2$ and consider $\Sigma = \Sigma(u, e)$. Let φ be a $\mathcal{L}[\Sigma]$-sentence having quantifier rank r. By Lemma 3 and Lemma 4 about granularity, the formula φ is essentially equisatisfiable with the formula $\operatorname{grtr} \varphi$, which is a $\Sigma(u + \|g(u, r)\|, e - 1)$-sentence. Note that $\|g(u, r)\|$ is exponentially bounded by the length $\|\varphi\|$ of the formula. So we have a reduction:*

$$\text{(FIN)SAT}\mathcal{L}_1 e\text{E}_{\mathsf{refine}} \le_{\text{m}}^{\text{ExpTime}} \text{(FIN)SAT}\mathcal{L}_1 (e-1)\text{E}_{\mathsf{refine}}.$$

If u is a constant independent of φ, then $\|g(u, r)\|$ is polynomially bounded by $\|\varphi\|$. So we have a reduction:

$$\text{(FIN)SAT}\mathcal{L}_0 e\text{E}_{\mathsf{refine}} \le_{\text{m}}^{\text{PTime}} \text{(FIN)SAT}\mathcal{L}_1 (e-1)\text{E}_{\mathsf{refine}}.$$

Remark 21. *Let $u \in \mathbb{N}$ and consider $\Sigma = \Sigma(u, 1) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{d} \rangle$. Let $r \in \mathbb{N}^+$. There is $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{A} \equiv_r \mathfrak{B}$ and $|B| \leq g.r.2^u$ for $g = g(u, r) = r.((r+1)^{2^u} - 1)$.*

Proof. Let $\Sigma' = \Sigma + \langle \boldsymbol{e} \rangle$ be an enrichment of Σ with the builtin equivalence symbols $\boldsymbol{e}$. Consider an enrichment $\mathfrak{A}'$ of $\mathfrak{A}$ to a Σ'-structure, where $\boldsymbol{e}^{\mathfrak{A}'} = A \times A$ is interpreted as the full relation on A. Then $\langle \boldsymbol{d}^{\mathfrak{A}'}, \boldsymbol{e}^{\mathfrak{A}'} \rangle$ is a sequence of equivalence relations on A in refinement. By Remark 20, there is $\mathfrak{B}' \subseteq \mathfrak{A}'$ such that $\mathfrak{B}' \equiv_r \mathfrak{A}'$ and $\langle \boldsymbol{d}^{\mathfrak{B}'}, \boldsymbol{e}^{\mathfrak{B}'} \rangle$ is g-granular. Consider the reduct $\mathfrak{B}$ of $\mathfrak{B}'$ to a Σ-structure. Let $D = \boldsymbol{d}^{\mathfrak{B}}$ and $E = \boldsymbol{e}^{\mathfrak{B}}$. Since every $\mathfrak{B}$-cell has cardinality at most r and every D-class includes at most 2^u cells, we have that every D-class has cardinality at most $r.2^u$. Since $\boldsymbol{e}$ was interpreted in $\mathfrak{A}$ as the full relation, it is also interpreted in $\mathfrak{B}$ as the full relation, so there is a single E-class—the whole domain B. Since the sequence $\langle D, E \rangle$ is g-granular, there are at most g D-classes, so $|B| \leq g.r.2^u$. ☐

Corollary 2. *The logic $\mathcal{L}_1$1E has the finite model property and its (finite) satisfiability problem is in* N2ExpTime.

Combinig Corollary 2 with Corollary 1, we get by induction on e:

Proposition 5. *For $e \in \mathbb{N}^+$, the logic $\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$ has the finite model property and its (finite) satisfiability problem is in* N$(e+1)$ExpTime.
By Proposition 1 and Proposition 3, the same holds for $\mathcal{L}_1 e\mathrm{E}_{\mathsf{global}}$ and $\mathcal{L}_1 e\mathrm{E}_{\mathsf{local}}$.

Proposition 6. *The logic $\mathcal{L}_1\mathrm{E}_{\mathsf{refine}}$ has the finite model property and its (finite) satisfiability problem is in the forth level of the Grzegorczyk hierarchy $\mathcal{E}^4$.*
By Proposition 2 and Proposition 4, the same holds for $\mathcal{L}_1\mathrm{E}_{\mathsf{global}}$ and $\mathcal{L}_1\mathrm{E}_{\mathsf{local}}$.

Proposition 7. *For $e \geq 2$, the logic $\mathcal{L}_0 e\mathrm{E}_{\mathsf{refine}}$ has the finite model property and its (finite) satisfiability problem is in* NeExpTime.
By Proposition 1 and Proposition 3, the same holds for $\mathcal{L}_0 e\mathrm{E}_{\mathsf{global}}$ and $\mathcal{L}_0 e\mathrm{E}_{\mathsf{local}}$.

5.4 Hardness with a single equivalence

In this section we show that the (finite) satisfiability of monadic first-order logic with a single equivalence symbol $\mathcal{L}_1$1E is N2ExpTime-hard by reducing the doubly exponential tiling problem to such satisfiability. Our strategy is to employ a counter setup of u unary predicate symbols to encode the exponentially many positions of a binary encoding of a doubly exponentially bounded quantity, encoding the coordinaes of a cell of the doubly exponential tiling square.

Consider the counter setup $\mathrm{C}(u) = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u \rangle$ for $u \equiv \mathbb{N}^+$. Recall that the intention of a counter setup is to encode an arbitrary exponentially bounded value at every element of a structure. Let $\mathrm{D}(u) = \mathrm{C}(u) + \langle \boldsymbol{d} \rangle$ be a predicate signature enriching $\mathrm{C}(u)$ with the builtin equivalence symbol $\boldsymbol{d}$. We will define a system where every $\boldsymbol{d}$-equivalence class includes exponentially many cells. These cells will correspond to the exponentially many positions of the binary encoding of a doubly exponential value for the $\boldsymbol{d}$-class. The bit values at each cell position will be encoded by the cardinality of that cell: bit value 0 if the cardinality of the cell is 1 and bit value 1 if the cardinality is greater than 1. This will allow us to encode a doubly exponential value at each $\boldsymbol{d}$-class. Call the data $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a$, encoded by the counter setup at a the *position* of a.

Let $\mathfrak{A}$ be a $\mathrm{D} = \mathrm{D}(u)$-structure.

Definition 45. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{x}, \boldsymbol{y}) = [\mathrm{C}\mathbf{:}\mathsf{eq}](\boldsymbol{x}, \boldsymbol{y}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}eq}](a, b)$ iff a and b are at the same positions (in possibly distinct $\boldsymbol{d}$-classes): $[\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}a = [\mathrm{C}\mathbf{:}\mathrm{data}]^{\mathfrak{A}}b$.

Definition 46. *Define the quantifier-rank-*1 $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{bit\text{-}0}](\boldsymbol{x})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{bit\text{-}0}](\boldsymbol{x}) = \forall \boldsymbol{y}\, (\boldsymbol{d}(\boldsymbol{y}, \boldsymbol{x}) \wedge [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{y}, \boldsymbol{x}) \rightarrow \boldsymbol{y} = \boldsymbol{x}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{bit\text{-}0}](a)$ iff the cell of a has cardinality 1.

Definition 47. *Define the quantifier-rank-*1 $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{bit\text{-}1}](\boldsymbol{x})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{bit\text{-}1}](\boldsymbol{x}) = \exists \boldsymbol{y}\, (\boldsymbol{d}(\boldsymbol{y}, \boldsymbol{x}) \wedge [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{y}, \boldsymbol{x}) \wedge \boldsymbol{y} \neq \boldsymbol{x}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{bit\text{-}1}](a)$ iff the cell of a has cardinality greater than 1.

Definition 48. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}zero}](\boldsymbol{x})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}zero}](\boldsymbol{x}) = \bigwedge_{1 \leq i \leq u} \neg \boldsymbol{u}_i(\boldsymbol{x}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}zero}](a)$ iff the position of a is 0.

Definition 49. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}largest}](\boldsymbol{x})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}largest}](\boldsymbol{x}) = \bigwedge_{1 \leq i \leq u} \boldsymbol{u}_i(\boldsymbol{x}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}largest}](a)$ iff the position of a is the largest u-bit number N_u.

Definition 50. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}less}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}less}](\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \wedge [\mathrm{C}\mathbf{:}\mathsf{less}](\boldsymbol{x}, \boldsymbol{y}).$$

Then $\mathfrak{A} \models [\mathrm{D}\mathbf{:}\mathsf{pos\text{-}less}](a, b)$ iff a and b are in the same $\boldsymbol{d}$-class and the position of a is less than the position of b.

Definition 51. *Define the quantifier-free* $\mathcal{L}^2[\mathrm{D}]$*-formula* $[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}succ}](\boldsymbol{x}, \boldsymbol{y})$ *by:*

$$[\mathrm{D}\mathbf{:}\mathsf{pos\text{-}succ}](\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \wedge [\mathrm{C}\mathbf{:}\mathsf{succ}](\boldsymbol{x}, \boldsymbol{y}).$$

Then $\mathfrak{A} \models [\text{D:}\mathsf{pos\text{-}succ}](a, b)$ iff a and b are in the same $\boldsymbol{d}$-class and the position of b is the successor of the position of a.

Definition 52. *Define the closed $\mathcal{L}^2[\mathrm{D}]$-sentence* $[\text{D:}\mathsf{pos\text{-}full}]$ *by:*

$$[\text{D:}\mathsf{pos\text{-}full}] = \forall \boldsymbol{x} \exists \boldsymbol{y} \Big(\boldsymbol{d}(\boldsymbol{y}, \boldsymbol{x}) \wedge [\text{D:}\mathsf{pos\text{-}zero}](\boldsymbol{y}) \Big) \wedge$$
$$\forall \boldsymbol{x} \Big(\neg [\text{D:}\mathsf{pos\text{-}largest}](\boldsymbol{x}) \to \exists \boldsymbol{y} [\text{D:}\mathsf{pos\text{-}succ}](\boldsymbol{x}, \boldsymbol{y}) \Big).$$

The first part of this formula asserts that every $\boldsymbol{d}$-class has an element at position 0. The second part asserts that if a is an element at position p, that is not the largest possible, there exists an element b in the same $\boldsymbol{d}$-class at position $p+1$. Therefore in any model of $[\text{D:}\mathsf{pos\text{-}full}]$, every $\boldsymbol{d}$-class has 2^u cells. For example, in particular, every $\boldsymbol{d}$-class has cardinality at least 2^u. For the rest of the section, suppose that $\mathfrak{A} \models [\text{D:}\mathsf{pos\text{-}full}]$.

Definition 53. *For every u-bit number $p \in \mathbb{B}_u$, define the $\mathcal{L}^2[\mathrm{D}]$-formula* $[\text{D:}\mathsf{pos}\text{-}p](\boldsymbol{x})$ *recursively by:*

$$[\text{D:}\mathsf{pos}\text{-}0](\boldsymbol{x}) = [\text{D:}\mathsf{pos\text{-}zero}](\boldsymbol{x})$$

and for $p \in [0, N_u - 1]$:

$$[\text{D:}\mathsf{pos}\text{-}(p+1)](\boldsymbol{x}) = \exists \boldsymbol{y} \Big([\text{D:}\mathsf{pos}\text{-}p](\boldsymbol{y}) \wedge [\text{D:}\mathsf{pos\text{-}succ}](\boldsymbol{y}, \boldsymbol{x}) \Big).$$

In this case, for the formula to be a two-variable formula, the formula $[\text{D:}\mathsf{pos}\text{-}p](\boldsymbol{y})$ is obtained from $[\text{D:}\mathsf{pos}\text{-}p](\boldsymbol{x})$ by swapping all occurences (not only the unbounded ones) of the variables $\boldsymbol{x}$ and $\boldsymbol{y}$[1]. Note that the length of the formula $[\text{D:}\mathsf{pos}\text{-}p](\boldsymbol{x})$ grows linearly as p grows.

Then $\mathfrak{A} \models [\text{D:}\mathsf{pos}\text{-}p](a)$ iff p is the position of a.

Definition 54. *Let $\mathfrak{A}$ be a D-structure. Let $D = \boldsymbol{d}^{\mathfrak{A}}$. Define the function $[\text{D:Data}]^{\mathfrak{A}} : \mathscr{E}D \to \mathbb{B}^{2^u}$, assiging a 2^u-bit bitstring to any D-class X by:*

$$[\text{D:Data}]^{\mathfrak{A}}_p X = \begin{cases} 1 & \text{if } [\text{C:data}]^{\mathfrak{A}}(a) = (p-1) \text{ implies } \mathfrak{A} \models [\text{D:}\mathsf{bit}\text{-}1](a) \text{ for all } a \in X \\ 0 & \text{otherwise} \end{cases}$$

for $p \in [1, 2^u]$.

Definition 55. *Define the quantifier-rank-1 $\mathcal{L}^2[\mathrm{D}]$-formula* $[\text{D:}\mathsf{Zero}](\boldsymbol{x})$ *by:*

$$[\text{D:}\mathsf{Zero}](\boldsymbol{x}) = \forall \boldsymbol{y} \Big(\boldsymbol{d}(\boldsymbol{y}, \boldsymbol{x}) \to [\text{D:}\mathsf{bit}\text{-}0](\boldsymbol{y}) \Big).$$

Then $\mathfrak{A} \models [\text{D:}\mathsf{Zero}](a)$ iff the data at the D-class of a encodes 0: $\underline{[\text{D:Data}]^{\mathfrak{A}} D[a]} = 0$.

[1] this is reminiscent to the process of defining a standard translation of monadic logic to the two-variable first-order fragment

Definition 56. *Define the quantifier-rank-1 $\mathcal{L}^2[\mathrm{D}]$-formula* $\textsf{[D:Largest]}(\boldsymbol{x})$ *by:*

$$\textsf{[D:Largest]}(\boldsymbol{x}) = \forall \boldsymbol{y}\Big(\boldsymbol{d}(\boldsymbol{y},\boldsymbol{x}) \to \textsf{[D:bit-1]}(\boldsymbol{y})\Big).$$

Then $\mathfrak{A} \vDash \textsf{[D:Zero]}(a)$ iff the data at the D-class of a encodes the largest 2^u-bit number: $\underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[a]} = N_{2^u}$.

Definition 57. *Let $M \in \mathbb{B}_{2^u}$ be a t-bit number (where $t \leq 2^u$). Define the $\mathcal{L}^2[\mathrm{D}]$-formula* $\textsf{[D:Eq-}M\textsf{]}(\boldsymbol{x})$ *by:*

$$\textsf{[D:Eq-}M\textsf{]}(\boldsymbol{x}) = \forall \boldsymbol{y}\Bigg(\boldsymbol{d}(\boldsymbol{y},\boldsymbol{x}) \to \bigwedge_{0\leq p<t} \Big(\textsf{[D:pos-}p\textsf{]}(\boldsymbol{y}) \to \textsf{[D:bit-}(\overline{M}_{p+1})\textsf{]}(\boldsymbol{y})\Big) \wedge$$
$$\forall \boldsymbol{x}\Big(\textsf{[D:pos-}(t-1)\textsf{]}(\boldsymbol{y}) \wedge \textsf{[D:pos-less]}(\boldsymbol{y},\boldsymbol{x}) \to \textsf{[D:bit-0]}(\boldsymbol{x})\Big)\Bigg).$$

The first part of this formula asserts that the bits at the first t positions of the $\boldsymbol{d}$-class of $\boldsymbol{x}$ encode the number M. The second part asserts that all the remaining bits at larger positions are zeroes. Note that the length of this formula is polynomially bounded by t, the bitsize of M. We have $\mathfrak{A} \vDash \textsf{[D:Eq-}M\textsf{]}(a)$ iff the data at the D-class of a encodes M: $\underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[a]} = M$.

Definition 58. *Define the $\mathcal{L}^6[\mathrm{D}]$-formula* $\textsf{[D:Less]}(\boldsymbol{x},\boldsymbol{y})$ *by:*

$$\textsf{[D:Less]}(\boldsymbol{x},\boldsymbol{y}) = \exists \boldsymbol{x}'\exists \boldsymbol{y}'\Bigg(\boldsymbol{d}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{d}(\boldsymbol{y}',\boldsymbol{y}) \wedge$$
$$\Big(\textsf{[D:pos-eq]}(\boldsymbol{x}',\boldsymbol{y}') \wedge \textsf{[D:bit-0]}(\boldsymbol{x}') \wedge \textsf{[D:bit-1]}(\boldsymbol{y}')\Big) \wedge \tag{Less1}$$
$$\forall \boldsymbol{x}''\Big(\textsf{[D:pos-less]}(\boldsymbol{x}',\boldsymbol{x}'') \to \exists \boldsymbol{y}''\Big(\boldsymbol{d}(\boldsymbol{y}'',\boldsymbol{y}') \wedge$$
$$\textsf{[D:pos-eq]}(\boldsymbol{y}'',\boldsymbol{x}'') \wedge (\textsf{[D:bit-0]}(\boldsymbol{y}'') \leftrightarrow \textsf{[D:bit-0]}(\boldsymbol{x}''))\Big)\Big)\Bigg). \tag{Less2}$$

Then $\mathfrak{A} \vDash \textsf{[D:Less]}(a,b)$ iff $\underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[a]} < \underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[b]}$. By rearrangement and reusing variables, this can be also written using just three variables (but not using just two variables). Indeed, $\textsf{[D:Less]}(\boldsymbol{x},\boldsymbol{y})$ is logically equivalent to:

$$\exists \boldsymbol{z}\Bigg(\boldsymbol{d}(\boldsymbol{z},\boldsymbol{x}) \wedge \exists \boldsymbol{x}\Bigg(\boldsymbol{x} = \boldsymbol{z} \wedge \exists \boldsymbol{z}\Bigg(\boldsymbol{d}(\boldsymbol{z},\boldsymbol{y}) \wedge \exists \boldsymbol{y}\Bigg(\boldsymbol{y} = \boldsymbol{z} \wedge$$
$$\Big(\textsf{[D:pos-eq]}(\boldsymbol{x},\boldsymbol{y}) \wedge \textsf{[D:bit-0]}(\boldsymbol{x}) \wedge \textsf{[D:bit-1]}(\boldsymbol{y})\Big) \wedge \tag{Less1}$$
$$\forall \boldsymbol{z}\Big(\textsf{[D:pos-less]}(\boldsymbol{x},\boldsymbol{z}) \to \exists \boldsymbol{x}\Big(\boldsymbol{x} = \boldsymbol{z} \wedge \exists \boldsymbol{z}\Big(\boldsymbol{d}(\boldsymbol{z},\boldsymbol{y}) \wedge \exists \boldsymbol{y}\Big(\boldsymbol{y} = \boldsymbol{z} \wedge$$
$$\textsf{[D:pos-eq]}(\boldsymbol{y},\boldsymbol{x}) \wedge (\textsf{[D:bit-0]}(\boldsymbol{y}) \leftrightarrow \textsf{[D:bit-0]}(\boldsymbol{x}))\Big)\Big)\Big)\Big)\Bigg)\Bigg)\Bigg)\Bigg). \tag{Less2}$$

Definition 59. *Define the* $\mathcal{L}^6[\mathrm{D}]$*-formula* $[\mathsf{D{:}Succ}](\boldsymbol{x},\boldsymbol{y})$ *by:*

$$[\mathsf{D{:}Succ}](\boldsymbol{x},\boldsymbol{y}) = \exists\boldsymbol{x}'\exists\boldsymbol{y}'\Big(\boldsymbol{d}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{d}(\boldsymbol{y}',\boldsymbol{y}) \wedge$$

$$\Big([\mathsf{D{:}pos\text{-}eq}](\boldsymbol{x}',\boldsymbol{y}') \wedge [\mathrm{D{:}bit\text{-}0}](\boldsymbol{x}') \wedge [\mathrm{D{:}bit\text{-}1}](\boldsymbol{y}')\Big) \wedge \tag{Succ1}$$

$$\forall\boldsymbol{x}''\Big([\mathsf{D{:}pos\text{-}less}](\boldsymbol{x}'',\boldsymbol{x}') \to [\mathrm{D{:}bit\text{-}1}](\boldsymbol{x}'')\Big) \wedge \tag{Succ2}$$

$$\forall\boldsymbol{y}''\Big([\mathsf{D{:}pos\text{-}less}](\boldsymbol{y}'',\boldsymbol{y}') \to [\mathrm{D{:}bit\text{-}0}](\boldsymbol{y}'')\Big) \wedge \tag{Succ3}$$

$$\forall\boldsymbol{x}''\Big([\mathsf{D{:}pos\text{-}less}](\boldsymbol{x}',\boldsymbol{x}'') \to \exists\boldsymbol{y}''\Big(\boldsymbol{d}(\boldsymbol{y}'',\boldsymbol{y}') \wedge$$

$$[\mathsf{D{:}pos\text{-}eq}](\boldsymbol{y}'',\boldsymbol{x}'') \wedge ([\mathrm{D{:}bit\text{-}0}](\boldsymbol{y}'') \leftrightarrow [\mathrm{D{:}bit\text{-}0}](\boldsymbol{x}''))\Big)\Big)\Big). \tag{Succ4}$$

By rearrangement and reusing variables, this can be also written using just three variables (but not using just two variables).

Then $\mathfrak{A} \models [\mathsf{D{:}Succ}](a,b)$ iff $\underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[b]} = 1 + \underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} D[a]}$.

Definition 60. *Define the* $\mathcal{L}^3[\mathrm{D}]$*-sentence* $[\mathsf{D{:}Full}]$ *by:*

$$[\mathsf{D{:}Full}] = \exists\boldsymbol{x}[\mathsf{D{:}Zero}](\boldsymbol{x}) \wedge \forall\boldsymbol{x}\Big(\neg[\mathsf{D{:}Largest}](\boldsymbol{x}) \to \exists\boldsymbol{y}[\mathsf{D{:}Succ}](\boldsymbol{x},\boldsymbol{y})\Big).$$

If $\mathfrak{A}$ satisfies $[\mathsf{D{:}Full}]$ then $\mathfrak{A}$ contains a $\boldsymbol{d}$-class of encoding any possible data: for every $M \in [0, N_{2^u}]$, there is a $\boldsymbol{d}$-class X such that $\underline{[\mathrm{D{:}Data}]^{\mathfrak{A}} X} = M$.

Definition 61. *Define the* $\mathcal{L}^4[\mathrm{D}]$*-formula* $[\mathsf{D{:}Eq}](\boldsymbol{x},\boldsymbol{y})$ *by:*

$$[\mathsf{D{:}Eq}](\boldsymbol{x},\boldsymbol{y}) = \forall\boldsymbol{x}'\forall\boldsymbol{y}'\Big(\boldsymbol{d}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{d}(\boldsymbol{y}',\boldsymbol{y}) \wedge$$

$$[\mathsf{D{:}pos\text{-}eq}](\boldsymbol{x}',\boldsymbol{y}') \to ([\mathrm{D{:}bit\text{-}0}](\boldsymbol{x}') \leftrightarrow [\mathrm{D{:}bit\text{-}0}](\boldsymbol{y}'))\Big).$$

By rearrangement and reusing variables, this can be also written using just three variables (but not using just two variables).

Then $\mathfrak{A} \models [\mathsf{D{:}Eq}](\boldsymbol{x},\boldsymbol{y})$ iff $[\mathrm{D{:}Data}]^{\mathfrak{A}} D[a] = [\mathrm{D{:}Data}]^{\mathfrak{A}} D[b]$.

Definition 62. *Define the* $\mathcal{L}^4[\mathrm{D}]$*-sentence* $[\mathsf{D{:}Alldiff}]$ *by:*

$$[\mathsf{D{:}Alldiff}] = \forall\boldsymbol{x}\forall\boldsymbol{y}\Big(\neg\boldsymbol{d}(\boldsymbol{x},\boldsymbol{y}) \to \exists\boldsymbol{x}'\exists\boldsymbol{y}'\Big(\boldsymbol{d}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{d}(\boldsymbol{y}',\boldsymbol{y}) \wedge$$

$$[\mathsf{D{:}pos\text{-}eq}](\boldsymbol{x}',\boldsymbol{y}') \wedge \neg([\mathrm{D{:}bit\text{-}0}](\boldsymbol{x}') \leftrightarrow [\mathrm{D{:}bit\text{-}0}](\boldsymbol{y}'))\Big)\Big).$$

By rearrangement and reusing variables, this can be also written using just three variables (but not using just two variables).

If $\mathfrak{A}$ satisfies [D:Alldiff] then all D-classes in $\mathfrak{A}$ encode different data.

Recall from Section 1.7 that an instance of the *doubly exponential tiling problem* is an initial condition $c^0 = \langle t_1^0, t_2^0, \ldots, t_n^0 \rangle \subseteq T = [1, k]$ of tiles from the "Turing-complete" domino system $D_0 = (T, H, V)$, where $H, V \subseteq T \times T$ are the horizontal and vertical matching relations. We need to define a predicate signature capable enough to express a doubly exponential grid of tiles. Consider the predicate signature

$$\mathrm{D} = \Big\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H; \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V; \boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_k^T; \boldsymbol{d} \Big\rangle .$$

It has the following relevant subsignatures:

- $\mathrm{D}^H = \Big\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H, \boldsymbol{d} \Big\rangle$ encodes the horizontal index of a tile
- $\mathrm{D}^V = \Big\langle \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V, \boldsymbol{d} \Big\rangle$ encodes the vertical index of a tile
- $\mathrm{D}^{HV} = \Big\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H, \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V, \boldsymbol{d} \Big\rangle$ encodes the combined horizontal and vertical index of a tile; we need this to define the full grid
- $\mathrm{D}^T = \Big\langle \boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_k^T \Big\rangle$ encodes the type of a tile.

Let $\mathfrak{A}$ be a D-structure satisfying [D^{HV}:pos-full] and let $D = \boldsymbol{d}^{\mathfrak{A}}$. The sentence

$$[\mathrm{D}^{HV}\textsf{:Full}] \wedge [\mathrm{D}^{HV}\textsf{:Alldiff}] \tag{5.3}$$

asserts that the D-classes form a doubly exponential grid. The sentence

$$\forall \boldsymbol{x} \Big(\bigwedge_{1 \le i \le k} \boldsymbol{u}_i^T(\boldsymbol{x}) \to \bigwedge_{i < j \le k} \neg \boldsymbol{u}_j^T(\boldsymbol{x}) \Big) \tag{5.4}$$

asserts that every element has a unique type. The sentence

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \Big(\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \to \bigwedge_{1 \le i \le k} (\boldsymbol{u}_i^T(\boldsymbol{x}) \leftrightarrow \boldsymbol{u}_i^T(\boldsymbol{x})) \Big) \tag{5.5}$$

asserts that all elements in a D-class have the same type—the type of the tile corresponding to that D-class. For $j \in [1, n]$, the sentence

$$\forall \boldsymbol{x} \Big([\mathrm{D}^H\textsf{:Eq-}(j-1)](\boldsymbol{x}) \wedge [\mathrm{D}^V\textsf{:Zero}](\boldsymbol{x}) \to \boldsymbol{u}_{t_j^0}^T(\boldsymbol{x}) \Big) \tag{5.6}$$

encodes the initial segment in the first row of the square. The sentence

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \Big([\mathrm{D}^H\textsf{:Succ}](\boldsymbol{x}, \boldsymbol{y}) \wedge [\mathrm{D}^V\textsf{:Eq}](\boldsymbol{x}, \boldsymbol{y}) \to \bigvee_{(i,j) \in H} \boldsymbol{u}_i^T(\boldsymbol{x}) \wedge \boldsymbol{u}_j^T(\boldsymbol{y}) \Big) \tag{5.7}$$

encodes the horizontal matching condition. The sentence

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \Big([\mathrm{D}^V\textsf{:Succ}](\boldsymbol{x}, \boldsymbol{y}) \wedge [\mathrm{D}^H\textsf{:Eq}](\boldsymbol{x}, \boldsymbol{y}) \to \bigvee_{(i,j) \in V} \boldsymbol{u}_i^T(\boldsymbol{x}) \wedge \boldsymbol{u}_j^T(\boldsymbol{y}) \Big) \tag{5.8}$$

encodes the vertical matching condition.

Combining $[\mathrm{D}^{HV}\textsf{:pos-full}]$ with the formulas 5.3–5.8, we may encode the instance an instance of the doubly exponential tiling problem as a (finite) satisfiability of a formula, so we have:

Proposition 8. *The (finite) satisfiability problem for the monadic first-order logic with a single equivalence symbol $\mathcal{L}_1 1\mathrm{E}$ is* N2ExpTime*-hard. More precisely, even the three-variable fragment $\mathcal{L}_1^3 1\mathrm{E}$ has this property.*

5.5 Hardness with many equivalences in refinement

The argument from the previous section can be iterated to yield the hardness of the (finite) satisfiability of the monadic first-order logic with several builtin equivalence symbols in refinement $\mathcal{L}_1 e\mathrm{E}_{\textsf{refine}}$. Our strategy is to encode $(e+1)$-exponential numbers at every equivalence class of the coarsest relation by thinking of the e-exponential numbers at the classes of the second-to-coarsest relation as bit positions.

For $e \in \mathbb{N}^+$, consider the predicate signature $\mathrm{E}(e) = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$ consisting of the builtin equivalence symbols $\boldsymbol{e}_i$ in refinement. Abbreviate the *coarsest* equivalence symbol $\boldsymbol{d} = \boldsymbol{e}_e$.

Definition 63. *Let $e \in \mathbb{N}^+$. An* e-exponential setup *is a uniform effective polynomial-time process for creating the following data structure. For every $u \in \mathbb{N}^+$, there is a predicate signature $\mathrm{D}(e,u)$ having length polynomial in u, consisting of unary predicate symbols and containing $\mathrm{E}(e)$. The following data is effectively defined:*

E1 There is a $\mathcal{L}^3[\mathrm{D}(e,u)]$-sentence $[\mathrm{D}(e,u)\textsf{:pos-full}]$, whose length grows polynomially as u grows.

E2 If $\mathfrak{A}$ is a $\mathrm{D}(e,u)$-structure, $\mathfrak{A} \vDash [\mathrm{D}(e,u)\textsf{:pos-full}]$ and $D = \boldsymbol{d}^{\mathfrak{A}}$, then there is a function $[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} : \mathscr{E}D \to \mathbb{B}^{\exp_2^e(u)}$ that assigns an e-exponential bitstring to every D-class.

E3 There is a $\mathcal{L}^3[\mathrm{D}(e,u)]$-formula $[\mathrm{D}(e,u)\textsf{:Eq}](\boldsymbol{x},\boldsymbol{y})$ whose length grows polynomially as u grows, such that for all $a, b \in A$:

$$\mathfrak{A} \vDash [\mathrm{D}(e,u)\textsf{:Eq}](a,b) \textit{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} = \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[b]}.$$

E4 There is a $\mathcal{L}^3[\mathrm{D}(e,u)]$-formula $[\mathrm{D}(e,u)\textsf{:Zero}](\boldsymbol{x})$, whose length grows polynomially as u grows, such that for all $a \in A$:

$$\mathfrak{A} \vDash [\mathrm{D}(e,u)\textsf{:Zero}](a) \textit{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} = 0.$$

E5 There is a $\mathcal{L}^3[\mathrm{D}(e,u)]$-formula $[\mathrm{D}(e,u)\textsf{:Largest}](\boldsymbol{x})$, whose length grows polynomially as u grows, such that for all $a \in A$:

$$\mathfrak{A} \vDash [\mathrm{D}(e,u)\textsf{:Largest}](a) \textit{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} = N_{\exp_2^e(u)} = \exp_2^{e+1}(u) - 1.$$

E6 *There is a* $\mathcal{L}^3[\mathrm{D}(e,u)]$*-formula* $[\mathrm{D}(e,u)\mathbf{:}\mathsf{Less}](\boldsymbol{x},\boldsymbol{y})$*, whose length grows polynomially as* u *grows, such that for all* $a, b \in A$*:*

$$\mathfrak{A} \models [\mathrm{D}(e,u)\mathbf{:}\mathsf{Less}](a,b) \text{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} < \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[b]}.$$

E7 *There is a* $\mathcal{L}^3[\mathrm{D}(e,u)]$*-formula* $[\mathrm{D}(e,u)\mathbf{:}\mathsf{Succ}](\boldsymbol{x},\boldsymbol{y})$*, whose length grows polynomially as* u *grows, such that for all* $a, b \in A$*:*

$$\mathfrak{A} \models [\mathrm{D}(e,u)\mathbf{:}\mathsf{Succ}](a,b) \text{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[b]} = \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} + 1.$$

E8 *For every* $\exp_2^e(u)$*-bit number* M*, there is a* $\mathcal{L}^3[\mathrm{D}(e,u)]$*-formula* $[\mathrm{D}(e,u)\mathbf{:}\mathsf{Eq}\text{-}M](\boldsymbol{x})$*, whose length grows polynomially as* u *and* M *grow, such that for all* $a \in A$*:*

$$\mathfrak{A} \models [\mathrm{D}(e,u)\mathbf{:}\mathsf{Eq}\text{-}M](a) \text{ iff } \underline{[\mathrm{D}(e,u)\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} D[a]} = M.$$

The last section defines a 1-exponential setup. Suppose that we have an e-exponential setup having predicate signature $\mathrm{D} = \mathrm{D}(e,u)$. Analogously to the previous section, we will describe an $(e+1)$-exponential setup $\mathrm{D}' = \mathrm{D}(e+1,u) = \mathrm{D} + \langle \boldsymbol{e} \rangle$ which is based on D, where $\boldsymbol{e} = \boldsymbol{e}_{e+1}$ is the new coarsest builtin equivalence symbol in D'. Define the following formulas:

$$\begin{aligned}
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{x},\boldsymbol{y}) = [\mathrm{D}\mathbf{:}\mathsf{Eq}](\boldsymbol{x},\boldsymbol{y})\\
&[\mathrm{D}'\mathbf{:}\mathsf{bit\text{-}0}](\boldsymbol{x}) = \forall \boldsymbol{y}(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \wedge [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{y},\boldsymbol{x}) \to \boldsymbol{y} = \boldsymbol{x})\\
&[\mathrm{D}'\mathbf{:}\mathsf{bit\text{-}1}](\boldsymbol{x}) = \exists \boldsymbol{y}(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \wedge [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}eq}](\boldsymbol{y},\boldsymbol{x}) \wedge \boldsymbol{y} \neq \boldsymbol{x})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}zero}](\boldsymbol{x}) = [\mathrm{D}\mathbf{:}\mathsf{Zero}](\boldsymbol{x})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}largest}](\boldsymbol{x}) = [\mathrm{D}\mathbf{:}\mathsf{Largest}](\boldsymbol{x})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}less}](\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{e}(\boldsymbol{x},\boldsymbol{y}) \wedge [\mathrm{D}\mathbf{:}\mathsf{Less}](\boldsymbol{x},\boldsymbol{y})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}succ}](\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{e}(\boldsymbol{x},\boldsymbol{y}) \wedge [\mathrm{D}\mathbf{:}\mathsf{Succ}](\boldsymbol{x},\boldsymbol{y})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}full}] = \forall \boldsymbol{x} \exists \boldsymbol{y}\Big(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \wedge [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}zero}](\boldsymbol{y})\Big) \wedge \qquad \text{(E1)}\\
&\quad \forall \boldsymbol{x}\Big(\neg[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}largest}](\boldsymbol{x}) \to \exists \boldsymbol{y}[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}succ}](\boldsymbol{x},\boldsymbol{y})\Big)\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}0}](\boldsymbol{x}) = [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}zero}](\boldsymbol{x})\\
&[\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}}(p+1)](\boldsymbol{x}) = \exists \boldsymbol{y}\Big([\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}}p](\boldsymbol{y}) \wedge [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}succ}](\boldsymbol{y},\boldsymbol{x})\Big)\\
&\quad \text{for } p \in [0, N_{\exp_2^e(u)} - 1].
\end{aligned}$$

Let $\mathfrak{A}$ be a D'-structure, $\mathfrak{A} \models [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}full}]$ and let $E = \boldsymbol{e}^{\mathfrak{A}}$. Define the function $[\mathrm{D}'\mathbf{:}\mathrm{Data}]^{\mathfrak{A}} : \mathscr{E}E \to \mathbb{B}^{\exp_2^{e+1}(u)}$ assiging a $\exp_2^{e+1}(u)$-bit bitstring to any E-class X by:

$$[\mathrm{D}'\mathbf{:}\mathrm{Data}]^{\mathfrak{A}}_p X = \begin{cases} 1 \text{ if } \mathfrak{A} \models [\mathrm{D}'\mathbf{:}\mathsf{pos\text{-}}(p-1)](a) \text{ implies } \mathfrak{A} \models [\mathrm{D}'\mathbf{:}\mathsf{bit\text{-}1}](a) \text{ for all } a \in X \\ 0 \text{ otherwise} \end{cases} \qquad \text{(E2)}$$

for $p \in [1, \exp_2^{e+1}(u)]$.

Define the following formulas:

$$[\mathrm{D}'\textsf{:Eq}](\boldsymbol{x},\boldsymbol{y}) = \forall \boldsymbol{x}' \forall \boldsymbol{y}' \Big(\boldsymbol{e}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{e}(\boldsymbol{y}',\boldsymbol{y}) \wedge \\ [\mathrm{D}'\textsf{:pos-eq}](\boldsymbol{x}',\boldsymbol{y}') \to ([\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}') \leftrightarrow [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{y}')) \Big) \tag{E3}$$

$$[\mathrm{D}'\textsf{:Zero}](\boldsymbol{x}) = \forall \boldsymbol{y} \Big(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \to [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{y}) \Big) \tag{E4}$$

$$[\mathrm{D}'\textsf{:Largest}](\boldsymbol{x}) = \forall \boldsymbol{y} \Big(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \to [1\textsf{:bit-}\mathrm{D}'](\boldsymbol{y}) \Big) \tag{E5}$$

$$[\mathrm{D}'\textsf{:Less}](\boldsymbol{x},\boldsymbol{y}) = \exists \boldsymbol{x}' \exists \boldsymbol{y}' \Bigg(\boldsymbol{e}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{e}(\boldsymbol{y}',\boldsymbol{y}) \wedge \\ \Big([\mathrm{D}'\textsf{:pos-eq}](\boldsymbol{x}',\boldsymbol{y}') \wedge [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}') \wedge [\mathrm{D}'\textsf{:bit-1}](\boldsymbol{y}') \Big) \wedge \\ \forall \boldsymbol{x}'' \Big([\mathrm{D}'\textsf{:pos-less}](\boldsymbol{x}',\boldsymbol{x}'') \to \exists \boldsymbol{y}'' \Big(\boldsymbol{e}(\boldsymbol{y}'',\boldsymbol{y}') \wedge \\ [\mathrm{D}'\textsf{:pos-eq}](\boldsymbol{y}'',\boldsymbol{x}'') \wedge ([\mathrm{D}'\textsf{:bit-0}](\boldsymbol{y}'') \leftrightarrow [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}'')) \Big) \Big) \Bigg) \tag{E6}$$

$$[\mathrm{D}'\textsf{:Succ}](\boldsymbol{x},\boldsymbol{y}) = \exists \boldsymbol{x}' \exists \boldsymbol{y}' \Bigg(\boldsymbol{e}(\boldsymbol{x}',\boldsymbol{x}) \wedge \boldsymbol{e}(\boldsymbol{y}',\boldsymbol{y}) \wedge \\ \Big([\mathrm{D}'\textsf{:pos-eq}](\boldsymbol{x}',\boldsymbol{y}') \wedge [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}') \wedge [\mathrm{D}'\textsf{:bit-1}](\boldsymbol{y}') \Big) \wedge \\ \forall \boldsymbol{x}'' \Big([\mathrm{D}'\textsf{:pos-less}](\boldsymbol{x}'',\boldsymbol{x}') \to [\mathrm{D}'\textsf{:bit-1}](\boldsymbol{x}'') \Big) \wedge \\ \forall \boldsymbol{y}'' \Big([\mathrm{D}'\textsf{:pos-less}](\boldsymbol{y}'',\boldsymbol{y}') \to [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{y}'') \Big) \wedge \\ \forall \boldsymbol{x}'' \Big([\mathrm{D}'\textsf{:pos-less}](\boldsymbol{x}',\boldsymbol{x}'') \to \exists \boldsymbol{y}'' \Big(\boldsymbol{e}(\boldsymbol{y}'',\boldsymbol{y}') \wedge \\ [\mathrm{D}'\textsf{:pos-eq}](\boldsymbol{y}'',\boldsymbol{x}'') \wedge ([\mathrm{D}'\textsf{:bit-0}](\boldsymbol{y}'') \leftrightarrow [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}'')) \Big) \Big) \Bigg). \tag{E7}$$

If $M \in \mathbb{B}_{\exp_2^{e+1}(u)}$ is a $\exp_2^{e+1}(u)$-bit number, where $t = \|M\|$, define the formula

$$[\mathrm{D}'\textsf{:Eq-}M](\boldsymbol{x}) = \forall \boldsymbol{y} \Bigg(\boldsymbol{e}(\boldsymbol{y},\boldsymbol{x}) \to \bigwedge_{0 \le p < t} \Big([\mathrm{D}'\textsf{:pos-}p](\boldsymbol{y}) \to [\mathrm{D}'\textsf{:bit-}\overline{M}_{p+1}](\boldsymbol{y}) \Big) \wedge \\ \forall \boldsymbol{x} \Big([\mathrm{D}'\textsf{:pos-}(t-1)](\boldsymbol{y}) \wedge [\mathrm{D}'\textsf{:pos-less}](\boldsymbol{y},\boldsymbol{x}) \to [\mathrm{D}'\textsf{:bit-0}](\boldsymbol{x}) \Big) \Bigg). \tag{E8}$$

This completes the definition of the $(e+1)$-exponential setup.

We can encode an instance of the $(e+1)$-exponential tiling problem into a (finite) satisfiability D-formula completely analogously to the previous section. Thus we have:

Proposition 9. *The (finite) satisfiability problem for the monadic first-order logic with e equivalence symbols in refinement $\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$ is* N(e+1)EXPTIME*-hard. Even the three-variable fragment $\mathcal{L}_1^3 e\mathrm{E}_{\mathsf{refine}}$ has this property.*

By Proposition 1 and Proposition 3, the same holds for $\mathcal{L}_1^{(3)} e\mathrm{E}_{\mathsf{global}}$ and $\mathcal{L}_1^{(3)} e\mathrm{E}_{\mathsf{local}}$.

Proposition 10. *The (finite) satisfiability problem for the monadic first-order logic with many equivalence symbols in refinement $\mathcal{L}_1\mathrm{E}_{\mathsf{refine}}$ is* ELEMENTARY*-hard. Even the three-variable fragment $\mathcal{L}_1^3\mathrm{E}_{\mathsf{refine}}$ has this property.*

By Proposition 2 and Proposition 4, the same holds for $\mathcal{L}_1^{(3)}\mathrm{E}_{\mathsf{global}}$ and $\mathcal{L}_1^{(3)}\mathrm{E}_{\mathsf{local}}$.

6 Two-variable logics

TODO: Introduction

6.1 Type realizibility of the two-variable first-order logic

Recall from Section 1.6 about normal forms that every $\mathcal{L}^2$-sentence φ can be reduced in polynomial time sentence $\operatorname{sctr}\varphi$ in Scott normal form:

$$\forall \boldsymbol{x} \forall \boldsymbol{y}(\alpha_0(\boldsymbol{x},\boldsymbol{y}) \lor \boldsymbol{x} = \boldsymbol{y}) \land \bigwedge_{1 \le i \le m} \forall \boldsymbol{x} \exists \boldsymbol{y}(\alpha_i(\boldsymbol{x},\boldsymbol{y}) \land \boldsymbol{x} \neq \boldsymbol{y}),$$

where the formulas α_i are quantifier-free and use at most linearly many new unary predicate symbols. We refer to α_0 as the *universal part*, or the $\forall\forall$-part of the formula $\operatorname{sctr}\varphi$ and to α_i as the *existential parts*, or the $\forall\exists$-parts of the formula, for $i \in [1, m]$. For any formula in Scott normal form, we may replace its existential parts by fresh binary predicate symbols: for $i \in [1, m]$, let $\boldsymbol{m}_i$ be a fresh binary predicate symbol with the intended interpretation $\forall \boldsymbol{x} \forall \boldsymbol{y}(\boldsymbol{m}_i(\boldsymbol{x},\boldsymbol{y}) \leftrightarrow \alpha_i(\boldsymbol{x},\boldsymbol{y}))$. Since this is a universal sentence, it can be incorporated into the $\forall\forall$-part α_0 of the formula. We refer to the symbols $\boldsymbol{m}_i$ as the *message symbols*. Hence the formula can be transformed in polynomial time to the form:

$$\forall \boldsymbol{x} \forall \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \lor \boldsymbol{x} = \boldsymbol{y}) \land \bigwedge_{1 \le i \le m} \forall \boldsymbol{x} \exists \boldsymbol{y}(\boldsymbol{m}_i(\boldsymbol{x},\boldsymbol{y}) \land \boldsymbol{x} \neq \boldsymbol{y}), \tag{6.1}$$

where the $\forall\forall$-part α is quantifier-free and over an extended signature. For convenience, we make the existential parts of the formula part of the signature, so we can focus only on the universal part. The following is a term similar to the one defined in [4]:

Definition 64. *A* classified signature $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *for the two-variable first-order logic* $\mathcal{L}^2$ *is a predicate signature* Σ *together with a sequence* $\bar{\boldsymbol{m}} = \boldsymbol{m}_1 \boldsymbol{m}_2 \ldots \boldsymbol{m}_m$ *of distinct binary predicate symbols from* Σ *having intended interpretation*

$$\bigwedge_{1 \le i \le m} \forall \boldsymbol{x} \exists \boldsymbol{y}(\boldsymbol{m}_i(\boldsymbol{x},\boldsymbol{y}) \land \boldsymbol{x} \neq \boldsymbol{y}). \tag{6.2}$$

That is, a classified signature *automatically includes* the $\forall\exists$-part of formulas and $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$-structures automatically satisfy the $\forall\exists$-part.

Definition 65. *The* (finite) classified satisfiability problem for two-variable first-order logic *is: given a classified signature* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *and a quantifier-free* $\mathcal{L}^2[\Sigma]$*-formula* $\alpha(\boldsymbol{x},\boldsymbol{y})$*, is there a (finite)* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$*-structure satisfying eq.* (6.1)*. Denote tha classified satisfiability problem for two-variable first-order logic by* CLSAT$\mathcal{L}^2$ *and its finite version by* FINCLSAT$\mathcal{L}^2$*.*

Scott normal form shows that (finite) satisfiability reduces in polynomial time to (finite) classified satisfiability:

$$\text{(FIN)SAT}\mathcal{L}^2 \leq_{\mathrm{m}}^{\text{PTime}} \text{(FIN)CLSAT}\mathcal{L}^2.$$

Let $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ be a classified signature for $\mathcal{L}^2$.

Definition 66. *A* type instance (Π, T) *over* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *is a pair of a set of* 1*-types* $\Pi \subseteq \Pi[\Sigma]$ *and a set of* 2*-types* $\mathrm{T} \subseteq \mathrm{T}[\Sigma]$ *such that* $\mathrm{tp}_{\boldsymbol{x}}\, \tau \in \Pi$ *and* $\mathrm{tp}_{\boldsymbol{y}}\, \tau \in \Pi$ *for all* $\tau \in \mathrm{T}$.

A $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$*-structure* $\mathfrak{A}$ realizes *(or is a model of)* (Π, T) *if it contains at least* 2 *elements,* $\mathrm{tp}^{\mathfrak{A}}[a] \in \Pi$ *for all* $a \in A$ *and* $\mathrm{tp}^{\mathfrak{A}}[a, b] \in \mathrm{T}$ *for all* $a \in A$ *and* $b \in A \setminus \{a\}$*. The structure* fully realizes (Π, T) *if additionally for every* 1*-type* $\pi \in \Pi$ *there is* $a \in A$ *such that* $\mathrm{tp}^{\mathfrak{A}}[a] = \pi$ *(we don't require an analogous condition for the* 2*-types). The* characteristic type instance *of* $\mathfrak{A}$ *is the type instance* (Π, T) *defined by:*

$$\Pi = \left\{ \mathrm{tp}^{\mathfrak{A}}[a] \,\middle|\, a \in A \right\}$$
$$\mathrm{T} = \left\{ \mathrm{tp}^{\mathfrak{A}}[a, b] \,\middle|\, a \neq b \in A \right\}.$$

The (finite) (full) type realizibility problem for $\mathcal{L}^2$ *is the following: given a classified signature* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *and a type instance* (Π, T) *over* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$*, is there a (finite)* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$*-structure that (fully) realizes* (Π, T)*. Denote the type realizibility problem for* $\mathcal{L}^2$ *by* $\text{REAL}\mathcal{L}^2$ *and its finite version by* $\text{FINREAL}\mathcal{L}^2$*. Denote the full type realizibility problem for* $\mathcal{L}^2$ *by* $\text{FULLREAL}\mathcal{L}^2$ *and its finite version by* $\text{FINFULLREAL}\mathcal{L}^2$*.*

Remark 22. *Since the set of* 1*-types realized in any model of* (Π, T) *is a subset of* Π*, by guessing this subset we can reduce the (finite) type satisfiability problem to the (finite) full type satisfiability problem in nondeterministic polynomial time:*

$$\text{(FIN)REAL}\mathcal{L}^2 \leq_{\mathrm{m}}^{\text{NPTime}} \text{(FIN)FULLREAL}\mathcal{L}^2.$$

Remark 23. *Let* $\alpha(\boldsymbol{x}, \boldsymbol{y})$ *be a quantifier-free* $\mathcal{L}^2[\Sigma]$*-formula. Let* Π *be the set of those* 1*-types over* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *that are consistent with* α *and* T *be the set of those* 2*-types over* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ *that are consistent with* α *and the intended interpretation eq.* (6.2)*. Then a* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$*-structure* $\mathfrak{A}$ *is a model for* α *iff it is a model for* (Π, T)*.*

Recall that the number of possible 1*-types or* 2*-types over* Σ *is exponentially bounded by the length* s *of* Σ *and that the cardinality of a* 1*-type or a* 2*-type over* Σ *is polynomially bounded by* s*. Hence we can reduce the (finite) classified satisfiability problem to the (finite) type realizibility problem in exponential time:*

$$\text{(FIN)SAT}\mathcal{L}^2 \leq_{\mathrm{m}}^{\text{ExpTime}} \text{(FIN)REAL}\mathcal{L}^2.$$

We study the type realizibility problem instead of directly studying the satisfiability problem because it is flexible enough to allow us to employ its solutions in simpler logics to construct a solution in logics with more equivalences.

Let (Π, T) be a type instance over $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$.

Definition 67. *Two* 1*-types* $\pi, \pi' \in \Pi$ *are* connectable *(written* $\pi \sim \pi'$*), if* $\pi = \operatorname{tp}_{\boldsymbol{x}} \tau$ *and* $\pi' = \operatorname{tp}_{\boldsymbol{y}} \tau$ *for some* $\tau \in \mathrm{T}$*. A* 1*-type* $\kappa \in \Pi$ *is a* king type *if* $\kappa \not\sim \kappa$*.*

Remark 24. *If two distinct* 1*-types* $\pi, \pi' \in \Pi$ *are not connectable, then* (Π, T) *is not fully realizible.*

If $\mathfrak{A}$ *is a full model for* (Π, T) *then every two distinct elements* $a \neq b \in A$ *realize connectable* 1*-types. Hence every king type* $\kappa \in \Pi$ *is realized once in* $\mathfrak{A}$*.*

A type instance (Π, T) *is* connected *if every two distinct* 1*-types* $\pi, \pi' \in \Pi$ *are connectable.*

The next definition characterizes the local structure of the ray of 2-types realized around an element of a structure.

Definition 68. *A* star-type $\sigma \subseteq \mathrm{T}$ *for the type instance* (Π, T) *is a nonempty set of* 2*-types satisfying the following conditions:*

1. *If* $\tau, \tau' \in \sigma$ *then* $\operatorname{tp}_{\boldsymbol{x}} \tau = \operatorname{tp}_{\boldsymbol{x}} \tau'$*, that is the* $\boldsymbol{x}$*-type of every element of* σ *is the same. Denote the* $\boldsymbol{x}$*-type of every element of* σ *by* $\operatorname{tp}_{\boldsymbol{x}} \sigma$*.*
2. *If* $\kappa = \operatorname{tp}_{\boldsymbol{x}} \sigma$ *is a king type, then no* $\tau \in \sigma$ *has* $\operatorname{tp}_{\boldsymbol{y}} \tau = \kappa$*.*
3. *If* $\kappa \neq \operatorname{tp}_{\boldsymbol{x}} \sigma$ *is any king type, then one* $\tau \in \sigma$ *has* $\operatorname{tp}_{\boldsymbol{y}} \tau = \kappa$*.*
4. *If* $\boldsymbol{m} \in \bar{\boldsymbol{m}}$*, then* $\boldsymbol{m} \in \tau$ *for some (possibly many)* $\tau \in \sigma$*.*

Note that the size of a star-type is polynomially bounded by the size of the type instance.

If $\mathfrak{A}$ *is a* (Π, T)*-structure and* $a \in A$*, the* star-type realized by a *is:*

$$\operatorname{stp}^{\mathfrak{A}}[a] = \left\{ \operatorname{tp}^{\mathfrak{A}}[a, b] \,\middle|\, b \in A \setminus \{a\} \right\}.$$

It is straighforward to check that this indeed defines a star-type.

Remark 25 (Star-type extension)**.** *Let* σ *be a star-type, with* $\boldsymbol{x}$*-type* $\pi = \operatorname{tp}_{\boldsymbol{x}} \sigma$*. Let* $\tau \in \mathrm{T}$ *be a* 2*-type and suppose that both* $\operatorname{tp}_{\boldsymbol{x}} \tau$ *and* $\operatorname{tp}_{\boldsymbol{y}} \tau$ *are not king types. Then* $\sigma \cup \{\tau\}$ *is also a star-type.*

Definition 69. *A* certificate $\mathcal{S}$ *for the type instance* (Π, T) *is a nonempty set of star-types satisfying the following conditions:*

1. *If* $\kappa \in \Pi$ *is a king type, then one* $\sigma \in \mathcal{S}$ *has* $\operatorname{tp}_{\boldsymbol{x}} \sigma = \kappa$*, that is every king type is witnessed once.*
2. *If* $\pi \in \Pi$ *is not a king type, then some (possibly many)* $\sigma \in \mathcal{S}$ *has* $\operatorname{tp}_{\boldsymbol{x}} \sigma = \pi$*, that is every other* 1*-type is witnessed.*
3. *If* $\tau \in \cup\mathcal{S}$*, then* $\tau^{-1} \in \cup\mathcal{S}$*, that is there are witnesses for the endpoints of every* 2*-type used in the certificate.*

Note that in general the size of a certificate may be exponential in terms of the size of the type instance. However, polynomial certificates exist:

Lemma 7 (Certificate extraction)**.** *Let $\mathfrak{A}$ be a full model for the type instance* (Π, T)*. Let* $\mathrm{T}' \subseteq \mathrm{T}$ *be a (nonempty) set of* 2*-types realized in* $\mathfrak{A}$*. For every* 2*-type* $\tau \in \mathrm{T}'$ *realized in* $\mathfrak{A}$*, let* $(a_\tau, b_\tau) \in A \times A$ *be a pair of distinct elements realizing* τ*. Let*

$$\mathcal{S} = \left\{ \operatorname{stp}^{\mathfrak{A}}[a_\tau], \operatorname{stp}^{\mathfrak{A}}[b_\tau] \;\middle|\; \tau \in \mathrm{T}' \right\}.$$

Then $\mathcal{S}$ *is a certificate for the type instance. Moreover, its size is polynomially bounded by the size of the type instance.*

Proof. We check the conditions for a certificate:

1. If $\kappa \in \Pi$ is a king type, by Remark 24 it is realized once in $\mathfrak{A}$. If $c \in A$ realizes κ and $\sigma \in \mathcal{S}$ has $\operatorname{tp}_{\boldsymbol{x}} \sigma = \kappa$, then $a_\sigma = c$ and hence $\sigma = \operatorname{stp}^{\mathfrak{A}}[c]$ is unique. On the other hand, such $\sigma \in \mathcal{S}$ exists since $\mathfrak{A}$ is a full model for (Π, T) and A has cardinality at least 2.

2. Let $\pi \in \Pi$ be a 1-type. Again, since $\mathfrak{A}$ is a full model for (Π, T) and A has cardinality at least 2, we may choose a pair of distinct elements such that the first element realizes π.

3. The third condition follows immediately from the definition of $\mathcal{S}$.

□

Lemma 8 (Certificate expansion)**.** *Let* $\mathcal{S}$ *be a certificate for the connected type instance* (Π, T)*. Then* (Π, T) *has a finite full model.*

Proof. Let $t = |\mathrm{T}|$. We build a model $\mathfrak{A}$ for (Π, T) as follows. For every star-type $\sigma \in \mathcal{S}$, if $\operatorname{tp}_{\boldsymbol{x}} \sigma$ is a king type, there will be a unique element $c \in A$ that wants to realize σ. Otherwise, there will be $(t+1)$ elements that want to realize σ. This construction fixes the 1-type of every element the structure. Now we need to consistently assign 2-types between all pairs of distinct elements so that they realize their expected star-types.

Let $a \in A$ be an element having expected star-type σ and expected 1-type $\pi = \operatorname{tp}_{\boldsymbol{x}} \sigma$. By the third condition for a certificate, for every 2-type $\tau \in \sigma$ there is a star-type $\sigma_\tau \in \mathcal{S}$ such that $\tau^{-1} \in \sigma_\tau$. We claim that we may choose distinct elements b_τ (also distinct from a), such that b_τ has expected star-type σ_τ. First suppose that $\kappa' = \operatorname{tp}_{\boldsymbol{y}} \tau$ is a king type. By conditions 2 and 3 in the definition of a star-type, $\kappa' \neq \pi$ and there is no other $\tau' \in \sigma$ having $\operatorname{tp}_{\boldsymbol{y}} \tau' = \kappa'$. Then we take b_τ to be the unique element having expected 1-type κ'. Next suppose that $\pi' = \operatorname{tp}_{\boldsymbol{y}} \tau$ is not a king type. Since $\sigma \subseteq \mathrm{T}$ has cardinality at most t, there are at most t types $\tau' \in \sigma$ having $\operatorname{tp}_{\boldsymbol{y}} \tau' = \pi'$. For every such τ' the expected star-type of its target $\sigma' = \sigma_{\tau'}$ has a $\boldsymbol{x}$-type $\operatorname{tp}_{\boldsymbol{x}} \sigma' = \pi'$, which is not a king type, so there are $(t+1)$ elements that have expected star-type σ'. Accounting for the possibly of the element a itself to have expected star-type σ', we can see that there are enough distinct elements b_τ. Fixing $\operatorname{tp}^{\mathfrak{A}}[a, b_\tau] = \tau$ for these elements takes care of the star-type of a. By doing this step-by-step by readjusting the remaining sets of 2-types that need to be realized in order for an element to realize its expected star-type, we may realize every expected star-type of every element in the structure. However, there might still be unassigned 2-types between pairs of distict elements left. Suppose that $a \neq b \in A$ be such a pair. We claim that $\pi = \mathrm{tp}^{\mathfrak{A}}[a]$ and $\pi' = \mathrm{tp}^{\mathfrak{A}}[b]$ are not king types. Without loss of generality, suppose towards a contradiction that π is a king type. Then $\pi \neq \pi'$ and since the type instance is connected, we may find a 2-type $\tau \in \mathrm{T}$ connecting π and π'. We may fix $\mathrm{tp}^{\mathfrak{A}}[a, b] = \tau$ and this will not destroy the model since $\sigma \cup \{\tau\}$ remains a star-type. □

Proposition 11. *The logic $\mathcal{L}^2$ has the finite model property. The (finite) full type realizibility problem for $\mathcal{L}^2$ is in* NPTime.

Proof. Let (Π, T) be a type instance. If it is not connected, then it has no model by Remark 24. It is trivial to check satisfiability in the class of structures of cardinality 1. If this fails to yield a model, guess a certificate of polynomial size (and verify in polynomial time that this is indeed a certificate). By Lemma 7 and Lemma 8 such a certificate exists iff (Π, T) is fully satisfiable. □

Combining this with the reductions given in Remark 22 and Remark 23 gives us another proof of a standard result:

Corollary 3. *The logic $\mathcal{L}^2$ has the finite model property and its (finite) satisfiability problem is in* NExpTime.

Bibliography

[1] H.D. Ebbinghaus and J. Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer Berlin Heidelberg, 1999.

[2] Erich Grädel and Martin Otto. On logics with two variables. *Theoretical computer science*, 224(1):73–113, 1999.

[3] Dana Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27(377):74, 1962.

[4] Ian Pratt-Hartmann. The two-variable fragment with counting and equivalence. *Mathematical Logic Quarterly*, 61(6):474–515, 2015.

[5] I. Cervesato, H. Veith, and A. Voronkov. *Logic for Programming, Artificial Intelligence, and Reasoning: 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008, Proceedings.* Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008.

[6] E. Boerger, E. Grädel, and Y. Gurevich. *The classical decision problem.* Perspectives in mathematical logic. Springer, 1997.

[7] Peter van Emde Boas. The convenience of tilings. *Lecture Notes in Pure and Applied Mathematics*, pages 331–363, 1997.

[8] Leopold Löwenheim. Über möglichkeiten im relativkalkül. *Mathematische Annalen*, 76(4):447–470, 1915.

[9] A. Janiczak. Undecidability of some simple formalized theories. *Fundamenta Mathematicae*, 40(1):131–139, 1953.