

# **Satisfiability with Agreement and Counting**

Krasimir Georgiev

June 8, 2016



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Notation . . . . .	1
1.2	Syntax . . . . .	2
1.3	Semantics . . . . .	4
1.4	Games . . . . .	6
1.5	Types . . . . .	7
1.6	Normal forms . . . . .	8
1.7	Complexity . . . . .	8
<b>2</b>	<b>Counter setups</b>	<b>11</b>
2.1	Bits . . . . .	11
2.2	Counters . . . . .	11
2.3	Vectors . . . . .	13
2.4	Permutations . . . . .	15
<b>3</b>	<b>Equivalence relations</b>	<b>17</b>
3.1	Two equivalence relations in agreement . . . . .	17
3.2	Many equivalence relations in agreement . . . . .	18
<b>4</b>	<b>Reductions</b>	<b>21</b>
4.1	Global agreement to refinement . . . . .	22
4.2	Local agreement to refinement . . . . .	25
4.3	Granularity . . . . .	29
<b>5</b>	<b>Monadic logics</b>	<b>33</b>
5.1	Cells . . . . .	33
5.2	Organs . . . . .	34
5.3	Satisfiability . . . . .	36
5.4	Hardness with a single equivalence . . . . .	37
5.5	Hardness with many equivalences in refinement . . . . .	43
<b>6</b>	<b>Two-variable first-order logics</b>	<b>47</b>
6.1	Type realizability of the two-variable first-order logic . . . . .	47
6.2	Type realizability of the two-variable first-order logic with equivalences in refinement . . . . .	55
6.3	NOTES . . . . .	62



# Glossary

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

## Glossary

- $\text{qr } \varphi$  the quantifier rank of  $\varphi$ . 3  
 $\mathcal{L}_r[\Sigma]$  the  $r$ -rank first-order formulas over  $\Sigma$ . 4  
 $\mathcal{C}_r[\Sigma]$  the  $r$ -rank first-order formulas with counting quantifiers over  $\Sigma$ . 4  
 $\mathcal{L}_r^v[\Sigma]$  the  $r$ -rank  $v$ -variable first-order formulas over  $\Sigma$ . 4  
 $\mathcal{C}_r^v[\Sigma]$  the  $r$ -rank  $v$ -variable first-order formulas with counting quantifiers over  $\Sigma$ . 4  
 $\mathfrak{A}$  a structure. 4  
 $\varphi^{\mathfrak{A}}$  interpretation of  $\varphi$  in  $\mathfrak{A}$ . 5  
 $\text{SAT-}\mathcal{K}$  the satisfiable sentences of  $\mathcal{K}$ . 5  
 $\text{FIN-SAT-}\mathcal{K}$  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  $\Sigma$ . 7  
 $\mathsf{T}[\Sigma]$  the set of 1-types over  $\Sigma$ . 7  
 $\tau^{-1}$  the inverse of the type  $\tau$ . 7  
 $\text{tp}_x \tau$  the  $x$ -type of  $\tau$ . 7  
 $\text{tp}_y \tau$  the  $y$ -type of  $\tau$ . 7  
 $\tau \parallel \tau'$  parallel 2-types. 7  
 $\text{tp}^{\mathfrak{A}}[a]$  the 1-type of  $a$  in  $\mathfrak{A}$ . 7  
 $\pi^{\mathfrak{A}}$  the interpretation of the 1-type  $\pi$  in  $\mathfrak{A}$ . 7  
 $\text{tp}^{\mathfrak{A}}[a, b]$  the 1-type of  $a$  in  $\mathfrak{A}$ . 7  
 $\tau^{\mathfrak{A}}$  the interpretation of the 2-type  $\tau$  in  $\mathfrak{A}$ . 7  
 $\text{PTime}$  complexity class. 8  
 $A \leq_m^{\text{PTime}} B$   $A$  is polynomial-time reducible to  $B$ . 9  
 $A =_m^{\text{PTime}} B$   $A$  and  $B$  are polynomial-time equivalent. 9  
 $\mathsf{B}$  a bit setup. 11  
 $[\mathbf{u}:\text{data}]^{\mathfrak{A}}$   $\mathbf{u}$ -data at  $\mathfrak{A}$ . 11  
 $[\mathbf{u}:\text{eq-}d](x)$   $\mathbf{u}$ -data at  $x$  is  $d$ . 11  
 $[\mathbf{u}:\text{eq}](x, y)$   $\mathbf{u}$ -data equal at  $x$  and  $y$ . 11  
 $[\mathbf{u}:\text{eq-01}](x, y)$   $\mathbf{u}$ -data at  $x$  and  $y$  is 0 and 1. 11  
 $[\mathbf{u}:\text{eq-10}](x, y)$   $\mathbf{u}$ -data at  $x$  and  $y$  is 1 and 0. 11  
 $\mathsf{C}$  a counter setup. 11  
 $[\mathsf{C}:\text{data}]^{\mathfrak{A}}$   $\mathsf{C}$ -data at  $\mathfrak{A}$ . 12  
 $[\mathsf{C}:\text{eq-}d](x)$   $\mathsf{C}$ -data at  $x$  is  $d$ . 12  
 $[\mathsf{C}:\text{eq}](x, y)$   $\mathsf{C}$ -data equal at  $x$  and  $y$ . 12  
 $[\mathsf{C}:\text{less}](x, y)$   $\mathsf{C}$ -data at  $x$  less than  $\mathsf{C}$ -data at  $y$ . 12  
 $[\mathsf{C}:\text{succ}](x, y)$   $\mathsf{C}$ -data at  $y$  succeeds  $\mathsf{C}$ -data at  $x$ . 12  
 $[\mathsf{C}:\text{less}]d(x)$   $\mathsf{C}$ -data at  $x$  less than  $d$ . 13  
 $[\mathsf{C}:\text{betw-}d\text{-}e](x)$   $\mathsf{C}$ -data at  $x$  between  $d$  and  $e$ . 13  
 $[\mathsf{C}:\text{allbetw-}d\text{-}e]$   $\mathsf{C}$ -data between  $d$  and  $e$ . 13  
 $[\mathsf{V}(p):\text{data}]^{\mathfrak{A}}a$  the value of the  $p$ -th counter at  $a$ . 13  
 $[\mathsf{V}:\text{data}]^{\mathfrak{A}}$  the  $\mathsf{V}$ -data at  $a$ . 13  
 $[\mathsf{V}:\text{eq-v}](x)$  the  $\mathsf{V}$ -data at  $x$ . 13  
 $[\mathsf{V}(pq):\text{at-}i\text{-eq}](x)$  equal  $i$ -th bits at  $p$  and  $q$  at  $x$ . 14  
 $[\mathsf{V}(pq):\text{at-}i\text{-eq-01}](x)$  equal  $i$ -th bits at  $p$  and  $q$  are 0 and 1. 14  
 $[\mathsf{V}(pq):\text{at-}i\text{-eq-10}](x)$  equal  $i$ -th bits at  $p$  and  $q$  are 1 and 0. 14  
 $[\mathsf{V}(pq):\text{eq}](x)$  equal  $p$  and  $q$   $\mathsf{V}$ -data at  $x$ . 14  
 $[\mathsf{V}(pq):\text{less}](x)$   $\mathsf{V}$ -data at  $p$  less than at  $q$ . 14  
 $[\mathsf{V}(pq):\text{succ}](x)$   $\mathsf{V}$ -data at  $q$  succeeds the data at  $p$ . 14  
 $[\mathsf{P}:\text{alldiff}]$   $\mathsf{P}$ -data at different positions is different. 15  
 $[\mathsf{P}:\text{perm}]$   $\mathsf{P}$ -data is a permutation. 15  
 $\mathcal{E}E$  the set of equivalence classes of  $E$ . 17  
 $[\mathbf{e}:\text{refl}]$   $\mathbf{e}$  is reflexive. 17  
 $[\mathbf{e}:\text{symm}]$   $\mathbf{e}$  is symmetric. 17  
 $[\mathbf{e}:\text{trans}]$   $\mathbf{e}$  is transitive. 17  
 $[\mathbf{e}:\text{equiv}]$   $\mathbf{e}$  is transitive. 17  
 $[\mathbf{d}, \mathbf{e}:\text{refine}]$  refinement. 18

- $[d, e:\text{global}]$  global agreement. 18  
 $[d, e:\text{local}]$  local agreement. 18  
 $[e_1, e_2, \dots, e_e:\text{refine}]$  symbols in refinement. 19  
 $[e_1, e_2, \dots, e_e:\text{global}]$  symbols in global agreement. 19  
 $[e_1, e_2, \dots, e_e:\text{local}]$  symbols in local agreement. 19  
 $\Lambda_p^v e E_a$  logic notation. 21  
 $[P:\text{alleq}]$  P-data equal everywhere. 22  
 $[P:\text{globperm}]$  P-data is a global permutation. 22  
 $[L:\text{eg-}i](x, y)$  global refinement induced by levels. 23  
 gtr translation of global agreement to refinement. 23  
 $[E:\text{chperm}]^{\mathfrak{A}}$  characteristic E-permutation in  $\mathfrak{A}$ . 25  
 $[L:\text{fixperm}]$  fixed permutation condition. 26  
 $[L:\text{locperm}]$  local agreement condition. 27  
 $[L:\text{el-}i]$  local refinement induced by levels. 27  
 ltr translation of local agreement to refinement. 28  
 $g$  granularity. 29  
 $G$  granularity color setup. 29  
 $[\Gamma:d]$  finer equivalence granularity formula. 30  
 grtr granularity translation. 30  
 $[\Sigma:\text{cell}](x, y)$   $\Sigma$ -cell formula. 33  
 $\mathcal{O}$  organ-equivalence relation. 34  
 $\mathcal{O}$  sub-organ-equivalence relation. 35  
 $[D:\text{Data}]^{\mathfrak{A}}$  D-Data. 39  
 $[D:\text{Zero}](x)$  zero D-Data at  $x$ . 39  
 $[D:\text{Largest}](x)$  maximum D-Data at  $x$ . 40  
 $m_i$  message symbols. 47  
 $\langle \Sigma, \bar{m} \rangle$  classified signature. 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  $\wp^+ A = \wp A \setminus \{\emptyset\}$  is the *set of nonempty subsets* of  $A$ . If  $\kappa$  is a cardinal number, the set  $\wp^\kappa A = \{S \in \wp A \mid |S| = \kappa\}$  is the  $\kappa$ -*powerset* of  $A$ . The *cartesian product* of  $A$  and  $B$  is  $A \times B$ . The sets  $A$  and  $B$  *properly intersect* if  $A \cap B \neq \emptyset$ ,  $A \setminus B \neq \emptyset$  and  $B \setminus A \neq \emptyset$ .

If  $R$  is a binary relation, its *domain* is  $\text{dom } R$  and its *range* is  $\text{ran } R$ . 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

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

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

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

A *function* is formally just a functional relation. The *identity function* on  $A$  is  $\text{id}_A$ . A *total function* from  $A$  to  $B$  is denoted  $f : A \rightarrow B$ . 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  $f(a) \simeq \perp$  means that  $f$  is not defined at  $a$ . If  $S \subseteq A$ , the *characteristic function* of  $S$  in  $A$  is  $\text{ch}_S^A : A \rightarrow \{0, 1\}$ .

A *sequence* is formally just a function with domain an ordinal number. If  $A$  is a sequence, its *length*  $\|A\|$  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  $\varepsilon$ . A *finite sequence* is a sequence of finite length. If  $A$  and  $B$  are two sequences, their *concatenation* is  $A + B$ , and the sequence obtained from  $A$  by dropping all elements of  $B$  is  $A - B$ .

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  $n$

## 1 Introduction

and  $m$  is

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

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

An  $n$ -vector  $\mathbf{v} = (v_1, v_2, \dots, v_n) \in \mathbb{N}^n$  is just a tuple of natural numbers. The  $n$ -vector  $\mathbf{v}$  is *lexicographically smaller*<sup>1</sup> than the  $n$ -vector  $\mathbf{w}$  (written  $\mathbf{v} \prec \mathbf{w}$ ) if there is a position  $p \in [1, n]$  such that  $v_p < w_p$  and  $v_q = w_q$  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  $\nu$  as an  $n$ -vector  $\nu = (\nu(1), \nu(2), \dots, \nu(n))$ .

A function  $f : \mathbb{N} \rightarrow \mathbb{N}$  is *polynomially bounded* if there is a polynomial  $p$  and a number  $n_0 \in \mathbb{N}$  such that  $f(n) \leq p(n)$  for all  $n \geq n_0$ . The function  $f$  is *exponentially bounded* if there is a polynomial  $p$  and a number  $n_0 \in \mathbb{N}$  such that  $f(n) \leq 2^{p(n)}$  for all  $n \geq n_0$ . 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  $\text{exp}_a^e(x)$  by  $\text{exp}_a^0(x) = x$  and  $\text{exp}_a^{e+1}(x) = a^{\text{exp}_a^e(x)}$ , so  $\text{exp}_a^e(x) = a^{a^{\dots^{a^x}}}$  is a tower of  $e$  exponentiations.

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

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  $b = 10$  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  $b \in \mathbb{B}^t$ , the *number encoded by*  $b$  is  $\underline{b}$ . 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

$$\Omega_C = \left\{ \neg, \wedge, \vee, \rightarrow, \leftrightarrow; \exists, \forall; =; (, , , ); \leq, =, \geq, {}^0, {}^1 \right\}.$$

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

<sup>1</sup>the higher positions to the right are more significant; it may *look like* this ordering is the anti-lexicographic one, for example  $(1, 1, 0) \prec (0, 0, 1)$ .

implication  $\rightarrow$  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  $\exists^{\leq \bar{m}}$  or  $\exists^{=\bar{m}}$  or  $\exists^{\geq \bar{m}}$ , where  $m \in \mathbb{N}$  and  $\bar{m} \in \mathbb{B}^+$  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  $\mathcal{V} = \langle \mathbf{v}_1, \mathbf{v}_2, \dots \rangle$  is a countable sequence of distinct *variable symbols*. We pay special attention to  $\mathbf{x} = \mathbf{v}_1$ ,  $\mathbf{y} = \mathbf{v}_2$  and  $\mathbf{z} = \mathbf{v}_3$ , the *first*, *second* and *third* variable symbol, respectively.

A *predicate signature*  $\Sigma = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_s \rangle$  is a finite sequence of distinct *predicate symbols*  $\mathbf{p}_i$  together with their *arities*  $\text{ar } \mathbf{p}_i \in \mathbb{N}^+$ . 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  $\Sigma$  are disjoint.

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

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

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

The set of *literals*  $\mathcal{Lit}[\Sigma] \subset (\Omega_{\mathcal{C}} \cup \mathcal{V} \cup \Sigma)^*$  over  $\Sigma$  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  $\Sigma$  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  $\Sigma$  consists of the formulas that do not feature a counting quantifier.

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

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

## 1 Introduction

- $(x = y)$ , then  $\text{qr } \varphi = 0$
- $p(x_1, x_2, \dots, x_n)$ , then  $\text{qr } \varphi = 0$
- $(\neg\psi)$ , then  $\text{qr } \varphi = \text{qr } \psi$
- $\psi_1 \oplus \psi_2$  for  $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ , then  $\text{qr } \varphi = \max(\text{qr } \psi_1, \text{qr } \psi_2)$
- $(\exists x\psi)$  or  $(\forall x\psi)$ , then  $\text{qr } \varphi = 1 + \text{qr } \psi$
- $(\exists^{\leq m}x\psi)$  or  $(\exists^=m x\psi)$ , then  $\text{qr } \varphi = m + 1 + \text{qr } \psi$
- $(\exists^{\geq m}x\psi)$ , then  $\text{qr } \varphi = m + \text{qr } \psi$ .

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

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

We will omit unnecessary brackets in formulas.

## 1.3 Semantics

If  $\Sigma$  is a predicate signature, a  $\Sigma$ -structure  $\mathfrak{A}$  consists of a nonempty set  $A$  (the *domain* of  $\mathfrak{A}$ ), together with a relation  $p^{\mathfrak{A}} \subseteq A^{\text{ar } p}$  (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  $\varphi$ ”;  $\exists^=m x\varphi$  means that “exactly  $m$  elements satisfy  $\varphi$ ”;  $\exists^{\geq m}x\varphi$  means that “at least  $m$  elements satisfy  $\varphi$ ”.

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

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

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

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

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

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

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

A predicate signature with intended interpretations  $\Sigma$  is formally a predicate signature together with an *intended interpretation condition*  $\mathcal{A}$ , which is formally a class of  $\Sigma$ -structures. A  $\Sigma$ -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  $\Sigma = \langle e \rangle$  is a predicate signature consisting of the single binary predicate symbol  $e$ , having intended interpretation as an equivalence, then the  $\Sigma$ -formula  $\forall x e(x, x)$  is logically valid. From now on, we will use the term *predicate signature* as *predicate signature with possible intended interpretations*.

The predicate signature  $\Sigma'$  is an *enrichment* of the predicate signature  $\Sigma$  if  $\Sigma'$  contains all predicate symbols of  $\Sigma$  and respects their intended interpretation in  $\Sigma$ . A  $\Sigma'$ -structure  $\mathfrak{A}'$  is an enrichment of the  $\Sigma$ -structure  $\mathfrak{A}$  if they have the same domain and the same interpretation of the predicate symbols of  $\Sigma$ . The basic semantic significance of enrichment is that if  $\varphi(x_1, x_2, \dots, x_n)$  is a  $\Sigma$ -formula and  $a_1, a_2, \dots, a_n \in A$ , then  $\mathfrak{A} \models \varphi(a_1, a_2, \dots, a_n)$  iff  $\mathfrak{A}' \models \varphi(a_1, a_2, \dots, a_n)$ . If  $\mathfrak{A}'$  is an enrichment of  $\mathfrak{A}$  then  $\mathfrak{A}$  is a *reduct*<sup>2</sup> of  $\mathfrak{A}'$ .

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

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

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

<sup>2</sup>or why not *empoverishment*?

## 1 Introduction

is  $\text{FIN-SAT-}\mathcal{K} \subseteq \mathcal{K}$ . The family  $\mathcal{K}$  has the *finite model property* if  $\text{SAT-}\mathcal{K} = \text{FIN-SAT-}\mathcal{K}$ . By the Löwenheim-Skolem theorem, every satisfiable sentence  $\varphi$  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  $\Sigma$ -sentences  $\varphi$  and  $\psi$  are *logically equivalent* (written  $\varphi \equiv \psi$ ) if they have the same models.

Two  $\Sigma$ -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  $\mathfrak{A} \equiv_r^v \mathfrak{B}$ ) if they satisfy the same  $r$ -rank  $v$ -variable first-order sentences.

## 1.4 Games

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

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 a structure and places it on some element in that 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  $\bar{a} = \langle a_1, a_2, \dots, a_r \rangle$  and  $\bar{b} = \langle b_1, b_2, \dots, b_r \rangle$ . Duplicator wins the match if the relation  $\bar{a} \mapsto \bar{b} = \{a_1 \mapsto b_1, a_2 \mapsto b_2, \dots, a_r \mapsto b_r\} \subseteq A \times B$ , 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  $(\mathfrak{I}_0, \mathfrak{I}_1, \dots, \mathfrak{I}_r)$  is a sequence of nonempty sets of partial isomorphisms between  $\mathfrak{A}$  and  $\mathfrak{B}$  with the following properties:*

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

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

## 1.5 Types

Let  $\Sigma = \langle p_1, p_2, \dots, p_s \rangle$  be a predicate signature. A 1-type  $\pi$  over  $\Sigma$  is a maximal consistent set of literals featuring only the variable symbol  $\mathbf{x}$ <sup>3</sup>. The set of 1-types over  $\Sigma$  is  $\Pi[\Sigma]$ . Note that consistency here is relativised by the intended interpretations of the predicate signature. For example if  $\Sigma$  contains the binary predicate symbol  $e$  with intended interpretation as an equivalence, then every 1-type over  $\Sigma$  includes the literal  $e(\mathbf{x}, \mathbf{x})$ . Also note that the cardinality of a 1-type over  $\Sigma$  is polynomially bounded by the length  $s$  of  $\Sigma$  and the cardinality of  $\Pi[\Sigma]$  is exponentially bounded by  $s$ .

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

If  $\tau \in T[\Sigma]$ , the *inverse*  $\tau^{-1}$  of  $\tau$  is the 2-type obtained from  $\tau$  by swapping the variables  $\mathbf{x}$  and  $\mathbf{y}$  in every literal. The  $\mathbf{x}$ -type of  $\tau$  is the 1-type  $\text{tp}_{\mathbf{x}} \tau$  consisting of all the literals of  $\tau$  featuring only the variable symbol  $\mathbf{x}$ . Similarly, the  $\mathbf{y}$ -type of  $\tau$  is the 1-type  $\text{tp}_{\mathbf{y}} \tau$  consisting of all the literals of  $\tau$  featuring only the variable symbol  $\mathbf{y}$ , that is replaced by  $\mathbf{x}$ . For instance we have the identity  $\text{tp}_{\mathbf{x}} \tau^{-1} = \text{tp}_{\mathbf{y}} \tau$ . We say that  $\tau$  *connects* the 1-types  $\text{tp}_{\mathbf{x}} \tau$  and  $\text{tp}_{\mathbf{y}} \tau$  and we refer to  $\text{tp}_{\mathbf{x}} \tau$  and  $\text{tp}_{\mathbf{y}} \tau$  as the *endpoints* of  $\tau$ . Two 2-types  $\tau, \tau'$  are *parallel* if they have the same endpoints, that is if  $\text{tp}_{\mathbf{x}} \tau = \text{tp}_{\mathbf{x}} \tau'$  and  $\text{tp}_{\mathbf{y}} \tau = \text{tp}_{\mathbf{y}} \tau'$ . We use the notation  $\tau \parallel \tau'$  to denote parallel 2-types.

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

$$\text{tp}^{\mathfrak{A}}[a] = \{\lambda(\mathbf{x}) \in \text{Lit}[\Sigma] \mid \mathfrak{A} \models \lambda(a)\}.$$

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

$$\pi^{\mathfrak{A}} = \{a \in A \mid \text{tp}^{\mathfrak{A}}[a] = \pi\}.$$

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

$$\text{tp}^{\mathfrak{A}}[a, b] = \{\lambda(\mathbf{x}, \mathbf{y}) \in \text{Lit}[\Sigma] \mid \mathfrak{A} \models \lambda(a, b)\}.$$

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

$$\tau^{\mathfrak{A}} = \{(a, b) \in A \times A \mid a \neq b \wedge \text{tp}^{\mathfrak{A}}[a, b] = \tau\}.$$

---

<sup>3</sup>this is different than the commonly used notion of type in model theory, where types are sets of general formulas, not just literals

## 1.6 Normal forms

In two-variable logics, a common technique of reducing formula quantifier rank while preserving satisfiability is Skolemization [2]: Let  $\varphi$  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  $\varphi$ . Consider a subformula  $\psi$  of  $\varphi$  that has the lowest possible nontrivial quantifier rank 1. Then  $\psi = \psi(y) = \exists x\alpha(x, y)$ , where the formula  $\alpha$  is quantifier-free,  $\{x, y\} = \{\mathbf{x}, \mathbf{y}\}$  and  $y$  may or may not necessarily occur freely in  $\alpha$ . Introduce a new unary predicate symbol  $\mathbf{u}_\psi$  with the intended interpretation  $\forall y(\mathbf{u}_\psi(y) \leftrightarrow \exists x\alpha(x, y))$  and let  $\varphi'$  be the formula obtained from  $\varphi$  by replacing the subformula  $\psi$  by  $\mathbf{u}_\psi(y)$ . The original formula  $\varphi$  is equisatisfiable with  $\varphi_1 = \forall y(\mathbf{u}_\psi(y) \leftrightarrow \exists x\alpha(x, y)) \wedge \varphi'$  in a strict sense, that is any model for  $\varphi$  can be  $\mathbf{u}_\psi$ -enriched into a model for  $\varphi_1$  and any model for  $\varphi_1$  is a model for  $\varphi$ . 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  $\text{sctr} : \mathcal{L}^2 \rightarrow \mathcal{L}^2$  which reduces every sentence  $\varphi$  to a sentence  $\text{sctr } \varphi$  in Scott normal form:*

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

where the formulas  $\alpha_i$  are quantifier-free and use at most linearly many new unary predicate symbols. The sentences  $\varphi$  and  $\text{sctr } \varphi$  are satisfiable over the same domains. Moreover the length  $\text{sctr } \varphi$  is linear in the length of  $\varphi$ .

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  $\text{prtr} : \mathcal{C}^2 \rightarrow \mathcal{C}^2$  with reduces every sentence  $\varphi$  to a sentence  $\text{prtr } \varphi$  in the form:*

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

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

## 1.7 Complexity

We denote the complexity classes  $\mathbf{PTIME} = \text{TIME}[\text{poly}(n)] = \bigcup_{c \in \mathbb{N}^+} \text{TIME}[n^c]$ ,  $\mathbf{NPTIME}$ ,  $\mathbf{PSpace}$ ,  $\mathbf{EXPTIME}$  and  $\mathbf{NEXPTIME}$ . For  $e \in \mathbb{N}^+$ , the  $e$ -exponential deterministic and nondeterministic time classes are  $e\mathbf{EXPTIME} = \text{TIME}[\exp_e^e(\text{poly}(n))]$  and  $\mathbf{NeEXPTIME}$ . The complexity class  $\mathbf{ELEMENTARY}$  is the union of the complexity classes  $e\mathbf{EXPTIME}$  for  $e \in \mathbb{N}^+$ .



The Grzegorzcyk 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  $\text{zero}(n) = 0$ , the successor function  $\text{succ}(n) = n + 1$  and the projection functions  $\text{proj}_i^u(n_1, n_2, \dots, n_u) = n_i$ . If  $u, v \in \mathbb{N}$ ,  $f : \mathbb{N}^u \rightarrow \mathbb{N}$  and  $g_1, g_2, \dots, g_u : \mathbb{N}^v \rightarrow \mathbb{N}$  are functions, their *superposition* is the function  $h : \mathbb{N}^v \rightarrow \mathbb{N}$  defined by  $h(\bar{n}) = f(g_1(\bar{n}), g_2(\bar{n}), \dots, g_u(\bar{n}))$  for  $\bar{n} \in \mathbb{N}^v$ . If  $u \in \mathbb{N}$ ,  $f : \mathbb{N}^u \rightarrow \mathbb{N}$  and  $g : \mathbb{N}^{u+2} \rightarrow \mathbb{N}$ , their *primitive recursion* is the function  $h : \mathbb{N}^{u+1} \rightarrow \mathbb{N}$  defined by:

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

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

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

For  $i \in \mathbb{N}$ , the  $i$ -th level of the Grzegorzcyk 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  $h : \mathbb{N}^{u+1} \rightarrow \mathbb{N}$  of the functions  $f : \mathbb{N}^u \rightarrow \mathbb{N}$ ,  $g : \mathbb{N}^{u+2} \rightarrow \mathbb{N}$ ,  $f, g \in \mathcal{E}^i$ , such that there is a function  $b : \mathbb{N}^{u+1} \rightarrow \mathbb{N}$ ,  $b \in \mathcal{E}^i$  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 Grzegorzcyk hierarchy just in case its characteristic function occurs at that level. The primitive recursive functions are partitioned by the Grzegorzcyk hierarchy. The complexity class ELEMENTARY coincides with the third level of the Grzegorzcyk 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  $A \leq_m^{\text{PTIME}} B$ ) if there is a polynomial-time algorithm  $f : \Omega_1^* \rightarrow \Omega_2^*$  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  $A =_m^{\text{PTIME}} B$ ) if  $A \leq_m^{\text{PTIME}} B$  and  $B \leq_m^{\text{PTIME}} A$ .

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*  $c^0 = \langle t_1^0, t_2^0, \dots, t_n^0 \rangle$ , where  $n \leq m$ , is a mapping  $t : [1, m] \times [1, m] \rightarrow T$  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 domino system  $D_0$  for which:

## 1 Introduction

- 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*  $\mathbf{B} = \langle \mathbf{u} \rangle$  is a predicate signature consisting of a single unary predicate symbol  $\mathbf{u}$ .

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

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

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

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

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

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

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

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

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

### 2.2 Counters

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

## 2 Counter setups

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

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

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

$$[\text{C:eq-}d](\mathbf{x}) = \bigwedge_{1 \leq i \leq t} [\mathbf{u}_i:\text{eq-}\bar{d}_i](\mathbf{x}).$$

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

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

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

$$[\text{C:eq}](\mathbf{x}, \mathbf{y}) = \bigwedge_{1 \leq i \leq t} [\mathbf{u}_i:\text{eq}](\mathbf{x}, \mathbf{y}).$$

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

The bitstring  $a \in \mathbb{B}^t$  encodes a number less than the number encoded by the bitstring  $b \in \mathbb{B}^t$ , 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 two conditions hold:

$$a_j = 0 \text{ and } b_j = 1 \quad (\text{Less1})$$

$$a_k = b_k \text{ for all } k \in [j+1, t]. \quad (\text{Less2})$$

**Definition 7.** Define the quantifier-free  $\mathcal{L}^2[\text{C}]$ -formula  $[\text{C:less}](\mathbf{x}, \mathbf{y})$  by:

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

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

The bitstring  $b \in \mathbb{B}^t$  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:

$$a_j = 0 \text{ and } b_j = 1 \quad (\text{Succ1})$$

$$a_i = 1 \text{ for all } i \in [1, j-1] \quad (\text{Succ2})$$

$$b_i = 0 \text{ for all } i \in [1, j-1] \quad (\text{Succ3})$$

$$a_k = b_k \text{ for all } k \in [j+1, t]. \quad (\text{Succ4})$$

**Definition 8.** Define the quantifier-free  $\mathcal{L}^2[\text{C}]$ -formula  $[\text{C:succ}](\mathbf{x}, \mathbf{y})$  by:

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

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

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

**Definition 9.** Let  $d \in \mathbb{B}_t$ . Define the quantifier-free  $\mathcal{L}^1[\text{C}]$ -formula  $[\text{C:less}]d(\mathbf{x})$  by:

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

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

**Definition 10.** Let  $d \leq e \in \mathbb{B}_t$ . Define the quantifier-free  $\mathcal{L}^1[\text{C}]$ -formula  $[\text{C:betw-}d-e](\mathbf{x})$  by:

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

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

$$\mathfrak{A} \models [\text{C:betw-}d-e](a) \text{ iff } d \leq [\text{C:data}]^{\mathfrak{A}}a \leq e.$$

**Definition 11.** Let  $d \leq e \in \mathbb{B}_t$ . Define the  $\mathcal{L}^1[\text{C}]$ -sentence  $[\text{C:allbetw-}d-e]$  by:

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

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

## 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  $V = \langle \mathbf{u}_{11}, \mathbf{u}_{12}, \dots, \mathbf{u}_{nt} \rangle$  of  $(nt)$  distinct unary predicate symbols. The counter setup  $V(p)$  of  $V$  at position  $p \in [1, n]$  is  $V(p) = \langle \mathbf{u}_{p1}, \mathbf{u}_{p2}, \dots, \mathbf{u}_{pt} \rangle$ .

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

$$[\text{V:data}]^{\mathfrak{A}}a = ([\text{V}(1):\text{data}]^{\mathfrak{A}}a, [\text{V}(2):\text{data}]^{\mathfrak{A}}a, \dots, [\text{V}(n):\text{data}]^{\mathfrak{A}}a).$$

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

$$[\text{V:eq-v}](\mathbf{x}) = \bigwedge_{1 \leq p \leq n} [\text{V}(p):\text{eq-}d_p](\mathbf{x}).$$

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

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

## 2 Counter setups

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 2.4 Permutations

Let  $n \in \mathbb{N}^+$ . An  $n$ -permutation setup  $P = \langle \mathbf{u}_{11}, \mathbf{u}_{12}, \dots, \mathbf{u}_{nt} \rangle$  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 all permutations of  $[1, n]$  is a subset of  $\mathbb{B}_t^n$ .

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

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

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

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

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

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

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





### 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  $\mathcal{E}E = \{E[a] \mid a \in A\}$ .

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

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

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

$$[e:\text{symm}] = \forall x \forall y (e(x, y) \rightarrow e(y, x)).$$

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

$$[e:\text{trans}] = \forall x \forall y \forall z (e(x, y) \wedge e(y, z) \rightarrow e(x, z)).$$

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

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

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

#### 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,

$$(\forall a \in A)(\forall b \in A) (D(a, b) \rightarrow E(a, b)).$$

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,

$$(\forall a \in A) ((\forall b \in A) (D(a, b) \rightarrow E(a, b)) \vee (\forall b \in A) (E(a, b) \rightarrow D(a, b))).$$

### 3 Equivalence relations

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

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

$$[\mathbf{d}, \mathbf{e}:\text{refine}] = \forall x \forall y (\mathbf{d}(x, y) \rightarrow \mathbf{e}(x, y)).$$

Then  $\langle D, E \rangle$  is in refinement iff  $\mathfrak{A} \models [\mathbf{d}, \mathbf{e}:\text{refine}]$ .

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

$$[\mathbf{d}, \mathbf{e}:\text{global}] = [\mathbf{d}, \mathbf{e}:\text{refine}] \vee [\mathbf{e}, \mathbf{d}:\text{refine}].$$

Then  $\langle D, E \rangle$  is in global agreement iff  $\mathfrak{A} \models [\mathbf{d}, \mathbf{e}:\text{global}]$ .

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

$$[\mathbf{d}, \mathbf{e}:\text{local}] = \forall x (\forall y (\mathbf{d}(x, y) \rightarrow \mathbf{e}(x, y)) \vee \forall y (\mathbf{e}(x, y) \rightarrow \mathbf{d}(x, y))).$$

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

**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  $D[b] \not\subseteq E[b]$  and  $E[b] \not\subseteq D[b]$ . 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  $\langle E_1, E_2, \dots, E_e \rangle$  be a sequence of equivalence relations on  $A$ .

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

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  $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \dots \subseteq E_{\nu(e)}$ .

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

### 3.2 Many equivalence relations in agreement

Let  $E = \langle e_1, e_2, \dots, e_e \rangle$  be a predicate signature consisting of  $e$  binary predicate symbols. Let  $\mathfrak{A}$  be an  $E$ -structure and suppose that the symbols  $e_i$  are interpreted as equivalence relations on  $A$ . Let  $E_i = e_i^{\mathfrak{A}}$  for  $i \in [1, e]$ .

**Definition 25.** Define the  $\mathcal{L}^2[E]$ -sentence  $[e_1, e_2, \dots, e_e:\text{refine}]$  by:

$$[e_1, e_2, \dots, e_e:\text{refine}] = \forall \mathbf{x} \forall \mathbf{y} \bigwedge_{1 \leq i < e} (e_i(\mathbf{x}, \mathbf{y}) \rightarrow e_{i+1}(\mathbf{x}, \mathbf{y})).$$

Then  $\langle E_1, E_2, \dots, E_e \rangle$  is in refinement iff  $\mathfrak{A} \models [e_1, e_2, \dots, e_e:\text{refine}]$ .

**Definition 26.** Define the  $\mathcal{L}^2[E]$ -sentence  $[e_1, e_2, \dots, e_e:\text{global}]$  by:

$$[e_1, e_2, \dots, e_e:\text{global}] = \bigvee_{\nu \in \mathbb{S}_e} [e_{\nu(1)}, e_{\nu(2)}, \dots, e_{\nu(e)}:\text{refine}].$$

Then  $\langle E_1, E_2, \dots, E_e \rangle$  is in global agreement iff  $\mathfrak{A} \models [e_1, e_2, \dots, e_e:\text{global}]$ . Note that the length of the formula  $[e_1, e_2, \dots, e_e:\text{global}]$  grows exponentially as  $e$  grows.

**Definition 27.** Define the  $\mathcal{L}^2[E]$ -sentence  $[e_1, e_2, \dots, e_e:\text{local}]$  by:

$$[e_1, e_2, \dots, e_e:\text{local}] = \forall \mathbf{x} \bigvee_{\nu \in \mathbb{S}_e} \forall \mathbf{y} \bigwedge_{1 \leq i < e} (e_{\nu(i)}(\mathbf{x}, \mathbf{y}) \rightarrow e_{\nu(i+1)}(\mathbf{x}, \mathbf{y})).$$

Then  $\langle E_1, E_2, \dots, E_e \rangle$  is in local agreement iff  $\mathfrak{A} \models [e_1, e_2, \dots, e_e:\text{local}]$ . Note that the length of the formula  $[e_1, e_2, \dots, e_e:\text{local}]$  grows exponentially as  $e$  grows.

Let  $E = \langle E_1, E_2, \dots, E_e \rangle$  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  $S = \{E_{i(1)}, E_{i(2)}, \dots, E_{i(s)}\}$ , where  $1 \leq i(1) < i(2) < \dots < i(s) \leq e$ , 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,  $D = \cup\{E_{i(1)}, E_{i(2)}, \dots, E_{i(s-1)}\}$  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  $D[a] = E_{i(1)}[a] \cup E_{i(2)}[a] \cup \dots \cup E_{i(s-1)}[a]$  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  $E_i[a] \not\subseteq E_j[a]$  and  $E_j[a] \not\subseteq E_i[a]$ . 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  $E = \langle E_1, E_2, \dots, E_e \rangle$  is in local agreement.

### 3 Equivalence relations

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

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

That is,  $(E \upharpoonright I)$  just collects the equivalences having indices from  $I$ .

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

$$L_k = \cap \left\{ \cup(E \upharpoonright I) \mid I \in \wp^k[1, e] \right\}.$$

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

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

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

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

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

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

**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  $K = \{\nu(1), \nu(2), \dots, \nu(k)\}$ . By the definition of  $L_k$ , followed by eq. (3.1), we have  $L_k[a] \subseteq \cup(E \upharpoonright K)[a] = E_{\nu(k)}[a] = E_i[a]$ . Next, let  $K \subseteq \wp^k[1, e]$  be any  $k$ -index set. By the pigeonhole principle, there is some  $k' \geq k$  such that  $k' \in K$ . By eq. (3.1) we have:

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

Hence  $E_i[a] \subseteq \cap \left\{ \cup(E \upharpoonright K)[a] \mid K \in \wp^k[1, e] \right\} = L_k[a]$ .  $\square$

## 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

$$\Lambda_p^v e E_a$$

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
- $a \in \{\text{refine}, \text{global}, \text{local}\}$ , 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.  $\mathcal{L}_0 1E$  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.  $\mathcal{L}^2 2E$  is the two-variable logic, featuring unary, binary predicate symbols and two builtin equivalence symbols.  $\mathcal{C}_1^2 2E_{\text{local}}$  is the two-variable logic with counting quantifiers, featuring unary predicate symbols and two builtin equivalence symbols in local agreement.  $\mathcal{L}_1 E_{\text{global}}$  is the monadic first-order logic featuring many equivalence symbols in global agreement.

When we working with a concrete logic, for example  $\mathcal{C}^2 2E_{\text{local}}$ , we implicitly assume an appropriate generic predicate signature  $\Sigma$  for it. In this case, there are two builtin equivalence symbols  $d$  and  $e$  in  $\Sigma$  and in addition  $\Sigma$  contains arbitrary many unary

## 4 Reductions

and binary predicate symbols. The *intended interpretation* of the builtin equivalence symbols is fixed by an appropriate condition  $\theta$ . In this case:

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

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  $[d:\text{equiv}]$  uses the variables  $x, y$  and  $z$  and the logic  $\mathcal{C}^2\mathcal{E}_{\text{local}}$  is a two-variable logic. Recall that when talking about semantics, we include the intended interpretation condition in the definition of  $\Sigma$ -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  $\Sigma$  as a predicate signature for the logics  $\Lambda\mathcal{E}_{\text{global}}$  and  $\Lambda\mathcal{E}_{\text{refine}}$ . The  $e$  builtin equivalence symbols of  $\Sigma$  are  $e_1, e_2, \dots, e_e$ .

Let  $\varphi$  be a  $\Lambda[\Sigma]$ -sentence. The class of  $\Lambda\mathcal{E}_{\text{refine}}$ -structures satisfying  $\varphi$  coincides with the class of  $\Lambda\mathcal{E}_{\text{global}}$ -structures satisfying:

$$\varphi \wedge [e_1, e_2, \dots, e_e:\text{refine}].$$

Hence:

$$(\text{FIN-})\text{SAT-}\Lambda\mathcal{E}_{\text{refine}} \leq_m^{\text{PTIME}} (\text{FIN-})\text{SAT-}\Lambda\mathcal{E}_{\text{global}}.$$

Since the length of the formula  $[e_1, e_2, \dots, e_e:\text{refine}]$  grows polynomially as  $e$  grows:

$$(\text{FIN-})\text{SAT-}\Lambda\mathcal{E}_{\text{refine}} \leq_m^{\text{PTIME}} (\text{FIN-})\text{SAT-}\Lambda\mathcal{E}_{\text{global}}.$$

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

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

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

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

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

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

#### 4.1 Global agreement to refinement

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

Let  $L = \langle l_1, l_2, \dots, l_e \rangle + P$  be a predicate signature consisting of the binary predicate symbols  $l_k$  in addition to the symbols from  $P$ .

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

$$[\text{L:eg-}i](x, y) = \bigwedge_{1 \leq k \leq e} ([P(k):\text{eq-}i](x) \rightarrow l_k(x, y)).$$

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

$$[\text{L:eg-}i]^{\mathfrak{A}} = l_{\nu^{-1}(i)}^{\mathfrak{A}}.$$

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

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

$$\mathfrak{A} \models [P(k'):\text{eq-}i](a) \text{ iff } [P(k'):\text{data}]^{\mathfrak{A}}a = i \text{ iff } k' = k. \quad (4.1)$$

Let  $a, b \in A$ . First suppose that  $\mathfrak{A} \models [\text{L:eg-}i](a, b)$ . By eq. (4.1) we must have that  $\mathfrak{A} \models [P(k):\text{eq-}i](a)$ , hence  $\mathfrak{A} \models l_k(a, b)$ .

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

$$\mathfrak{A} \models \neg([P(k'):\text{eq-}i](a) \rightarrow l_{k'}(a, b)) \equiv [P(k'):\text{eq-}i](a) \wedge \neg l_{k'}(a, b).$$

By eq. (4.1) we have  $k' = k$ , hence  $\mathfrak{A} \models \neg l_k(a, b)$ . □

Let  $E = \langle e_1, e_2, \dots, e_e \rangle$  be a predicate signature consisting of the binary predicate symbols  $e_i$ . Let  $\Sigma$  be a predicate signature enriching  $E$  and not containing any symbols from  $L$ . Let  $\Sigma' = \Sigma \cup L$  and  $L' = \Sigma' - E$ .

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

$$\text{gtr } \varphi = \varphi' \wedge [\text{P:globperm}],$$

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

#### 4 Reductions

**Remark 5.** Let  $\varphi$  be a  $\Lambda[\Sigma]$ -formula and let  $\mathfrak{A}$  be a  $\Sigma$ -structure. Suppose that  $\mathfrak{A} \models \varphi$  and that the symbols  $e_1, e_2, \dots, e_e$  are interpreted in  $\mathfrak{A}$  as equivalence relations on  $A$  in global agreement. Then there is a  $\Sigma'$ -enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  such that  $\mathfrak{A}' \models \text{gtr } \varphi$  and that the symbols  $l_1, l_2, \dots, l_e$  are interpreted in  $\mathfrak{A}'$  as equivalence relations on  $A$  in refinement.

*Proof.* There is a permutation  $\nu \in \mathbb{S}_e$  such that  $e_{\nu(1)}^{\mathfrak{A}} \subseteq e_{\nu(2)}^{\mathfrak{A}} \subseteq \dots \subseteq e_{\nu(e)}^{\mathfrak{A}}$ . Consider an enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  to a  $\Sigma'$ -structure where  $l_k^{\mathfrak{A}'} = e_{\nu(k)}^{\mathfrak{A}}$ , so the interpretations of  $l_k$  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  $\mathfrak{A}' \models [P:\text{globperm}]$  and  $[P:\text{data}]^{\mathfrak{A}'}a = \nu$  for all  $a \in A$ . By Remark 4, for every  $i \in [1, e]$ :

$$[L:\text{eg-}i]^{\mathfrak{A}'} = l_{\nu^{-1}(i)}^{\mathfrak{A}'} = e_{\nu(\nu^{-1}(i))}^{\mathfrak{A}'} = e_i^{\mathfrak{A}'} = e_i^{\mathfrak{A}}.$$

Hence  $\mathfrak{A}' \models \forall x \forall y (e_i(x, y) \leftrightarrow [L:\text{eg-}i](x, y))$ . Since  $\mathfrak{A}' \models \varphi$  we have  $\mathfrak{A}' \models \text{gtr } \varphi$ .  $\square$

**Remark 6.** Let  $\varphi$  be a  $\Lambda[\Sigma]$ -formula and let  $\mathfrak{A}$  be an  $L'$ -structure. Suppose that  $\mathfrak{A} \models \text{gtr } \varphi$  and that the symbols  $l_1, l_2, \dots, l_e$  are interpreted in  $\mathfrak{A}$  as equivalence relations on  $A$  in refinement. Then there is a  $\Sigma'$ -enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  such that  $\mathfrak{A}' \models \varphi$  and that the symbols  $e_1, e_2, \dots, e_e$  are interpreted as equivalence relations on  $A$  in global agreement in  $\mathfrak{A}'$ .

*Proof.* Consider an enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  to a  $\Sigma'$ -structure where  $e_i^{\mathfrak{A}'} = [L:\text{eg-}i]^{\mathfrak{A}'}$ . By Remark 4,  $\langle e_1^{\mathfrak{A}'}, e_2^{\mathfrak{A}'}, \dots, e_e^{\mathfrak{A}'} \rangle$  is a sequence of equivalence relations on  $A$  in global agreement. For every  $i \in [1, e]$  we have  $\mathfrak{A}' \models \forall x \forall y (e_i(x, y) \leftrightarrow [L:\text{eg-}i](x, y))$  by definition. Since  $\mathfrak{A}' \models \text{gtr } \varphi$  we have  $\mathfrak{A}' \models \varphi$ .  $\square$

The last two remarks show that a  $\Lambda eE_{\text{global}}$ -formula  $\varphi$  has essentially the same models as the  $\Lambda eE_{\text{refine}}$ -formula  $\text{gtr } \varphi$ , so we have shown:

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

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

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

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  $\Lambda_0^2 eE_a$ ,  $\Lambda_1^2 eE_a$  and  $\Lambda_1^2 E_a$  for  $a \in \{\text{global}, \text{refine}\}$  (but not for the fragment  $\Lambda_0^2 E_a$ ).



## 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  $\Sigma$  as a predicate signature for the logics  $\Lambda eE_{\text{local}}$  and  $\Lambda eE_{\text{refine}}$ . The  $e$  builtin equivalence symbols of  $\Sigma$  are  $e_1, e_2, \dots, e_e$ .

Let  $\varphi$  be a  $\Lambda[\Sigma]$ -sentence. The class of  $\Lambda eE_{\text{refine}}$ -structures satisfying  $\varphi$  coincides with the class of  $\Lambda eE_{\text{local}}$ -structures satisfying

$$\varphi \wedge [e_1, e_2, \dots, e_e : \text{refine}].$$

Hence:

$$(\text{FIN-})\text{SAT-}\Lambda eE_{\text{refine}} \leq_m^{\text{PTIME}} (\text{FIN-})\text{SAT-}\Lambda eE_{\text{local}}.$$

Since the size of the formula  $[e_1, e_2, \dots, e_e : \text{refine}]$  grows polynomially as  $e$  grows, we have:

$$(\text{FIN-})\text{SAT-}\Lambda E_{\text{refine}} \leq_m^{\text{PTIME}} (\text{FIN-})\text{SAT-}\Lambda E_{\text{local}}.$$

Consider the opposite direction. Let  $E = \langle e_1, e_2, \dots, e_e \rangle$  be a predicate signature consisting of the binary predicate symbols  $e_i$  (later, we will need these to be not necessarily interpreted as equivalences, but for now we will interpret them as such). Let  $\mathfrak{A}$  be an  $E$ -structure and suppose that the symbols  $e_i$  are interpreted in  $\mathfrak{A}$  as equivalence relations on  $A$  in local agreement. Let  $E_i = e_i^{\mathfrak{A}}$  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):

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

**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  $[\text{E:chperm}]^{\mathfrak{A}} : A \rightarrow \mathbb{S}_e$  so that  $[\text{E:chperm}]^{\mathfrak{A}} a$  is the characteristic  $E$ -permutation of  $a$  in  $\mathfrak{A}$ .

**Remark 7.** Let  $a \in A$ ,  $\nu = [\text{E:chperm}]^{\mathfrak{A}} a$  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  $\nu$  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,  $\mu$  is a permutation satisfying eq. (4.2) that is lexicographically smaller than  $\nu$  — a contradiction.  $\square$

#### 4 Reductions

**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  $\alpha$  and  $q > p$  is the position of  $i$  in the permutation  $\beta$ . By the pigeonhole principle, there is  $k \in [1, e]$  that occurs after  $i$  in  $\alpha$  and before  $j$  in  $\beta$ :  $p < \alpha^{-1}(k)$  and  $\beta^{-1}(k) < q$ . Since  $\beta$  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  $\alpha$ , 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  $\beta$ , 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.  $\square$

Let  $L = \langle L_1, L_2, \dots, L_e \rangle$  be the levels of  $E = \langle E_1, E_2, \dots, E_e \rangle$ . 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  $\alpha$  satisfies eq. (4.2), by Lemma 2:

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

$\square$

**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  $\beta(k) = \alpha(k)$ .  $\square$

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

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

$$[\text{L:fixperm}] = \forall \mathbf{x} \forall \mathbf{y} \bigwedge_{1 \leq k \leq e} (\mathbf{l}_k(\mathbf{x}, \mathbf{y}) \rightarrow [P(k):\text{eq}](\mathbf{x}, \mathbf{y})).$$

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

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

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

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

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

$$[\mathbf{L}:\text{el-}i](\mathbf{x}, \mathbf{y}) = \bigwedge_{1 \leq k \leq n} ([\mathbf{P}(k):\text{eq-}i](\mathbf{x}) \rightarrow \mathbf{l}_k(\mathbf{x}, \mathbf{y})).$$

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

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

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

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

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

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

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

*Proof.* Let  $E_i = [\mathbf{L}:\text{el-}i]^{\mathfrak{A}}$  and  $L_i = \mathbf{l}_i^{\mathfrak{A}}$  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  $\mathfrak{A} \models \mathbf{l}_k(a, b)$  and by Remark 11,  $i = \nu(a)(k) = \nu(b)(k)$ . By Remark 12:

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

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

#### 4 Reductions

- 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  $E_1, E_2, \dots, E_e$  are in local agreement.  $\square$

Let  $E = \langle e_1, e_2, \dots, e_e \rangle$  be a predicate signature consisting of binary predicate symbols. Let  $\Sigma$  be a predicate signature enriching  $E$  and not containing any symbols from  $L$ . Let  $\Sigma' = \Sigma + L$  and  $L' = \Sigma' - E$ .

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

$$\text{ltr } \varphi = \varphi' \wedge [L:\text{locperm}],$$

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

**Remark 14.** Let  $\varphi$  be a  $\Lambda[\Sigma]$ -formula and let  $\mathfrak{A}$  be a  $\Sigma$ -structure. Suppose that  $\mathfrak{A} \models \varphi$  and that the symbols  $e_1, e_2, \dots, e_e$  are interpreted in  $\mathfrak{A}$  as equivalence relations on  $A$  in local agreement. Then there is a  $\Sigma'$ -enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  such that  $\mathfrak{A}' \models \text{ltr } \varphi$  and that the symbols  $l_1, l_2, \dots, l_e$  are interpreted in  $\mathfrak{A}'$  as equivalence relations on  $A$  in refinement.

*Proof.* Since the binary symbols  $e_1, e_2, \dots, e_e$  are interpreted as equivalence relations on  $A$  in local agreement in  $\mathfrak{A}$ , we may define the levels  $L_1, L_2, \dots, L_e \subseteq A \times A$  and the characteristic E-permutation mapping  $\nu = [E:\text{chperm}]^{\mathfrak{A}} : A \rightarrow \mathbb{S}_e$ . Consider an enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  where  $l_i^{\mathfrak{A}'} = L_i$ . By Remark 3,  $L_i$  are equivalences on  $A$  in refinement. We interpret the unary symbols from the permutation setup  $P$  so that  $[P:\text{data}]^{\mathfrak{A}'} a = \nu(a)$  for all  $a \in A$ . By Remark 10,  $\mathfrak{A}' \models [L:\text{fixperm}]$ . By Remark 12, followed by Lemma 2, for every  $i \in [1, e]$  and  $a \in A$  we have:

$$[L:\text{el-}i]^{\mathfrak{A}'}[a] = l_{\nu(a)^{-1}(i)}^{\mathfrak{A}'}[a] = e_{\nu(a)(\nu(a)^{-1}(i))}^{\mathfrak{A}'}[a] = e_i^{\mathfrak{A}'}[a].$$

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

**Remark 15.** Let  $\varphi$  be a  $\Lambda[\Sigma]$ -formula and let  $\mathfrak{A}$  be an  $L'$ -structure. Suppose that  $\mathfrak{A} \models \text{ltr } \varphi$  and that the symbols  $l_1, l_2, \dots, l_e$  are interpreted as equivalence relations on  $A$  in refinement in  $\mathfrak{A}$ . Then there is a  $\Sigma'$ -enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  such that  $\mathfrak{A}' \models \varphi$  and that the binary symbols  $e_1, e_2, \dots, e_e$  are interpreted as equivalence relations on  $A$  in global agreement in  $\mathfrak{A}'$ .

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

The last two remarks show that a  $\Lambda eE_{\text{local}}$ -formula  $\varphi$  has essentially the same models as the  $\Lambda eE_{\text{refine}}$ -formula  $\text{ltr } \varphi$ , so we have shown:

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

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

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

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  $\Lambda_0^2 eE_a$ ,  $\Lambda_1^2 eE_a$  and  $\Lambda_1^2 E_a$  for  $a \in \{\text{local}, \text{refine}\}$  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  $c : A \rightarrow [1, g]$  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  $\langle D, E \rangle$  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  $\iota : S \hookrightarrow [1, g]$  be any injective function. Define the color  $c$  on  $X$  as  $c(a) = \iota(D[a])$ .  $\square$

**Remark 17.** *Let  $E \subseteq A \times A$  be an equivalence relation on  $A$ ,  $g \in \mathbb{N}^+$  and  $c : A \rightarrow [1, g]$ . 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  $D = \{(a, b) \in E \mid c(a) = c(b)\}$ .  $\square$

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

#### 4 Reductions

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

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

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

**Definition 42.** Define the syntactic operation  $\text{grtr} : \Lambda[\Sigma] \rightarrow \Lambda[\Gamma]$  by:

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

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

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

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

$$\mathfrak{A}' \models \mathbf{e}(a, b) \text{ and } \mathfrak{A}' \models [\mathbf{G}:\mathbf{eq}](a, b),$$

which is equivalent to:

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

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

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

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

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

*Proof.* Since  $\mathfrak{A} \models [\mathbf{G}:\mathbf{betw-1-g}]$ , we have  $[\mathbf{G}:\mathbf{data}]^{\mathfrak{A}} a \in [1, g]$  for all  $a \in A$ . Define  $c : A \rightarrow [1, g]$  by  $c(a) = [\mathbf{G}:\mathbf{data}]^{\mathfrak{A}} a$ . 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  $\Sigma'$ -structure  $\mathfrak{A}'$ , where  $\mathbf{d}^{\mathfrak{A}'} = D$ . Since  $\mathfrak{A}'$  is an enrichment of  $\mathfrak{A} \models \text{grtr } \varphi$ , we have  $\mathfrak{A}' \models \text{grtr } \varphi$ . Let  $a, b \in A$ . Then  $\mathfrak{A}' \models [\Gamma:\mathbf{d}](a, b)$  is equivalent to:

$$\mathfrak{A}' \models \mathbf{e}(a, b) \text{ and } \mathfrak{A}' \models [\mathbf{G}:\mathbf{eq}](a, b),$$

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  $\mathfrak{A}' \models \forall \mathbf{x} \forall \mathbf{y} (e(\mathbf{x}, \mathbf{y}) \leftrightarrow [\Gamma : \mathbf{d}](\mathbf{x}, \mathbf{y}))$  and since  $\mathfrak{A}' \models \text{grtr } \varphi$ , 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  $\mathcal{L}_01E$  has the finite model property and its (finite) satisfiability problem is PSPACE-complete [6]
- The first-order logic of two equivalence relations  $\mathcal{L}_02E$  lacks the finite model property and both the satisfiability and finite satisfiability problems are undecidable [9].

Let  $U(u) = \langle \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_u \rangle$  be an unary predicate signature consisting of the unary predicate symbols  $\mathbf{u}_i$ . Let  $E(e) = \langle \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_e \rangle$  be a binary predicate signature consisting of the builtin equivalence symbols  $\mathbf{e}_j$  in refinement. Let  $\Sigma(u, e) = U(u) + E(e)$ , so  $\Sigma(u, e)$  is a generic predicate signature for the monadic first-order logic  $\mathcal{L}_1eE_{\text{refine}}$ .

### 5.1 Cells

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

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

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

If  $\mathfrak{A}$  is a  $\Sigma$ -structure and  $D = \mathbf{d}^{\mathfrak{A}}$ , then the interpretation  $C = [\Sigma:\text{cell}]^{\mathfrak{A}} \subseteq A \times A$  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  $\mathbf{u}$ -predicates.

**Remark 18.** Let  $\mathfrak{A}$  be a  $\Sigma$ -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  $\mathbf{u}$ -predicates. The equality condition ensures that the mapping is a bijection.  $\square$

**Lemma 5.** *Let  $\mathfrak{A}$  be a  $\Sigma$ -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  $a \in A$ :

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

For  $i \in [0, r]$  let  $\mathfrak{I}_i$  be the set of partial isomorphisms from  $\mathfrak{A}$  to  $\mathfrak{B}$  that have length  $i$  and that are included in  $h$ . The set  $\mathfrak{I}_0$  is nonempty since it contains the empty partial isomorphism. We claim that the sequence  $\mathfrak{I}_0, \mathfrak{I}_1, \dots, \mathfrak{I}_r$  satisfies the back-and-forth conditions of Theorem 1. Let  $i \in [0, r-1]$  and let

$$\mathfrak{p} = \bar{a} \mapsto \bar{b} = a_1 a_2 \dots a_i \mapsto b_1 b_2 \dots b_i \in \mathfrak{I}_i$$

be any partial isomorphism.

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

Suppose that  $a \neq a_k$  for all  $k \in [1, i]$ . Let  $S \subseteq C[a]$  be the set of  $\bar{a}$ -elements in the same  $\mathfrak{A}$ -cell as  $a$ :

$$S = \{a_k \in C[a] \mid k \in [1, i]\}.$$

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  $k \in [1, i]$  and this  $b$  is appropriate.

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

Suppose that  $b \neq b_k$  for all  $k \in [1, i]$ . 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  $\Sigma = \Sigma(u, e) = \langle \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_u, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_e \rangle$  be a predicate signature. Abbreviate the finest two equivalence symbols  $\mathbf{d} = \mathbf{e}_1$  and  $\mathbf{e} = \mathbf{e}_2$ .

**Definition 44.** *Let  $\mathfrak{A}$  be a  $\Sigma$ -structure and let  $D = \mathbf{d}^{\mathfrak{A}}$  and  $E = \mathbf{e}^{\mathfrak{A}}$ . Recall that the set of  $D$ -classes is  $\mathcal{E}D$ . Two  $D$ -classes  $X, Y \in \mathcal{E}D$  are organ-equivalent if they are included in the same  $E$ -class (equivalently  $X \times Y \subseteq E$ ), and the induced substructures  $(\mathfrak{A} \upharpoonright X)$  and  $(\mathfrak{A} \upharpoonright Y)$  are isomorphic. The organ-equivalence relation is  $\mathcal{O} \subseteq \mathcal{E}D \times \mathcal{E}D$ .*

Since  $D$  refines  $E$ , organ-equivalence is an equivalence relation on  $\mathcal{E}D$ . 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  $\mathfrak{h}_{XX} = \text{id}_X$ ,  $\mathfrak{h}_{YX} = \mathfrak{h}_{XY}^{-1}$  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  $\mathcal{O} \subseteq A \times A$  is an equivalence relation on  $A$  that refines  $E$ .

**Remark 19.** Let  $\mathfrak{A}$  be a  $\Sigma$ -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  $\mathfrak{A} \models \mathbf{d}(a_i, a_j)$  iff  $\mathfrak{A} \models \mathbf{d}(b_i, b_j)$  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  $\mathbf{d}$  ensures that the interpretation of  $\mathbf{d}$  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  $\mathcal{O} \subseteq A \times A$  refines the second finest equivalence relation  $E$ , the interpretation of all remaining equivalence symbols  $\mathbf{e}_j$  is preserved.  $\square$

**Lemma 6.** Let  $\mathfrak{A}$  be a  $\Sigma$ -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  $D = \mathbf{d}^{\mathfrak{A}}$ ,  $E = \mathbf{e}^{\mathfrak{A}}$  and let  $\mathcal{A} = \mathcal{E}D$  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 following 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  $a \in A$ :

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

For  $i \in [0, r]$  let  $\mathfrak{I}_i$  be the set of partial isomorphisms from  $\mathfrak{A}$  to  $\mathfrak{B}$  that have length  $i$  and that are included in  $h$ . The set  $\mathfrak{I}_0$  is nonempty since it contains the empty partial isomorphism. We claim that the sequence  $\mathfrak{I}_0, \mathfrak{I}_1, \dots, \mathfrak{I}_r$  satisfies the back-and-forth conditions of Theorem 1. Let  $i \in [0, r-1]$  and let

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

be any partial isomorphism.

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

$$\mathcal{S} = \{D[a_k] \in \mathcal{O}[D[a]] \mid k \in [1, i]\}.$$

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

2. For the back condition, let  $b \in B$ . We have to find some  $a \in A$  such that  $\bar{a}a \mapsto \bar{b}b \in \mathfrak{I}$ . If  $b \in D[b_k]$  for some  $k \in [1, i]$ , then  $a = \mathfrak{h}_{D[b_k]D[a_k]}(b)$  is appropriate. Suppose that  $b \notin D[b_k]$  for all  $k \in [1, i]$ . 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  $u, e \in \mathbb{N}$ ,  $e \geq 2$  and consider the predicate signature  $\Sigma = \Sigma(u, e) = \langle \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_u, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_e \rangle$ . Abbreviate  $\mathbf{d} = \mathbf{e}_1$  and  $\mathbf{e} = \mathbf{e}_2$ . Let  $r \in \mathbb{N}^+$ . There is  $\mathfrak{B} \subseteq \mathfrak{A}$  such that  $\mathfrak{B} \equiv_r \mathfrak{A}$  and  $\langle \mathbf{d}^{\mathfrak{B}}, \mathbf{e}^{\mathfrak{B}} \rangle$  is  $g$ -granular for  $g = g(u, r) = r \cdot ((r + 1)^{2^u} - 1)$ . 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  $D = \mathbf{d}^{\mathfrak{B}}$  and  $E = \mathbf{e}^{\mathfrak{B}}$ . 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  $((r + 1)^{2^u} - 1)$  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  $u, e \in \mathbb{N}$ ,  $e \geq 2$  and consider  $\Sigma = \Sigma(u, e)$ . Let  $\varphi$  be a  $\mathcal{L}[\Sigma]$ -sentence having quantifier rank  $r$ . By Lemma 3 and Lemma 4 about granularity, the formula  $\varphi$  is essentially equisatisfiable with the formula  $\text{grtr } \varphi$ , 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:

$$(\text{FIN-})\text{SAT-}^{\mathcal{L}} 1eE_{\text{refine}} \leq_m^{\text{EXPTIME}} (\text{FIN-})\text{SAT-}^{\mathcal{L}} 1(e - 1)E_{\text{refine}}.$$

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

$$(\text{FIN-})\text{SAT-}^{\mathcal{L}} 0eE_{\text{refine}} \leq_m^{\text{PTIME}} (\text{FIN-})\text{SAT-}^{\mathcal{L}} 1(e - 1)E_{\text{refine}}.$$

**Remark 21.** Let  $u \in \mathbb{N}$  and consider  $\Sigma = \Sigma(u, 1) = \langle \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_u, \mathbf{d} \rangle$ . 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  $\Sigma' = \Sigma + \langle e \rangle$  be an enrichment of  $\Sigma$  with the builtin equivalence symbols  $e$ . Consider an enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  to a  $\Sigma'$ -structure, where  $e^{\mathfrak{A}'} = A \times A$  is interpreted as the full relation on  $A$ . Then  $\langle \mathbf{d}^{\mathfrak{A}'}, e^{\mathfrak{A}'} \rangle$  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  $\langle \mathbf{d}^{\mathfrak{B}'}, e^{\mathfrak{B}'} \rangle$  is  $g$ -granular. Consider the reduct  $\mathfrak{B}$  of  $\mathfrak{B}'$  to a  $\Sigma$ -structure. Let  $D = \mathbf{d}^{\mathfrak{B}}$  and  $E = e^{\mathfrak{B}}$ . 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  $e$  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  $|B| \leq g.r.2^u$ .  $\square$

**Corollary 2.** The logic  $\mathcal{L}_1 1E$  has the finite model property and its (finite) satisfiability problem is in  $\text{N2EXPTIME}$ .

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

**Proposition 5.** For  $e \in \mathbb{N}^+$ , the logic  $\mathcal{L}_1 eE_{\text{refine}}$  has the finite model property and its (finite) satisfiability problem is in  $\text{N}(e+1)\text{EXPTIME}$ .

By Proposition 1 and Proposition 3, the same holds for  $\mathcal{L}_1 eE_{\text{global}}$  and  $\mathcal{L}_1 eE_{\text{local}}$ .

**Proposition 6.** The logic  $\mathcal{L}_1 E_{\text{refine}}$  has the finite model property and its (finite) satisfiability problem is in the forth level of the Grzegorzcz hierarchy  $\mathcal{E}^4$ .

By Proposition 2 and Proposition 4, the same holds for  $\mathcal{L}_1 E_{\text{global}}$  and  $\mathcal{L}_1 E_{\text{local}}$ .

**Proposition 7.** For  $e \geq 2$ , the logic  $\mathcal{L}_0 eE_{\text{refine}}$  has the finite model property and its (finite) satisfiability problem is in  $\text{NeEXPTIME}$ .

By Proposition 1 and Proposition 3, the same holds for  $\mathcal{L}_0 eE_{\text{global}}$  and  $\mathcal{L}_0 eE_{\text{local}}$ .

## 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  $\text{N2EXPTIME}$ -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 coordinates of a cell of the doubly exponential tiling square.

Consider the counter setup  $C(u) = \langle \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_u \rangle$  for  $u \in \mathbb{N}^+$ . Recall that the intention of a counter setup is to encode an arbitrary exponentially bounded value at every element of a structure. Let  $D(u) = C(u) + \langle \mathbf{d} \rangle$  be a predicate signature enriching  $C(u)$  with the builtin equivalence symbol  $\mathbf{d}$ . We will define a system where every  $\mathbf{d}$ -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  $\mathbf{d}$ -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  $\mathbf{d}$ -class. Call the data  $[C:\text{data}]^{\mathfrak{A}}a$ , encoded by the counter setup at  $a$  the *position* of  $a$ .

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

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

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

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

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

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

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

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

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

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

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

$$[D:\text{pos-zero}](x) = \bigwedge_{1 \leq i \leq u} \neg u_i(x).$$

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

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

$$[D:\text{pos-largest}](x) = \bigwedge_{1 \leq i \leq u} u_i(x).$$

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

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

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

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

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

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

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

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

$$[\mathbf{D:pos-full}] = \forall \mathbf{x} \exists \mathbf{y} \left( \mathbf{d}(\mathbf{y}, \mathbf{x}) \wedge [\mathbf{D:pos-zero}](\mathbf{y}) \right) \wedge \forall \mathbf{x} \left( \neg [\mathbf{D:pos-largest}](\mathbf{x}) \rightarrow \exists \mathbf{y} [\mathbf{D:pos-succ}](\mathbf{x}, \mathbf{y}) \right).$$

The first part of this formula asserts that every  $\mathbf{d}$ -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  $\mathbf{d}$ -class at position  $p+1$ . Therefore in any model of  $[\mathbf{D:pos-full}]$ , every  $\mathbf{d}$ -class has  $2^u$  cells. For example, in particular, every  $\mathbf{d}$ -class has cardinality at least  $2^u$ . For the rest of the section, suppose that  $\mathfrak{A} \models [\mathbf{D:pos-full}]$ .

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

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

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

$$[\mathbf{D:pos-(p+1)}](\mathbf{x}) = \exists \mathbf{y} \left( [\mathbf{D:pos-p}](\mathbf{y}) \wedge [\mathbf{D:pos-succ}](\mathbf{y}, \mathbf{x}) \right).$$

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

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

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

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

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

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

$$[\mathbf{D:Zero}](\mathbf{x}) = \forall \mathbf{y} \left( \mathbf{d}(\mathbf{y}, \mathbf{x}) \rightarrow [\mathbf{D:bit-0}](\mathbf{y}) \right).$$

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

---

<sup>1</sup>this is reminiscent to the process of defining a standard translation of modal logic to the two-variable first-order fragment

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

$$[D:\text{Largest}](x) = \forall y \left( d(y, x) \rightarrow [D:\text{bit-1}](y) \right).$$

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

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

$$\begin{aligned} [D:\text{Eq-}M](x) = & \forall y \left( d(y, x) \rightarrow \bigwedge_{0 \leq p < t} \left( [D:\text{pos-}p](y) \rightarrow [D:\text{bit-}(\overline{M}_{p+1})](y) \right) \right) \wedge \\ & \forall x \left( [D:\text{pos-}(t-1)](y) \wedge [D:\text{pos-less}](y, x) \rightarrow [D:\text{bit-0}](x) \right). \end{aligned}$$

The first part of this formula asserts that the bits at the first  $t$  positions of the  $d$ -class of  $x$  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  $\mathfrak{A} \models [D:\text{Eq-}M](a)$  iff the data at the  $D$ -class of  $a$  encodes  $M$ :  $[D:\text{Data}]^{\mathfrak{A}} D[a] = M$ .

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

$$\begin{aligned} [D:\text{Less}](x, y) = & \exists x' \exists y' \left( d(x', x) \wedge d(y', y) \wedge \right. \\ & \left( [D:\text{pos-eq}](x', y') \wedge [D:\text{bit-0}](x') \wedge [D:\text{bit-1}](y') \right) \wedge \quad (\text{Less1}) \\ & \forall x'' \left( [D:\text{pos-less}](x', x'') \rightarrow \exists y'' \left( d(y'', y') \wedge \right. \right. \\ & \left. \left. [D:\text{pos-eq}](y'', x'') \wedge ([D:\text{bit-0}](y'') \leftrightarrow [D:\text{bit-0}](x'')) \right) \right) \Big). \quad (\text{Less2}) \end{aligned}$$

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

$$\begin{aligned} & \exists z \left( d(z, x) \wedge \exists x \left( x = z \wedge \exists z \left( d(z, y) \wedge \exists y \left( y = z \wedge \right. \right. \right. \right. \\ & \quad \left. \left( [D:\text{pos-eq}](x, y) \wedge [D:\text{bit-0}](x) \wedge [D:\text{bit-1}](y) \right) \wedge \quad (\text{Less1}) \\ & \quad \forall z \left( [D:\text{pos-less}](x, z) \rightarrow \exists x \left( x = z \wedge \exists z \left( d(z, y) \wedge \exists y \left( y = z \wedge \right. \right. \right. \right. \\ & \quad \left. \left. [D:\text{pos-eq}](y, x) \wedge ([D:\text{bit-0}](y) \leftrightarrow [D:\text{bit-0}](x)) \right) \right) \right) \Big). \quad (\text{Less2}) \end{aligned}$$



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

$$\begin{aligned}
 [D:\text{Succ}](x, y) = & \exists x' \exists y' \left( d(x', x) \wedge d(y', y) \wedge \right. \\
 & ([D:\text{pos-eq}](x', y') \wedge [D:\text{bit-0}](x') \wedge [D:\text{bit-1}](y')) \wedge \quad (\text{Succ1}) \\
 & \forall x'' ([D:\text{pos-less}](x'', x') \rightarrow [D:\text{bit-1}](x'')) \wedge \quad (\text{Succ2}) \\
 & \forall y'' ([D:\text{pos-less}](y'', y') \rightarrow [D:\text{bit-0}](y'')) \wedge \quad (\text{Succ3}) \\
 & \left. \forall x'' ([D:\text{pos-less}](x', x'') \rightarrow \exists y'' (d(y'', y') \wedge \right. \\
 & [D:\text{pos-eq}](y'', x'') \wedge ([D:\text{bit-0}](y'') \leftrightarrow [D:\text{bit-0}](x'')))) \left. \right). \quad (\text{Succ4})
 \end{aligned}$$

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

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

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

$$[D:\text{Full}] = \exists x [D:\text{Zero}](x) \wedge \forall x (\neg [D:\text{Largest}](x) \rightarrow \exists y [D:\text{Succ}](x, y)).$$

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

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

$$\begin{aligned}
 [D:\text{Eq}](x, y) = & \forall x' \forall y' (d(x', x) \wedge d(y', y) \wedge \\
 & [D:\text{pos-eq}](x', y') \rightarrow ([D:\text{bit-0}](x') \leftrightarrow [D:\text{bit-0}](y'))).
 \end{aligned}$$

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

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

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

$$\begin{aligned}
 [D:\text{Alldiff}] = & \forall x \forall y (\neg d(x, y) \rightarrow \exists x' \exists y' (d(x', x) \wedge d(y', y) \wedge \\
 & [D:\text{pos-eq}](x', y') \wedge \neg ([D:\text{bit-0}](x') \leftrightarrow [D:\text{bit-0}](y')))).
 \end{aligned}$$

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

If  $\mathfrak{A}$  satisfies  $[D:\text{Alldiff}]$  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  $c^0 = \langle t_1^0, t_2^0, \dots, t_n^0 \rangle \subseteq T = [1, k]$  of tiles from the 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

$$D = \langle \mathbf{u}_1^H, \mathbf{u}_2^H, \dots, \mathbf{u}_n^H; \mathbf{u}_1^V, \mathbf{u}_2^V, \dots, \mathbf{u}_n^V; \mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_k^T; \mathbf{d} \rangle.$$

It has the following relevant subsignatures:

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

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

$$[D^{HV}:\text{Full}] \wedge [D^{HV}:\text{Alldiff}] \quad (5.3)$$

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

$$\forall \mathbf{x} \left( \bigwedge_{1 \leq i \leq k} \mathbf{u}_i^T(\mathbf{x}) \rightarrow \bigwedge_{i < j \leq k} \neg \mathbf{u}_j^T(\mathbf{x}) \right) \quad (5.4)$$

asserts that every element has a unique type. The sentence

$$\forall \mathbf{x} \forall \mathbf{y} \left( \mathbf{d}(\mathbf{x}, \mathbf{y}) \rightarrow \bigwedge_{1 \leq i \leq k} (\mathbf{u}_i^T(\mathbf{x}) \leftrightarrow \mathbf{u}_i^T(\mathbf{y})) \right) \quad (5.5)$$

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

$$\forall \mathbf{x} \left( [D^H:\text{Eq}-(j-1)](\mathbf{x}) \wedge [D^V:\text{Zero}](\mathbf{x}) \rightarrow \mathbf{u}_{t_j^0}^T(\mathbf{x}) \right) \quad (5.6)$$

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

$$\forall \mathbf{x} \forall \mathbf{y} \left( [D^H:\text{Succ}](\mathbf{x}, \mathbf{y}) \wedge [D^V:\text{Eq}](\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{(i,j) \in H} \mathbf{u}_i^T(\mathbf{x}) \wedge \mathbf{u}_j^T(\mathbf{y}) \right) \quad (5.7)$$

encodes the horizontal matching condition. The sentence

$$\forall \mathbf{x} \forall \mathbf{y} \left( [D^V:\text{Succ}](\mathbf{x}, \mathbf{y}) \wedge [D^H:\text{Eq}](\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{(i,j) \in V} \mathbf{u}_i^T(\mathbf{x}) \wedge \mathbf{u}_j^T(\mathbf{y}) \right) \quad (5.8)$$

encodes the vertical matching condition.

Combining  $[D^{HV}:\text{pos-full}]$  with the formulas 5.3–5.8, we may encode 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  $\mathcal{L}_1 1E$  is N2EXPTIME-hard. More precisely, even the three-variable fragment  $\mathcal{L}_1^3 1E$  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  $\mathcal{L}_1 eE_{\text{refine}}$ . 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  $E(e) = \langle e_1, e_2, \dots, e_e \rangle$  consisting of the builtin equivalence symbols  $e_i$  in refinement. Abbreviate the *coarsest* equivalence symbol  $\mathbf{d} = e_e$ .

**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  $D(e, u)$  having length polynomial in  $u$ , consisting of unary predicate symbols and containing  $E(e)$ . The following data is effectively defined:*

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

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

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

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

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

$$\mathfrak{A} \models [D(e, u):\text{Zero}](a) \text{ iff } \underline{[D(e, u):\text{Data}]^{\mathfrak{A}} D[a]} = 0.$$

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

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

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

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

## 5 Monadic logics

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

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

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

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

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

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

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

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

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

### 5.5 Hardness with many equivalences in refinement

Define the following formulas:

$$[D':\text{Eq}](x, y) = \forall x' \forall y' (e(x', x) \wedge e(y', y) \wedge \quad (E3)$$

$$[D':\text{pos-eq}](x', y') \rightarrow ([D':\text{bit-0}](x') \leftrightarrow [D':\text{bit-0}](y')))$$

$$[D':\text{Zero}](x) = \forall y (e(y, x) \rightarrow [D':\text{bit-0}](y)) \quad (E4)$$

$$[D':\text{Largest}](x) = \forall y (e(y, x) \rightarrow [1:\text{bit-D'}](y)) \quad (E5)$$

$$[D':\text{Less}](x, y) = \exists x' \exists y' (e(x', x) \wedge e(y', y) \wedge \quad (E6)$$

$$([D':\text{pos-eq}](x', y') \wedge [D':\text{bit-0}](x') \wedge [D':\text{bit-1}](y')) \wedge$$

$$\forall x'' ([D':\text{pos-less}](x', x'') \rightarrow \exists y'' (e(y'', y') \wedge$$

$$[D':\text{pos-eq}](y'', x'') \wedge ([D':\text{bit-0}](y'') \leftrightarrow [D':\text{bit-0}](x''))))$$

$$[D':\text{Succ}](x, y) = \exists x' \exists y' (e(x', x) \wedge e(y', y) \wedge \quad (E7)$$

$$([D':\text{pos-eq}](x', y') \wedge [D':\text{bit-0}](x') \wedge [D':\text{bit-1}](y')) \wedge$$

$$\forall x'' ([D':\text{pos-less}](x'', x') \rightarrow [D':\text{bit-1}](x'')) \wedge$$

$$\forall y'' ([D':\text{pos-less}](y'', y') \rightarrow [D':\text{bit-0}](y'')) \wedge$$

$$\forall x'' ([D':\text{pos-less}](x', x'') \rightarrow \exists y'' (e(y'', y') \wedge$$

$$[D':\text{pos-eq}](y'', x'') \wedge ([D':\text{bit-0}](y'') \leftrightarrow [D':\text{bit-0}](x''))))$$

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

$$[D':\text{Eq-M}](x) = \forall y (e(y, x) \rightarrow \bigwedge_{0 \leq p < t} ([D':\text{pos-p}](y) \rightarrow [D':\text{bit-}\overline{M}_{p+1}](y)) \wedge \quad (E8)$$

$$\forall x ([D':\text{pos-(t-1)}](y) \wedge [D':\text{pos-less}](y, x) \rightarrow [D':\text{bit-0}](x))$$

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  $\mathcal{L}_1 e\text{E}_{\text{refine}}$  is  $N(e+1)\text{EXP TIME}$ -hard. Even the three-variable fragment  $\mathcal{L}_1^3 e\text{E}_{\text{refine}}$  has this property.*

## 5 Monadic logics

By Proposition 1 and Proposition 3, the same holds for  $\mathcal{L}_1^{(3)}eE_{\text{global}}$  and  $\mathcal{L}_1^{(3)}eE_{\text{local}}$ .

**Proposition 10.** *The (finite) satisfiability problem for the monadic first-order logic with many equivalence symbols in refinement  $\mathcal{L}_1E_{\text{refine}}$  is ELEMENTARY-hard. Even the three-variable fragment  $\mathcal{L}_1^3E_{\text{refine}}$  has this property.*

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

## 6 Two-variable first-order logics

In this chapter we investigate questions about the complexity of satisfiability and finite satisfiability of the two-variable first-order logic  $\mathcal{L}^2$  with builtin equivalence relations in refinement. Recall that for this logic we are only interested in predicate signatures restricted to only unary and binary predicate symbols and the formal equality.

The base case for  $\mathcal{L}^2$  and the general case of several *unrelated* builtin equivalence symbols have been studied. The following is known:

- The two-variable first-order logic  $\mathcal{L}^2$  has the finite model property [10] and its (finite) satisfiability problem is NEXPTIME-complete [11].
- The two-variable first-order logic with a single builtin equivalence symbol  $\mathcal{L}^2 1E$  has the finite model property and its (finite) satisfiability problem is NEXPTIME-complete [12].
- The two-variable first-order logic with two *unrelated* builtin equivalence symbols  $\mathcal{L}^2 2E$  lacks the finite model property and both its satisfiability and finite satisfiability problems are N2EXPTIME-complete [13].
- The satisfiability and finite satisfiability problems for the two-variable first-order logic with  $e$  builtin equivalence symbols  $\mathcal{L}^2 eE$  are both undecidable for  $e \geq 3$  [14].

TODO: Summarise results!

### 6.1 Type realizability of the two-variable first-order logic

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

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

where the formulas  $\alpha_i$  are quantifier-free and use at most linearly many new unary predicate symbols. We refer to  $\alpha_0$  as the *universal part*, or the  $\forall\forall$ -part of the formula  $\text{sctr } \varphi$  and to  $\alpha_i$  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  $\mathbf{m}_i$  be a fresh binary predicate symbol with the intended interpretation  $\forall x \forall y (\mathbf{m}_i(x, y) \leftrightarrow \alpha_i(x, y))$ . Since this is a universal sentence, it can be incorporated into the  $\forall\forall$ -part  $\alpha_0$  of the formula. We refer to the symbols  $\mathbf{m}_i$

## 6 Two-variable first-order logics

as the *message symbols*. Hence the formula can be transformed in polynomial time to the form:

$$\forall \mathbf{x} \forall \mathbf{y} (\alpha(\mathbf{x}, \mathbf{y}) \vee \mathbf{x} = \mathbf{y}) \wedge \bigwedge_{1 \leq i \leq m} \forall \mathbf{x} \exists \mathbf{y} (\mathbf{m}_i(\mathbf{x}, \mathbf{y}) \wedge \mathbf{x} \neq \mathbf{y}), \quad (6.1)$$

where the  $\forall \forall$ -part  $\alpha$  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  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  for the two-variable first-order logic  $\mathcal{L}^2$  is a predicate signature  $\Sigma$  together with a sequence  $\bar{\mathbf{m}} = \mathbf{m}_1 \mathbf{m}_2 \dots \mathbf{m}_m$  of distinct binary predicate symbols from  $\Sigma$  having intended interpretation

$$\bigwedge_{1 \leq i \leq m} \forall \mathbf{x} \exists \mathbf{y} (\mathbf{m}_i(\mathbf{x}, \mathbf{y}) \wedge \mathbf{x} \neq \mathbf{y}). \quad (6.2)$$

That is, a classified signature *automatically includes* the  $\forall \exists$ -parts of formulas and  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structures *automatically satisfy* the  $\forall \exists$ -parts. We also require that all  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structures contain at least two elements (that can be enforced by adding the formula  $\forall \mathbf{x} \exists \mathbf{y} (\mathbf{x} \neq \mathbf{y})$  to the intended interpretation of  $\Sigma$  and it doesn't make the problem simpler, since satisfiability in the class of structures of cardinality 1 is trivial).

**Definition 65.** The (finite) classified satisfiability problem for two-variable first-order logic is: given a classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  and a quantifier-free  $\mathcal{L}^2[\Sigma]$ -formula  $\alpha(\mathbf{x}, \mathbf{y})$ , is there a (finite)  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure satisfying eq. (6.1). Note that by virtue of the structure being a  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure, it will also satisfy eq. (6.2) and have cardinality at least 2. Denote the classified satisfiability problem for two-variable first-order logic by CL-SAT- $\mathcal{L}^2$  and its finite version by FIN-CL-SAT- $\mathcal{L}^2$ .

Scott normal form shows that (finite) satisfiability reduces in polynomial time to (finite) classified satisfiability, since satisfiability in structures of cardinality 1 is trivial:

$$(\text{FIN-})\text{SAT-}\mathcal{L}^2 \leq_m^{\text{PTIME}} (\text{FIN-})\text{CL-SAT-}\mathcal{L}^2.$$

**Definition 66.** A type instance  $(\Pi, \mathbf{T})$  over the classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  is a pair of a nonempty set of 1-types  $\Pi \subseteq \Pi[\Sigma]$  and a nonempty set of 2-types  $\mathbf{T} \subseteq \mathbf{T}[\Sigma]$  such that:

- The set of 2-types  $\mathbf{T}$  is closed under inversion, that is  $\tau^{-1} \in \mathbf{T}$  for every  $\tau \in \mathbf{T}$ .
- Every 2-type  $\tau \in \mathbf{T}$  connects 1-types from  $\Pi$ , that is  $\text{tp}_{\mathbf{x}} \tau \in \Pi$  and  $\text{tp}_{\mathbf{y}} \tau \in \Pi$  for every  $\tau \in \mathbf{T}$ . Equivalently, since  $\mathbf{T}$  is closed under inversion, we require that  $(\text{tp}_{\mathbf{x}} \upharpoonright \mathbf{T}) \subseteq \Pi$ .

A  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure  $\mathfrak{A}$  realizes (or is a model of) the type instance  $(\Pi, \mathbf{T})$ , if all 1-types and 2-types realized in  $\mathfrak{A}$  come from the type instance, that is if:

- $\text{tp}^{\mathfrak{A}}[a] \in \Pi$  for all  $a \in A$ , and
- $\text{tp}^{\mathfrak{A}}[a, b] \in \mathbf{T}$  for all  $a \in A$  and  $b \in A \setminus \{a\}$ .



## 6.1 Type realizability of the two-variable first-order logic

The structure  $\mathfrak{A}$  fully realizes  $(\Pi, T)$ , if additionally for every 1-type  $\pi \in \Pi$  there is an element  $a \in A$  realizing  $\pi$  (we don't require analogous condition for the 2-types).

The type instance of  $\mathfrak{A}$  is the type instance  $(\Pi, T)$  defined by:

$$\begin{aligned}\Pi &= \{ \text{tp}^{\mathfrak{A}}[a] \mid a \in A \} \\ T &= \{ \text{tp}^{\mathfrak{A}}[a, b] \mid a \neq b \in A \}.\end{aligned}$$

Clearly,  $(\Pi, T)$  is indeed a type instance and  $\mathfrak{A}$  fully realizes  $(\Pi, T)$ .

The (finite) (full) type realizability problem<sup>1</sup> for  $\mathcal{L}^2$  is the following: given a classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  and a type instance  $(\Pi, T)$  over  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ , is there a (finite)  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure that (fully) realizes  $(\Pi, T)$ . Denote the type realizability problem for  $\mathcal{L}^2$  by TP-REALIZ- $\mathcal{L}^2$  and its finite version by FIN-TP-REALIZ- $\mathcal{L}^2$ . Similarly, denote the full type realizability problem for  $\mathcal{L}^2$  by FULL-TP-REALIZ- $\mathcal{L}^2$  and its finite version by FIN-FULL-TP-REALIZ- $\mathcal{L}^2$ .

**Remark 22.** Since the type instance  $(\Pi', T')$  of every model  $\mathfrak{A}$  of the type instance  $(\Pi, T)$  consists of subsets:  $\Pi' \subseteq \Pi$  and  $T' \subseteq T$ , by guessing these subsets we can reduce the (finite) type realizability problem to the (finite) full type realizability problem in nondeterministic polynomial time:

$$(\text{FIN-})\text{TP-REALIZ-}\mathcal{L}^2 \leq_{\text{m}}^{\text{NPTIME}} (\text{FIN-})\text{FULL-TP-REALIZ-}\mathcal{L}^2.$$

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

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

$$(\text{FIN-})\text{CL-SAT-}\mathcal{L}^2 \leq_{\text{m}}^{\text{EXPTIME}} (\text{FIN-})\text{TP-REALIZ-}\mathcal{L}^2.$$

**Definition 67.** Let  $(\Pi, T)$  be a type instance over  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . Two 1-types  $\pi, \pi' \in \Pi$  are connectable (written  $\pi \sim \pi'$ ), if  $\pi = \text{tp}_{\mathbf{x}} \tau$  and  $\pi' = \text{tp}_{\mathbf{y}} \tau$  for some  $\tau \in T$ . Note that connectability is symmetric since  $T$  is closed under inversion. However, connectability is not necessarily neither transitive nor reflexive:

A 1-type  $\kappa \in \Pi$  is a king type if  $\kappa \not\sim \kappa$ . A 1-type  $\pi \in \Pi$  that is not a king type is a peasant type. The set of king types for the type instance  $(\Pi, T)$  is  $K = K(\Pi, T)$  and the set of peasant types is  $P = P(\Pi, T) = \Pi \setminus K(\Pi, T)$ .

<sup>1</sup>The reason we study the type realizability problem instead of directly studying the satisfiability problem is that it is flexible enough to allow us to employ its solutions in simpler cases to construct solutions in more complicated cases.

**Remark 24.** If  $\mathfrak{A}$  is a full model for  $(\Pi, T)$ , then every two distinct elements  $a \neq b \in A$  realize connectable 1-types. Hence every king type  $\kappa \in K(\Pi, T)$  must be realized once in  $\mathfrak{A}$ .

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

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

- ( $\sigma 1$ ) If  $\tau, \tau' \in \sigma$ , then  $\text{tp}_x \tau = \text{tp}_x \tau'$ , that is the  $x$ -type of every element of  $\sigma$  is the same. Denote the  $x$ -type of every element of  $\sigma$  by  $\text{tp}_x \sigma$ .
- ( $\sigma 2$ ) If  $\kappa = \text{tp}_x \sigma \in K$  is a king type, then no  $\tau \in \sigma$  has  $\text{tp}_y \tau = \kappa$ .
- ( $\sigma 3$ ) If  $\kappa \in K \setminus \{\text{tp}_x \sigma\}$  is any king type distinct from  $\text{tp}_x \sigma$  (note that here  $\text{tp}_x \sigma$  may as well be a peasant type), then only one  $\tau \in \sigma$  has  $\text{tp}_y \tau = \kappa$ .
- ( $\sigma 4$ ) If  $\mathbf{m} \in \bar{\mathbf{m}}$ , then  $\mathbf{m}(x, y) \in \tau$  for some (possibly many)  $\tau \in \sigma$ .

If  $\sigma$  is a star-type such that  $\text{tp}_x \sigma$  is a king type, then  $\sigma$  is a king star-type. Otherwise  $\sigma$  is a peasant star-type.

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

If  $\mathfrak{A}$  is a full  $(\Pi, T)$ -model and  $a \in A$ , the star-type realized by  $a$  is:

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

We proceed to verify that the star-type realized by  $a$  is indeed a star-type:

**Remark 25.** Let  $(\Pi, T)$  be a type instance for  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ ,  $\mathfrak{A}$  be a full  $(\Pi, T)$ -model and let  $a \in A$ . Then the star-type  $\sigma = \text{stp}^{\mathfrak{A}}[a]$  realized by  $a$  is a star-type for the type instance  $(\Pi, T)$ .

*Proof.* The star-type  $\sigma$  is nonempty, since  $\mathfrak{A}$  — a  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure — contains at least 2 elements. We verify the conditions for a star-type:

- ( $\sigma 1$ ) If  $\tau, \tau' \in \sigma$ , then  $\tau = \text{tp}^{\mathfrak{A}}[a, b]$  and  $\tau' = \text{tp}^{\mathfrak{A}}[a, c]$  for some  $b, c \neq a \in A$ , hence  $\text{tp}_x \tau = \text{tp}_x \tau' = \text{tp}_x \sigma$ .
- ( $\sigma 2$ ) If  $\kappa = \text{tp}_x \sigma \in K$  is a king type, then by definition we have that no 2-type  $\tau \in T$  has  $\text{tp}_x \tau = \kappa$  and  $\text{tp}_y \tau = \kappa$ , and so no  $\tau \in \sigma$  has  $\text{tp}_y \tau = \kappa$  since  $\sigma \subseteq T$  since  $\mathfrak{A}$  realizes  $(\Pi, T)$ .
- ( $\sigma 3$ ) Let  $\kappa \in K \setminus \{\text{tp}_x \sigma\}$  be any king type. By Remark 24, there is a unique element  $c \neq a \in A$  realizing  $\kappa$ . Then the 2-type  $\tau = \text{tp}^{\mathfrak{A}}[a, c]$  is the unique satisfying  $\tau \in \sigma$  and  $\text{tp}_y \tau = \kappa$ .
- ( $\sigma 4$ ) Since  $\mathfrak{A}$  is a  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure, it automatically satisfies the  $\forall \exists$ -parts of eq. (6.2), hence for every  $\mathbf{m} \in \bar{\mathbf{m}}$  there must be some  $b \neq a \in A$  such that  $\mathbf{m}(x, y) \in \text{tp}^{\mathfrak{A}}[a, b]$ .

□

**Remark 26** (Star-type extension). *Let  $\sigma$  be a star-type having  $x$ -type  $\pi = \text{tp}_x \sigma$ . Let  $\tau \in T$  be a 2-type and suppose that  $\text{tp}_x \tau = \pi$  and that  $\text{tp}_y \tau$  is not a king type. Then  $\sigma' = \sigma \cup \{\tau\}$  is also a star-type.*

*Proof.* It is straightforward to verify the conditions for a star-type: (σ1) follows from the assumption that  $\text{tp}_x \pi = \text{tp}_x \sigma$ . Conditions (σ2) and (σ3) follow from the assumption that  $\text{tp}_y \pi$  is not a king type. Condition (σ4) follows from the assumption that  $\sigma'$  extends the star-type  $\sigma$ . □

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

(S1) *If  $\tau \in \sigma$  for some  $\sigma \in \mathcal{S}$ , then  $\tau^{-1} \in \sigma'$  for some  $\sigma' \in \mathcal{S}$ , that is there are witnesses for the endpoints for every 2-type used in the certificate. Equivalently,  $\cup \mathcal{S}$  is closed under inversion.*

*Let  $T' = \cup \mathcal{S}$ . Clearly,  $T' \subseteq T$ . We require that  $(\Pi, T')$  is a type instance — the filtered type instance — and that all star-types  $\sigma \in \mathcal{S}$  be star-types for the filtered type instance. In this context we refer to the king types  $K = K(\Pi, T')$  and the peasant types  $P = P(\Pi, T')$  with respect to the filtered type instance. Note that since  $T' \subseteq T$ , a king type with respect to the original type instance must remain a king type with respect to the refined type instance. However, there might be king types with respect to the filtered type instance that were peasant types with respect of the original type instance. This might happen if we remove all 2-types connecting a peasant type with itself when going from  $T$  to  $T'$ .*

(S2) *If  $\pi \in \Pi$  then some (possibly many)  $\sigma \in \mathcal{S}$  has  $\text{tp}_x \sigma = \pi$ , that is every 1-type is witnessed.*

(S3) *If  $\kappa \in K(\Pi, T')$  is a king type, then one  $\sigma \in \mathcal{S}$  has  $\text{tp}_x \sigma = \kappa$ , that is every king type is witnessed once. Note that the existence is already implied by (S2).*

(S4) *If  $\pi, \pi' \in \Pi$  are (possibly the same) 1-types that are not connected, that is no  $\tau \in T'$  has  $\text{tp}_x \tau = \pi$  and  $\text{tp}_y \tau = \pi'$ , then  $\pi = \pi' \in K(\Pi, T')$ , that is two 1-types are not connected iff they are the same king type.*

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  $(\Pi, T)$ . Let  $T' \subseteq T$  be the (nonempty) set of 2-types realized in  $\mathfrak{A}$ . Note that  $T'$  is closed under inversion. For every 2-type  $\tau \in T'$ , let  $a_\tau \neq b_\tau \in A$  be two distinct elements realizing  $\tau$ :  $\text{tp}^{\mathfrak{A}}[a_\tau, b_\tau] = \tau$ . The choice of the elements is made symmetric, that is  $a_{\tau^{-1}} = b_\tau$  and  $b_{\tau^{-1}} = a_\tau$  for every  $\tau \in T'$ . Let*

$$\mathcal{S} = \left\{ \text{stp}^{\mathfrak{A}}[a_\tau] \mid \tau \in T' \right\}.$$

Then  $\mathcal{S}$  is a certificate for the type instance. Moreover, its size is linearly bounded by  $|\mathbf{T}'|$ , hence also by the size of the type instance.

*Proof.* We check the conditions for a certificate:

- (S1) That  $(\Pi, \mathbf{T}')$  is a type instance follows from the observations that  $\cup \mathcal{S} = \mathbf{T}'$ ,  $\mathbf{T}' \subseteq \mathbf{T}$  and that all  $\mathbf{x}$ -types and  $\mathbf{y}$ -types of 2-types from  $\mathbf{T}'$  are contained in  $\Pi$ . Note that  $\mathfrak{A}$  is a full model for the type instance  $(\Pi, \mathbf{T}')$ , hence every star-type  $\text{stp}^{\mathfrak{A}}[a]$  is a star-type for the type instance  $(\Pi, \mathbf{T}')$ .
- (S2) That every 1-type is witnessed follows from the observation that  $\mathfrak{A}$  is a full model for  $(\Pi, \mathbf{T}')$ .
- (S3) If  $\kappa \in K(\Pi, \mathbf{T}')$  is a king type, by Remark 24 it is realized once in  $\mathfrak{A}$ . Let  $c \in A$  be the unique element realizing  $\kappa$ . If  $\sigma \in \mathcal{S}$  has  $\text{tp}_{\mathbf{x}} \sigma = \kappa$ , then  $\sigma = \text{stp}^{\mathfrak{A}}[c]$ , so such  $\sigma$  is unique (more precisely, for every  $\tau \in \mathbf{T}'$  such that  $\text{tp}_{\mathbf{x}} \tau = \kappa$  we must have that  $a_{\tau} = c$ ).
- (S4) Let  $\pi, \pi' \in \Pi$  be two 1-types and suppose that no  $\tau \in \mathbf{T}'$  connects them. Let  $a \in A$  be an element realizing  $\pi$  and  $b$  be an element realizing  $\pi'$ . If  $a \neq b$ , then  $\tau = \text{tp}_{\mathbf{x}}[a, b] \in \text{stp}^{\mathfrak{A}}[a] \subseteq \mathbf{T}'$  will connect  $\pi$  and  $\pi'$ , which is impossible. Hence we must have that  $a = b$ , so  $\pi = \pi' \in K(\Pi, \mathbf{T}')$ .

□

**Lemma 8** (Certificate expansion). *Let  $\mathcal{S}$  be a certificate for the type instance  $(\Pi, \mathbf{T})$  over the classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . Then  $(\Pi, \mathbf{T})$  has a finite full model.*

*More precisely, let  $\mathbf{T}' = \cup \mathcal{S} \subseteq \mathbf{T}$  and let  $t \geq |\mathbf{T}'|$  be a parameter. Then the filtered type instance  $(\Pi, \mathbf{T}')$  has a finite full model  $\mathfrak{A}$  in which every king star-type  $\sigma \in \mathcal{S}$ ,  $\text{tp}_{\mathbf{x}} \sigma \in K(\Pi, \mathbf{T}')$  is realized only once and every peasant star-type  $\sigma \in \mathcal{S}$ ,  $\text{tp}_{\mathbf{x}} \sigma \in P(\Pi, \mathbf{T}')$  is realized at least  $t$  times.*

*Proof.* We adapt the standard strategy<sup>2</sup> used in the proof of the finite model property for the logic  $\mathcal{L}^2$ , as presented in [2]. We build a model  $\mathfrak{A}$  for  $(\Pi, \mathbf{T}')$  as follows. The domain  $A$  of  $\mathfrak{A}$  is the union of the following disjoint sets of elements:

- The singleton set  $A^{\sigma} = \{a^{\sigma}\}$  for every king star-type  $\sigma \in \mathcal{S}$ ,  $\text{tp}_{\mathbf{x}} \sigma \in K(\Pi, \mathbf{T}')$ . Call the elements  $a^{\sigma}$  the kings.
- The three disjoint copies of  $t$  elements  $A^{\sigma} = A_0^{\sigma} \cup A_1^{\sigma} \cup A_2^{\sigma}$  for every peasant star-type  $\sigma \in \mathcal{S}$ ,  $\text{tp}_{\mathbf{x}} \sigma \in P(\Pi, \mathbf{T}')$ , where  $A_i^{\sigma} = \{a_{i1}^{\sigma}, a_{i2}^{\sigma}, \dots, a_{it}^{\sigma}\}$  for  $i \in \{0, 1, 2\}$ . Call the elements  $a_{ij}^{\sigma}$  the peasants.

Let  $\sigma : A \rightarrow \mathcal{S}$  denote the intended star-type of the elements:  $\sigma(a) = \sigma$  on  $A^{\sigma}$ . Let  $\pi : A \rightarrow \Pi$  denote the intended 1-type of the elements:  $\pi(a) = \text{tp}_{\mathbf{x}}(\sigma(a))$ . We proceed to consistently assign 2-types to pairs of distinct elements from the structure on stages.

<sup>2</sup>with the slight difference that our approach doesn't need a *court*, since the information about it is implicit in the certificate

**Realization of kings** We first find witnesses for the intended star-types of the kings. Let  $\sigma \in \mathcal{S}$ ,  $\kappa = \text{tp}_x \sigma \in K(\Pi, T')$  be a king star-type. Consider the unique element  $a = a^\sigma$  having intended star-type  $\sigma$ . For every  $\tau \in \sigma$  we will find an element  $b_\tau$  for the assignment  $\text{tp}^{\mathfrak{A}}[a, b_\tau] = \tau$ , such that all elements  $b_\tau$  are distinct and are distinct from  $a$  and also  $\text{tp}_y \tau = \pi(b_\tau)$ .

1. If  $\text{tp}_y \tau = \kappa' \in K(\Pi, T')$  is a king type, by (σ2) we have  $\kappa' \neq \kappa$ . By (S3) there is a unique star-type  $\sigma'$  having  $\text{tp}_x \sigma' = \kappa'$  and a unique king  $b_\tau = a_{\sigma'} \neq a$  having intended star-type  $\sigma'$ . Note that this assignment is symmetric, that is at the point of considering the type  $\tau^{-1} \in \sigma'$  for the king  $b_\tau$ , we would choose  $a$  as the opposite side of the assignment: we claim that  $\tau^{-1} \in \sigma'$  and it is the unique 2-type from  $\sigma'$  having  $\text{tp}_y \tau^{-1} = \kappa$ . Indeed,  $\tau^{-1} \in T'$ , since  $T'$  is closed under inversion; By (σ3) we have that there is a unique  $\tau' \in \sigma'$  connecting  $\kappa'$  and  $\kappa$ . Since  $\sigma'$  is the unique star-type in  $\mathcal{S}$  having  $\text{tp}_x \sigma' = \kappa'$ , we have that  $\tau'$  is the unique 2-type from  $T'$  connecting  $\kappa'$  and  $\kappa$ . Hence we must have  $\tau' = \tau^{-1}$ . Note that by (σ3) asserting that for the star-type  $\sigma$  there is a unique 2-type  $\tau \in \sigma$  that connects it to a king type (distinct from the origin), we have assigned a 2-types between every pair of distinct kings.
2. If  $\text{tp}_y \tau = \pi' \in P(\Pi, T')$  is a peasant type (hence  $\pi' \neq \kappa$ ), we simultaneously find distinct elements  $b_\eta$  for all 2-types  $\eta \in \sigma$  that are parallel to  $\tau$ . Let

$$U = \{\eta \in \sigma \mid (\eta \parallel \tau)\} = \{\eta \in \sigma \mid \text{tp}_y \eta = \pi'\}$$

be the set of all such 2-types  $\eta$ . Since  $U \subseteq \sigma \subseteq T'$ , there are at most  $t$  2-types  $\eta$  in  $U$ . Since  $T' = \cup \mathcal{S}$  and  $T'$  is closed under inversion, for every  $\eta \in U$  we can find a star-type  $\sigma'_\eta \in \mathcal{S}$  containing its inverse:  $\eta^{-1} \in \sigma'_\eta$ . Since  $\text{tp}_x \sigma'_\eta = \text{tp}_x \eta^{-1} = \pi'$  is not a king type, there are enough distinct elements  $b_\eta \in A_0^{\sigma'_\eta}$  having intended star-type  $\sigma'_\eta$ . Note that these will be distinct from  $a$ , since  $a$  is a king. We assign  $\text{tp}^{\mathfrak{A}}[a, b_\eta] = \eta$  for all  $\eta \in U$ .

**Realization of peasants** We now find witnesses for the intended star-types of the remaining peasant elements. Let  $\sigma \in \mathcal{S}$  be such that  $\pi = \text{tp}_x \sigma \in P(\Pi, T')$  is a peasant type. Let  $a = a_{ij}^\sigma$ , where  $i \in \{0, 1, 2\}$  and  $j \in [1, t]$  be an arbitrary peasant with intended star-type  $\sigma$ . For every  $\tau \in \sigma$  we will find an element  $b_\tau$  for the assignment  $\text{tp}^{\mathfrak{A}}[a, b_\tau] = \tau$ , and again we will ensure that all  $b_\tau$  are distinct and distinct from  $a$  and also that  $\text{tp}_y \tau = \pi(b_\tau)$ .

1. If  $\text{tp}_y \tau = \kappa'$  is a king type then let  $c$  be the element realizing it:  $\pi(c) = \kappa'$ . We consider two cases. First suppose that  $\text{tp}^{\mathfrak{A}}[c, a] = \eta$  has already been assigned during the realization of kings. By that construction we must have that  $a = b_\eta$  and so  $\sigma = \sigma(a) = \sigma'_\eta$ , so that  $\eta^{-1} \in \sigma(a)$ . Note that  $\text{tp}_y \eta^{-1} = \kappa'$ . We claim that  $\eta^{-1} = \tau$ . This is immediate by Condition (σ3), which asserts that there is a unique 2-type  $\tau \in \sigma$  having  $\text{tp}_y \tau = \kappa'$  — a king type. Hence in this case the needed 2-types  $\tau \in \sigma$  have already been assigned in the opposite direction during the realization of kings.

Next suppose that  $\text{tp}^{\mathfrak{A}}[c, a]$  has not been assigned during the realization of kings. Then just assign  $\text{tp}^{\mathfrak{A}}[a, c] = \tau$ . Note that this may extend the actual star-type of the king  $c$  beyond its intended star-type  $\sigma(c)$  by adding the type  $\tau^{-1}$ , but by Remark 26, this extension is still a star-type. That is, in the end, the structure may realize *more* than the intended star-types, but, importantly, *not less*.

2. If  $\text{tp}_{\mathbf{y}} \tau = \pi'$  is not a king type, we simultaneously find distinct peasants  $b_{\eta}$  for all  $\eta \in \sigma$  that are parallel to  $\tau$ . Let  $U = \{\eta \in \sigma \mid \text{tp}_{\mathbf{y}} \eta = \pi'\}$  be the set of all such 2-types  $\eta$ . The key to consistency is to use elements from the *next copy* as witnesses. Let  $i' = (i + 1 \bmod 3) \in \{0, 1, 2\}$ . Since  $U \subseteq \sigma \subseteq T'$ , there are at most  $t$  2-types  $\eta$  in  $U$ . Since  $T' = \cup \mathcal{S}$  and  $T'$  is closed under inversion, for every  $\eta \in U$  we can find a star-type  $\sigma'_{\eta} \in \mathcal{S}$  containing its inverse:  $\eta^{-1} \in \sigma'_{\eta}$ . Since  $\text{tp}_x \sigma'_{\eta} = \text{tp}_x \eta^{-1} = \pi'$  is not a king type, there are enough distinct elements  $b_{\eta} \in A_{i'}^{\sigma'_{\eta}}$  from the next copy of peasants having intended star-type  $\sigma'_{\eta}$ . Since these elements are from the next copy, they are distinct from  $a$ . We assign  $\text{tp}^{\mathfrak{A}}[a, b_{\eta}] = \eta$  for all  $\eta \in U$ . None of these assignments clash with each other, since they have been made between pairs of elements from consecutive copies.

**Completion** For any pair of distinct elements  $a, b \in A$  that has not yet been assigned a 2-type, assign  $\text{tp}^{\mathfrak{A}}[a, b] = \tau$  to arbitrary 2-type  $\tau$  that connects the 1-types  $\pi(a)$  and  $\pi(b)$ . This is possible because during realization of kings we have assigned a 2-type between every pair of distinct kings and by (S4).

The constructed model is a  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure by (σ4) and is a full model for  $(\Pi, T)$  by (S2).  $\square$

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

*Proof.* Let  $(\Pi, T)$  be a type instance for the classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . 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. By Lemma 7 and Lemma 8 such a certificate exists iff  $(\Pi, T)$  is fully satisfiable. Recall that Remark 23 states that a structure is a model for a formula iff it is a model for the type instance consisting of the types consistent with the formula. Also recall Remark 22 stating that type realizability reduces in nondeterministic polynomial time to full type realizability. The logic has the finite model property since in Lemma 8 we constructed a finite full model.  $\square$

As a corollary we get the standard result:

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

## 6.2 Type realizability of the two-variable first-order logic with equivalences in refinement

In this section we consider the logic  $\mathcal{L}^2eE_{\text{refine}}$  featuring  $e \geq 2$  equivalence symbols  $e_1, e_2, \dots, e_e$  in refinement. Abbreviate the coarsest equivalence symbol  $e = e_e$ . Recall from the previous section that every  $\mathcal{L}^2$ -sentence  $\varphi$  can be reduced in polynomial time to an equisatisfiable sentence in the form:

$$\forall x \forall y (\alpha(x, y) \vee x = y) \wedge \bigwedge_{1 \leq i \leq m} \forall x \exists y (\mathbf{m}_i(x, y) \wedge x \neq y),$$

that may use additional unary predicate symbols and the new message symbols  $\mathbf{m}_i$ , where the  $\forall\forall$ -part  $\alpha$  is quantifier-free.

**Definition 70.** A classified signature for  $\mathcal{L}^2eE_{\text{refine}}$  is a predicate signature  $\Sigma$  together with a sequence  $\bar{\mathbf{m}} = \mathbf{m}_1 \mathbf{m}_2 \dots \mathbf{m}_m$  of distinct binary predicate symbols distinct from the equivalence symbols from  $\Sigma$  having the intended interpretation

$$\bigwedge_{1 \leq i \leq m} \forall x \exists y (\mathbf{m}_i(x, y) \wedge x \neq y).$$

Again, note that a classified signature *automatically includes* the  $\forall\exists$ -part of formulas and  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structures *automatically satisfy* the  $\forall\exists$ -part.

The (finite) classified satisfiability problem for  $\mathcal{L}^2eE_{\text{refine}}$  is defined as in Definition 65. Again we have the reduction:

$$(\text{FIN-})\text{SAT-}\mathcal{L}^2 \leq_m^{\text{PTIME}} (\text{FIN-})\text{CL-SAT-}\mathcal{L}^2.$$

Let  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  be a classified signature for  $\mathcal{L}^2eE_{\text{refine}}$ . A type instance  $(\Pi, T)$  over  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  is defined as in Definition 66; also the notions of a model for a type instance, full realization, characteristic type instance of a model and the (finite) (full) type realizability problem for  $\mathcal{L}^2eE_{\text{refine}}$  are defined. Again, the reductions from Remark 22 and Remark 23 hold:

$$\begin{aligned} (\text{FIN-})\text{TP-REALIZ-}\mathcal{L}^2eE_{\text{refine}} &\leq_m^{\text{NPTIME}} (\text{FIN-})\text{FULL-TP-REALIZ-}\mathcal{L}^2eE_{\text{refine}} \\ (\text{FIN-})\text{CL-SAT-}\mathcal{L}^2eE_{\text{refine}} &\leq_m^{\text{EXPTIME}} (\text{FIN-})\text{TP-REALIZ-}\mathcal{L}^2eE_{\text{refine}}. \end{aligned}$$

We proceed to define new terms, specific to the case of many equivalence symbols. The terminology is loosely based on [4].

**Definition 71.** A 2-type  $\tau \in T[\Sigma]$  is a galactic type if  $e(x, y) \in \tau$ . Otherwise, if  $(\neg e(x, y)) \in \tau$ , the type is a cosmic type. The set of galactic 2-types is  $T_g[\Sigma]$ . The set of cosmic 2-types is  $T_c[\Sigma]$ .

Informally, we think of the  $e$ -classes in a structure as galaxies; of the whole structure as the cosmos; of the galactic 2-types as characterizing the interactions in the internals of the galaxies, while cosmic 2-types characterize the interactions between different galaxies.



**Definition 72.** Let  $(\Pi, T)$  be a type instance over  $\langle \Sigma, \bar{m} \rangle$ . Two 1-types  $\pi, \pi' \in \Pi$  are galactically connectable (written  $\pi \sim_g \pi'$ ), if  $\pi = \text{tp}_x \tau$  and  $\pi' = \text{tp}_y \tau$  for some galactic  $\tau \in T$ . The types are cosmically connectable (written  $\pi \sim_c \pi'$ ), if  $\pi = \text{tp}_x \tau$  and  $\pi' = \text{tp}_y \tau$  for some cosmic  $\tau \in T$ .

Note that the types may be both galactically and cosmically connectable and the (unqualified) definition of connectable types from Definition 67 is equivalent to:

$$\pi \sim \pi' \text{ iff } \pi \sim_g \pi' \text{ or } \pi \sim_c \pi'.$$

Recall that a 1-type  $\kappa$  is a *king type* if  $\kappa \not\sim \kappa$ . A 1-type  $\nu$  is a *noble type* if  $\nu \not\sim_c \nu$ . Note that every king type is a noble type. A 1-type  $\pi \in \Pi$  is a *peasant type* if it is not a noble type. Note that this is a different definition of a peasant type from the definition in the no builtin equivalence symbols case. Let  $K = K(\Pi, T)$  be the set of king types,  $N = N(\Pi, T)$  be the set of noble types and  $P = P(\Pi, T)$  be the set of peasant types. The following is a generalization of Remark 24 including noble types:

**Remark 27.** If  $\mathfrak{A}$  is a full model for  $(\Pi, T)$ , then every king type  $\kappa \in K(\Pi, T)$  is realized once in  $\mathfrak{A}$  and every noble type  $\nu \in N(\Pi, T)$  is realized (possibly many times) in only one galaxy of  $\mathfrak{A}$ .

Our strategy to resolve the type realizability problem would be to encode the galactic structure of enough galaxies into instances of the full type realizability problem for the logic featuring one less equivalence symbol.

The following definition characterizes the local structure of a single galaxy in a full model.

**Definition 73.** A cosmic spectrum  $\sigma \subseteq (T \cap T_c[\Sigma])$  for the type instance  $(\Pi, T)$  is a nonempty set of cosmic types satisfying the following conditions:

- ~~( $\sigma 1'$ )~~ Call the set of 1-types  $(\text{tp}_x \upharpoonright \sigma)$  the “inside” of  $\sigma$ . This is not a condition and Condition ( $\sigma 1$ ) for star-types has no analogue in this setting.
- ( $\sigma 2'$ ) If  $\nu \in (\text{tp}_x \upharpoonright \sigma) \cap N$  is a noble type “inside”  $\sigma$ , then no  $\tau \in \sigma$  has  $\text{tp}_y \tau = \nu$ . This is the analogue of Condition ( $\sigma 2$ ) for star-types.
- ( $\sigma 3\nu$ ) If  $\nu \in N \setminus (\text{tp}_x \upharpoonright \sigma)$  is any noble type not occurring “inside”  $\sigma$ , then some (possibly many)  $\tau \in \sigma$  has  $\text{tp}_y \tau = \nu$ . This is an analogue of Condition ( $\sigma 3$ ) for star-types.
- ( $\sigma 3\kappa$ ) If  $\kappa \in K \setminus (\text{tp}_x \upharpoonright \sigma)$  is any king type not occurring “inside”  $\sigma$ , then only one  $\tau \in \sigma$  has  $\text{tp}_y \tau = \kappa$ . Note that the existence already follows from ( $\sigma 3\nu$ ). This is an analogue of Condition ( $\sigma 3$ ) for star-types.
- ~~( $\sigma 4'$ )~~ This is not a condition. The notion of local consistency will be the analogue of Condition ( $\sigma 4$ ) for star-types.

The cosmic spectrum is noble if there is a noble 1-type  $\nu \in (\text{tp}_x \upharpoonright \sigma)$ .



## 6.2 Type realizability of the two-variable first-order logic with equivalences in refinement

**Definition 74.** Let  $\mathfrak{A}$  be a  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure and let  $E = \mathbf{e}^{\mathfrak{A}}$ . The cosmic spectrum  $\text{csp}^{\mathfrak{A}}[X]$  of an  $E$ -class  $X \in \mathcal{E}E$  is the set of cosmic types realized at  $X$ :

$$\text{csp}^{\mathfrak{A}}[X] = \left\{ \text{tp}^{\mathfrak{A}}[a, b] \in \text{T}_c[\Sigma] \mid a \in X, b \in A \setminus \{a\} \right\} = \left\{ \text{tp}^{\mathfrak{A}}[a, b] \mid a \in X, b \in A \setminus X \right\}.$$

Note that the cosmic spectrum is empty iff  $E = A \times A$ . Without loss of generality, we may assume that  $E$  is not the full relation, because the full relation is trivially definable in  $\mathcal{L}^2$ . So without loss of generality, we will consider models with nonempty cosmic spectrums of their  $\mathbf{e}$ -classes.

**Remark 28.** If  $\mathfrak{A}$  is a full  $(\Pi, \text{T})$ -model,  $E = \mathbf{e}^{\mathfrak{A}}$  is not the full relation on  $A$  and  $X \in \mathcal{E}E$  is a galaxy, then  $\text{csp}^{\mathfrak{A}}[X]$  is a cosmic spectrum for  $(\Pi, \text{T})$ .

*Proof.* We check the conditions for a cosmic spectrum. First,  $\sigma = \text{csp}^{\mathfrak{A}}[X]$  is nonempty since  $E$  is not the full relation on  $A$ .

1. If  $\nu \in (\text{tp}_x \upharpoonright \sigma)$  is a noble type, then all elements realizing  $\nu$  in  $\mathfrak{A}$  are in  $X$ , hence there is no cosmic  $\tau \in \sigma$  having  $\text{tp}_y \tau = \nu$ .
2. If  $\nu \notin (\text{tp}_x \upharpoonright \sigma)$  is a noble type, since  $\mathfrak{A}$  is a full model, there is some element  $b \in A \setminus X$  realizing  $\nu$ . Then  $\tau = \text{tp}^{\mathfrak{A}}[a, b]$  is a 2-type having  $\tau \in \sigma$  and  $\text{tp}_y \tau = \nu$ .
3. If  $\kappa \notin (\text{tp}_x \upharpoonright \sigma)$  is a king type, then there is a unique element  $b \in A$  realizing  $\kappa$ . Note that  $b \notin X$  and the 2-type  $\tau = \text{tp}^{\mathfrak{A}}[a, b]$  is the unique  $\tau \in \sigma$  such that  $\text{tp}_y \tau = \kappa$ .

□

We like to think about a cosmic spectrum of a galaxy as the *reason why* that galaxy is possible. For this we need to define the notion of a *locally-consistent* cosmic spectrum. For this we extract a type instance  $(\Pi_\sigma, \text{T}_\sigma)$  over the *simpler logic*  $\mathcal{L}^2(e-1)\text{E}_{\text{refine}}$  out of the cosmic spectrum  $\sigma$ :

**Definition 75.** Let  $(\Pi, \text{T})$  be a type instance over  $\langle \Sigma, \bar{\mathbf{m}} \rangle$  and  $\sigma \subseteq (\text{T} \cap \text{T}_c[\Sigma])$  be a cosmic spectrum for  $(\Pi, \text{T})$ . Let  $\mathbf{in}$  (intended to label the inside of the galaxy) and  $\mathbf{bh}$  (intended to label some black hole outside the galaxy) be two new unary predicate symbols. For a 1-type  $\pi$  or a 2-type  $\tau$ , denote by  $\pi^-$  and  $\tau^-$  the reducts of  $\pi$  and  $\tau$  to the language  $\Sigma - \{\mathbf{e}\}$ . That is,  $\pi^- \subset \pi$  and  $\tau^- \subset \tau$  consist of those literals that do not feature the predicate symbol  $\mathbf{e}$ . We define the spectral type instance  $(\Pi_\sigma, \text{T}_\sigma)$  corresponding to the cosmic spectrum  $\sigma$  as a type instance over the  $\mathcal{L}^2(e-1)\text{E}_{\text{refine}}$ -signature  $\langle \Sigma - \{\mathbf{e}\} + \{\mathbf{in}, \mathbf{bh}\}, \bar{\mathbf{m}} \rangle$  as follows:

- (I) For every  $\pi \in (\text{tp}_x \upharpoonright \sigma)$ , the 1-type  $\pi^- \cup \{\mathbf{in}(x), \neg \mathbf{bh}(x)\}$  is added to  $\Pi_\sigma$ .
- (O) For every  $\pi \in (\text{tp}_y \upharpoonright \sigma)$ , the 1-type  $\pi^- \cup \{\neg \mathbf{in}(x), \neg \mathbf{bh}(x)\}$  is added to  $\Pi_\sigma$ .
- (B) Let  $\beta$  be an arbitrary 1-type extending  $\{\neg \mathbf{in}(x), \mathbf{bh}(x)\}$ . Call  $\beta$  the black hole type. The black hole type is added to  $\Pi_\sigma$ .

**(II)** For every (galactic)  $\tau \in (\mathbf{T} \cap \mathbf{T}_g[\Sigma])$ , the 2-type

$$\tau^- \cup \{\mathbf{in}(x), \neg \mathbf{bh}(x), \mathbf{in}(y), \neg \mathbf{bh}(y)\}$$

is added to  $\mathbf{T}_\sigma$ .

**(IO),(OI)** For every (cosmic)  $\tau \in \sigma$ , the 2-type

$$\tau^- \cup \{\mathbf{in}(x), \neg \mathbf{bh}(x), \neg \mathbf{in}(y), \neg \mathbf{bh}(y)\}$$

and its inverse are added to  $\mathbf{T}_\sigma$ .

**(OO)** For every unordered pair of (not necessarily distinct) 1-types  $\{\pi_\sigma, \pi'_\sigma\} \subseteq \Pi_\sigma$  that are “out”-types:  $\{\neg \mathbf{in}(x), \neg \mathbf{bh}(x)\} \subseteq \pi_\sigma$  and  $\{\neg \mathbf{in}(x), \neg \mathbf{bh}(x)\} \subseteq \pi'_\sigma$ , add an arbitrary 2-type  $\tau$  connecting them (so  $\text{tp}_x \tau = \pi_\sigma$  and  $\text{tp}_y \tau = \pi'_\sigma$ ) and its inverse  $\tau^{-1}$  to  $\mathbf{T}_\sigma$ .

**(IB)** For every 1-type  $\pi_\sigma \in \Pi_\sigma$  that is an “in”-type, so  $\{\mathbf{in}(x), \neg \mathbf{bh}(x)\} \subseteq \pi_\sigma$ , let  $\tau$  be an arbitrary 2-type connecting  $\pi_\sigma$  and the black hole type  $\beta$  that witnesses no messages for  $\pi_\sigma$  and everything for  $\beta$ :  $(\neg \mathbf{m}(x, y)) \in \tau$  and  $\mathbf{m}(y, x) \in \tau$  for every  $\mathbf{m} \in \bar{\mathbf{m}}$ , add  $\tau$  and  $\tau^{-1}$  to  $\mathbf{T}_\sigma$ .

**(OB)** For every 1-type  $\pi_\sigma \in \Pi_\sigma$  that is an “out”-type, so  $\{\neg \mathbf{in}(x), \neg \mathbf{bh}(x)\} \subseteq \pi_\sigma$ , let  $\tau$  be an arbitrary 2-type connecting  $\pi_\sigma$  and the black hole type  $\beta$  that witnesses every message for  $\pi_\sigma$  and for  $\beta$ :  $\mathbf{m}(x, y) \in \tau$  and  $\mathbf{m}(y, x) \in \tau$  for every  $\mathbf{m} \in \bar{\mathbf{m}}$ , add  $\tau$  and  $\tau^{-1}$  to  $\mathbf{T}_\sigma$ .

**Definition 76.** Let  $(\Pi, \mathbf{T})$  be a type instance over the  $\mathcal{L}^2 e \mathbf{E}_{\text{refine}}$ -classified signature  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . The cosmic spectrum  $\sigma \subseteq (\mathbf{T} \cap \mathbf{T}_c[\Sigma])$  is locally consistent if its spectral type instance  $(\Pi_\sigma, \mathbf{T}_\sigma)$  is fully satisfiable (as a type instance for the  $\mathcal{L}^2(e-1) \mathbf{E}_{\text{refine}}$ -classified signature  $\langle \Sigma - \{\mathbf{e}\} + \{\mathbf{in}, \mathbf{bh}\}, \bar{\mathbf{m}} \rangle$ ).

**Remark 29.** Let  $\mathfrak{A}$  be a full  $(\Pi, \mathbf{T})$ -model,  $E = \mathbf{e}^\mathfrak{A}$  is not the full relation of  $A$  and  $X \in \mathcal{E}E$  be a galaxy. Then  $\text{csp}^\mathfrak{A}[X]$  is a locally consistent cosmic spectrum for  $(\Pi, \mathbf{T})$ .

*Proof.* Let  $\Sigma' = \Sigma - \{\mathbf{e}\} + \{\mathbf{in}, \mathbf{bh}\}$ . We build a  $\Sigma'$ -structure  $\mathfrak{A}'$  based on  $\mathfrak{A}$  that fully realizes the spectral type instance  $(\Pi_\sigma, \mathbf{T}_\sigma)$ . The domain of  $\mathfrak{A}'$  is  $A' = A \cup \{c\}$ , where  $c$  is a new black hole element. The 1-type of every element in  $\mathfrak{A}'$  is defined as follows:

**(I)** If  $a \in X$  is inside, then  $\text{tp}^{\mathfrak{A}'}[a] = \text{tp}^\mathfrak{A}[a]^- \cup \{\mathbf{in}(x), \neg \mathbf{bh}(x)\}$ , that is the elements from the galaxy are inside.

**(O)** If  $a \in A \setminus X$  is outside, then  $\text{tp}^{\mathfrak{A}'}[a] = \text{tp}^\mathfrak{A}[a]^- \cup \{\neg \mathbf{in}(x), \neg \mathbf{bh}(x)\}$ , that is the elements outside the galaxy are outside.

**(B)** If  $a = c$ , then  $\text{tp}^{\mathfrak{A}'}[a] = \beta$ , that is the black hole element has the black hole type.

The 2-type between every pair of distinct elements  $a \neq b \in A'$  is defined as follows:

## 6.2 Type realizability of the two-variable first-order logic with equivalences in refinement

(II) If  $a, b \in X$  are inside, then:

$$\text{tp}^{\mathfrak{A}'}[a, b] = \text{tp}^{\mathfrak{A}}[a, b]^- \cup \{\mathbf{in}(x), \neg\mathbf{bh}(x), \mathbf{in}(y), \neg\mathbf{bh}(y)\}.$$

(IO) If  $a \in X$  is inside and  $b \in A \setminus X$  is outside, then:

$$\text{tp}^{\mathfrak{A}'}[a, b] = \text{tp}^{\mathfrak{A}}[a, b]^- \cup \{\mathbf{in}(x), \neg\mathbf{bh}(x), \neg\mathbf{in}(y), \neg\mathbf{bh}(y)\}.$$

(OI) If  $a \in A \setminus X$  is outside and  $b \in X$  is inside, then:

$$\text{tp}^{\mathfrak{A}'}[a, b] = \text{tp}^{\mathfrak{A}}[a, b]^- \cup \{\neg\mathbf{in}(x), \neg\mathbf{bh}(x), \mathbf{in}(y), \neg\mathbf{bh}(y)\}.$$

Note that this produces the 2-type that is exactly the inverse of the previous point.

(OO) If  $a, b \in A \setminus X$  are outside, then let  $\tau \in T_\sigma$  be the 2-type connecting  $\pi_\sigma = \text{tp}^{\mathfrak{A}'}[a]$  and  $\pi'_\sigma = \text{tp}^{\mathfrak{A}'}[b]$  and assign  $\text{tp}^{\mathfrak{A}'}[a, b] = \tau$ . Note that this assignment is symmetric, since the unique 2-type connecting  $\pi'_\sigma$  and  $\pi_\sigma$  in  $T_\sigma$  is  $\tau^{-1}$ .

(IB) If  $a \in X$  is inside and  $b = c$  is the black hole, then let  $\tau \in T_\sigma$  be the unique 2-type connecting  $\pi_\sigma = \text{tp}^{\mathfrak{A}'}[a]$  and  $\beta$  and assign  $\text{tp}^{\mathfrak{A}'}[a, b] = \tau$ .

(OB) If  $a \in A \setminus X$  is outside and  $b = c$  is the black hole, then let  $\tau \in T_\sigma$  be the unique 2-type connecting  $\pi_\sigma = \text{tp}^{\mathfrak{A}'}[a]$  and  $\beta$  and assign  $\text{tp}^{\mathfrak{A}'}[a, b] = \tau$ .

We have to verify that  $\mathfrak{A}'$  is indeed a  $\langle \Sigma', \bar{\mathbf{m}} \rangle$ -structure, that is that the star-type of every element of  $\mathfrak{A}'$  contains  $\mathbf{m}(x, y)$  for every  $\mathbf{m} \in \bar{\mathbf{m}}$ .

(I) For the in-elements  $a \in X \subset A'$ , every  $\mathbf{m} \in \bar{\mathbf{m}}$  that is included in  $\text{stp}^{\mathfrak{A}}[a]$  is also included in  $\text{stp}^{\mathfrak{A}'}[a]$ , since a 2-type  $\tau^-$  contains  $\mathbf{m}(x, y)$  iff  $\tau$  contains  $\mathbf{m}(x, y)$ <sup>3</sup>.

(O) For the out-elements  $a \in (A \setminus X) \subset A'$ , we just need to recall that in the (OB) case, we included a 2-type  $\tau$  that witnesses every message symbol for both of its ends:  $\mathbf{m}(x, y) \in \tau$  and  $\mathbf{m}(y, x) \in \tau$ . By construction, this is exactly the 2-type between  $a$  and the black hole  $c$ .

(B) For the black hole  $c \in A'$ , we just have to recall that in the (IB) and (OB) cases, all included 2-types connecting to the black hole type contained  $\mathbf{m}(y, x)$  for every  $\mathbf{m} \in \bar{\mathbf{m}}$ .

Since  $\mathfrak{A}$  was a full  $(\Pi, T)$ -model, it is clear that  $\mathfrak{A}'$  is a full  $(\Pi_\sigma, T_\sigma)$ -model.  $\square$

**Definition 77.** Let  $\mathfrak{A}$  be a full model for the type instance  $(\Pi, T)$  and let  $E = e^{\mathfrak{A}}$ . Let  $N \subseteq \Pi$  be the set of noble types (realized in  $\mathfrak{A}$ ). For every noble type  $\nu \in N$ , let  $X_\nu \in \mathcal{E}E$  be the (unique)  $E$ -class containing a (not necessarily unique) element realizing  $\nu$ , that is the set  $\{X_\nu \mid \nu \in N\}$  is the set of all  $E$ -classes containing a noble element. Note that

<sup>3</sup>this is where we use the condition that the message symbols of a classified signature are distinct from the builtin equivalence symbols

the cardinality of this set is at most  $|\Pi|$ , since by definition no noble type is cosmically connected to itself. The model  $\mathfrak{A}$  is nobly distinguished if every element in  $X_\nu$  realizes a noble type for every  $\nu \in N$ . That is, a model is nobly distinguished if every class that contains a noble element contains only noble elements.

We want to restrict our attention to the problem of nobly distinguished type realizability. To do this we will use some new unary predicate symbols and will “color” the elements living along with nobles in a single  $e$ -class to make them noble as well. For this we need to encode possible connections between noble types and 1-types realized in the same class, so we enrich the type instance to accommodate such information.

**Definition 78.** Let  $(\Pi, T)$  be a type instance for  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . For every 1-type  $\pi \in \Pi$ , let  $\mathbf{p}_\pi$  be a new unary predicate symbol. Let  $\Sigma' = \Sigma + \langle \mathbf{p}_\pi \mid \pi \in \Pi \rangle$  be an enrichment of  $\Sigma$  featuring the new predicate symbols. For every  $\pi \in \Pi$ , let

$$\mathbf{p}^\pi = \mathbf{p}^\pi(\mathbf{x}) = \{\mathbf{p}_\pi(\mathbf{x})\} \cup \{\neg \mathbf{p}_\rho(\mathbf{x}) \mid \rho \in \Pi \setminus \{\pi\}\}.$$

For every pair of (not necessarily distinct) 1-types  $\pi, \pi' \in \Pi$ , let  $\pi'_\pi$  be a 1-type over  $\Sigma'$  defined as follows:

$$\pi'_\pi = \pi' \cup \mathbf{p}^\pi(\mathbf{x}).$$

We refer to  $\pi'_\pi$  as the  $\pi$ -copy of  $\pi'$ . Let  $\Pi_\Pi = \{\pi'_\pi \mid \pi', \pi \in \Pi\}$  be the set of all the copies. Note that this is a set of 1-types over  $\Sigma'$ . Let  $T_\Pi$  be a set of 2-types over  $\Sigma'$  defined as follows:

$$T_\Pi = \left\{ \tau \cup \mathbf{p}^\pi(\mathbf{x}) \cup \mathbf{p}^{\pi'}(\mathbf{y}) \mid \pi, \pi' \in \Pi \right\}.$$

Then  $(\Pi_\Pi, T_\Pi)$  is a type instance for  $\langle \Sigma', \bar{\mathbf{m}} \rangle$ . Note that the size of  $(\Pi_\Pi, T_\Pi)$  is polynomially bounded by the size of  $(\Pi, T)$ .

**Remark 30.** Let  $\mathfrak{A}$  be a full model for the type instance  $(\Pi, T)$  over  $\langle \Sigma, \bar{\mathbf{m}} \rangle$ . Then there exists a  $\langle \Sigma', \bar{\mathbf{m}} \rangle$ -type instance  $(\Pi', T')$  such that  $\Pi$  “is contained in”  $\Pi'$ ,  $\Pi' \subseteq \Pi_\Pi$ ,  $T' \subseteq T_\Pi$  and a  $\Sigma'$ -enrichment  $\mathfrak{A}'$  of  $\mathfrak{A}$  such that  $\mathfrak{A}'$  is a nobly distinguished full model of  $(\Pi', T')$ . The set of 1-types  $\Pi$  “is contained in”  $\Pi'$  if for every  $\pi' \in \Pi$  there is some  $\pi \in \Pi$  such  $\pi'_\pi \in \Pi'$ .

*Proof.* TODO SKETCH: Let  $E = e^{\mathfrak{A}}$ . Recall that for every noble  $\nu \in N \subseteq \Pi$ ,  $X_\nu \in \mathcal{E}E$  is the unique  $E$ -class containing a (not necessarily unique) element  $c \in X$  realizing  $\nu$ . For every  $E$ -class  $X$  that contains some noble element, arbitrary choose one noble type  $\nu$  realized by it:  $X = X_\nu$ . For every other  $E$ -class  $Y$  that contains no noble element, arbitrary choose one 1-type  $\pi$  realized in it. Define the enrichment  $\mathfrak{A}'$  as follows: for every  $a \in A$ , if  $a \in X_\nu$  is an element of an  $E$ -class containing a noble element, let  $\text{tp}^{\mathfrak{A}'}[a] = \text{tp}^{\mathfrak{A}}[a]_\nu$ . Otherwise  $a \in Y$ , where  $Y$  is an  $E$ -class containing no noble elements; let  $\pi$  be the chosen 1-type realized in  $Y$  and let  $\text{tp}^{\mathfrak{A}'}[a] = \text{tp}^{\mathfrak{A}}[a]_\pi$ . The characteristic type instance  $(\Pi', T')$  for  $\mathfrak{A}'$  satisfies the conditions of this remark.  $\square$

TODO: In nondeterministic polynomial time we may reduce the (full) (finite) type satisfiability problem to the (full) (finite) type satisfiability problem over the class of

## 6.2 Type realizability of the two-variable first-order logic with equivalences in refinement

nobly distinguished structures. That is, the nobly distinguished structures form a non-deterministic polynomial time reduction class for (full) (finite) satisfiability.

**Definition 79.** A certificate  $\mathcal{S}$  for the type instance  $(\Pi, T)$  is a nonempty set of locally consistent cosmic spectrums satisfying the following conditions:

1. If  $\pi \in \Pi$ , then some (possibly many)  $\sigma \in \mathcal{S}$  have  $\pi \in (\text{tp}_x \upharpoonright \sigma)$ .
2. If  $\nu \in \Pi$  is noble, then only one  $\sigma \in \mathcal{S}$  has  $\nu \in (\text{tp}_x \upharpoonright \sigma)$ . Note that the existence is already implied by Condition 1.
3. If  $\tau \in \cup \mathcal{S}$ , then  $\tau^{-1} \in \cup \mathcal{S}$ .
4. (If  $\sigma \in \mathcal{S}$  is noble, then for every  $\tau \in \sigma$  there is another  $\sigma' \in \mathcal{S}, \sigma' \neq \sigma$  such that  $\tau \in \sigma'$ .) TODO: we don't need this for nobly distinguished structures!

Again, a note that a certificate may have size that is exponential with respect to the size of the underlying type instance. However, polynomial certificates exist:

**Lemma 9** (Certificate extraction). *Let  $\mathfrak{A}$  be a full model for the type instance  $(\Pi, T)$  and suppose that  $E = e^{\mathfrak{A}}$  is not the full relation on  $A$ . Recall that the cardinality of  $\mathfrak{A}$  is at least 2. Let  $T' \subseteq (T \cap T_c[\Sigma])$  be the set of cosmic 2-types from the type instance that are realized in  $\mathfrak{A}$ . For every cosmic 2-type  $\tau \in T'$  consider two distinct elements  $a_\tau \neq b_\tau \in A$  realizing  $\tau$ :  $\text{tp}^{\mathfrak{A}}[a_\tau, b_\tau] = \tau$ . Let*

$$\mathcal{S} = \left\{ \text{csp}^{\mathfrak{A}}[E[a_\tau]] \mid \tau \in T' \right\} \cup \left\{ \text{csp}^{\mathfrak{A}}[E[b_\tau]] \mid \tau \in T' \right\}.$$

*Then  $\mathcal{S}$  is a certificate for the type instance. Moreover, its size is linearly bounded by  $|T|$ , hence also by the size of the type instance.*

*Proof.* By Remark 29, all elements  $\sigma \in \mathcal{S}$  are locally consistent cosmic spectrums. We check the conditions for a certificate:

1. If  $\pi \in \Pi$ , since  $\mathfrak{A}$  is a full model for  $(\Pi, T)$ , there is some  $a \in A$  such that  $\text{tp}^{\mathfrak{A}}[a] = \pi$ . Since  $E$  is not the full relation on  $A$ , there is some  $b \in A \setminus E[a]$  and so  $\tau = \text{tp}^{\mathfrak{A}}[a, b] \in T'$ , such that  $\pi = \text{tp}_x \tau$  and  $\tau \in \text{csp}^{\mathfrak{A}}[E[a_\tau]]$ .
2. Let  $\nu \in \Pi$  be noble. Then all elements  $a \in A$  realizing  $\nu$  are in the same  $E$ -class: for every  $\tau \in (T \cup T_c[\Sigma])$  such that  $\text{tp}_x \tau = \nu$ , we have  $E[a_\tau] = X$ .
3. That  $\tau \in \cup \mathcal{S}$  implies  $\tau^{-1} \in \cup \mathcal{S}$  follows immediately from the definition of  $\mathcal{S}$ .
4. Let  $\sigma \in \mathcal{S}$  be noble and let  $\tau \in \sigma$  be any cosmic 2-type. Then  $\sigma = \text{csp}^{\mathfrak{A}}[X]$  for some unique  $X \in \mathcal{E}E$  and  $\tau = \text{tp}^{\mathfrak{A}}[a_\tau, b_\tau]$ . If  $a_\tau \in X$ , since  $\tau$  is cosmic, we must have that  $b_\tau \notin X$ , so  $\sigma' = \text{csp}^{\mathfrak{A}}[E[b_\tau]] \neq \sigma$  is appropriate.
5. TODO: If  $a_\tau \notin X$ , this breaks! Need to define certificate more precisely! For this we may define a weak form of differentiation! In nobly distinguished structures this must work!

□

### 6.3 NOTES

TODO: Certificate extraction: for every cosmic 2-type  $\tau$  realized in  $\mathfrak{A}$ , let  $a_\tau, b_\tau$  be a pair of distinct elements (and in distinct  $e$ -classes, since  $\tau$  is cosmic) realizing it. Take:

$$\mathcal{S} = \left\{ \text{csp}^{\mathfrak{A}}[E[a_\tau]], \text{csp}^{\mathfrak{A}}[E[b_\tau]] \mid \tau \dots \right\}$$

# 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.
- [10] Michael Mortimer. On languages with two variables. *Mathematical Logic Quarterly*, 21(1):135–140, 1975.
- [11] Erich Grädel, Phokion G Kolaitis, and Moshe Y Vardi. On the decision problem for two-variable first-order logic. *Bulletin of symbolic logic*, 3(01):53–69, 1997.
- [12] Emanuel Kieroński. Results on the guarded fragment with equivalence or transitive relations. In *Computer Science Logic*, pages 309–324. Springer, 2005.
- [13] Emanuel Kieroński, Jakub Michaliszyn, Ian Pratt-Hartmann, and Lidia Tendera. Two-variable first-order logic with equivalence closure. *SIAM Journal on Computing*, 43(3):1012–1063, 2014.

## *Bibliography*

- [14] Emanuel Kieronski and Martin Otto. Small substructures and decidability issues for first-order logic with two variables. In *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*, pages 448–457. IEEE, 2005.