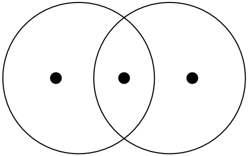
Satisfiability with Equivalences in Agreement, Part 1

Krasimir Georgiev

September 1, 2016

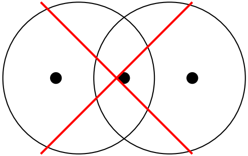
Agreement

- The sets A and B *intersect strictly* if $A \cap B$, $A \setminus B$ and $B \setminus A$ are nonempty.



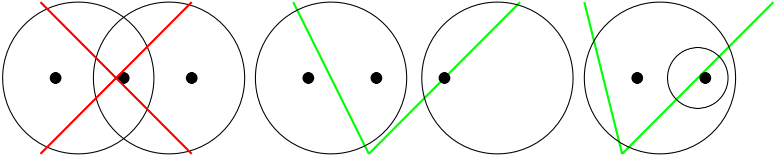
Agreement

- The sets A and B *intersect strictly* if $A \cap B$, $A \setminus B$ and $B \setminus A$ are nonempty.



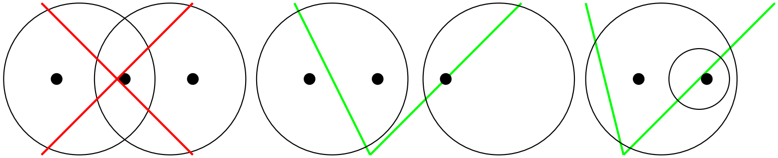
- **What happens if we avoid these?**

Agreement



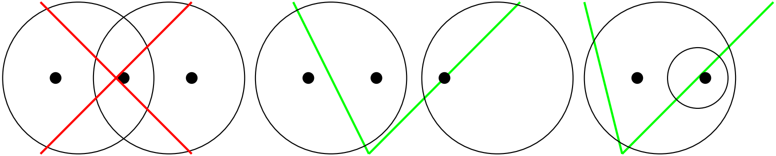
- Two equivalence relations G, E on A are *in local agreement* if their classes do not strictly intersect

Agreement



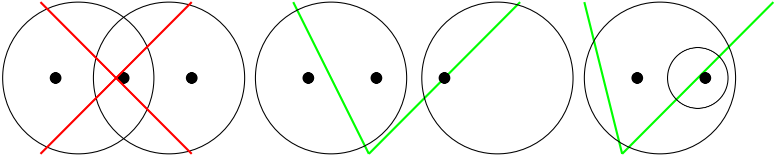
- Two equivalence relations G, E on A are *in local agreement* if their classes do not strictly intersect
- equivalently either $G[a] \subseteq E[a]$ or $E[a] \subseteq G[a]$ for every $a \in A$

Agreement



- Two equivalence relations G, E on A are *in local agreement* if their classes do not strictly intersect
- equivalently either $G[a] \subseteq E[a]$ or $E[a] \subseteq G[a]$ for every $a \in A$
- *(colored) foam hierarchy: balloons in baloons*

Agreement



- Two equivalence relations G, E on A are *in local agreement* if their classes do not strictly intersect
- equivalently either $G[a] \subseteq E[a]$ or $E[a] \subseteq G[a]$ for every $a \in A$
- *(colored) foam hierarchy: balloons in baloons*
- *siam twin matryoshkas*

This work

- define and characterize notions of agreement – ***refinement***, ***local agreement*** *and* ***global agreement***
- find the **computational complexity** of the **satisfiability** of the ***monadic*** and the ***two-variable*** fragment with respect to these notions

Overview

Setups

- use unary predicate symbols to “encode” data at elements of structures
- example: a permutation setup “encodes” a permutation at every element of a structure
- bits $\to$ binary counters $\to$ vectors $\to$ permutations

Bit Setups

The set of *bits* is $\mathbb{B} = \{0, 1\}$. A *bit setup* is a predicate signature $\mathrm{B} = \langle \boldsymbol{u} \rangle$ consisting of a single unary predicate symbol $\boldsymbol{u}$.

- Given $\mathfrak{A}$, define the function $[\boldsymbol{u}{:}\mathrm{data}]^{\mathfrak{A}} : A \to \mathbb{B}$ by:

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

- If $d \in \mathbb{B}$, define the formula $[\boldsymbol{u}{:}\text{eq-}d](\boldsymbol{x})$ by:

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

- Property: $\mathfrak{A} \models [\boldsymbol{u}{:}\text{eq-}d](a)$ iff $[\boldsymbol{u}{:}\mathrm{data}]^{\mathfrak{A}} a = d$.

Bit Setups Formulas

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

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

Counter Setups

The set of *t-bit numbers* is $\mathbb{B}_t = [0, 2^t - 1]$. A *t-bit counter setup* is a predicate signature $\mathrm{C} = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_t \rangle$ consisting of t unary predicate symbols.

- Given $\mathfrak{A}$, define the function $[\![\mathrm{C}\textbf{:}\mathrm{data}]\!]^{\mathfrak{A}} : A \to \mathbb{B}_t$ that returns a t-bit number for any $a \in A$ by:

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

Counter Setups Formulas

We can define ***small formulas*** with the following properties:

- $\mathfrak{A} \models [\mathrm{C}\text{:eq-}d](a)$ iff $[\mathrm{C}\text{:data}]^{\mathfrak{A}} a = d$.
- $\mathfrak{A} \models [\mathrm{C}\text{:eq}](a, b)$ iff $[\mathrm{C}\text{:data}]^{\mathfrak{A}} a = [\mathrm{C}\text{:data}]^{\mathfrak{A}} b$.
- $\mathfrak{A} \models [\mathrm{C}\text{:less}](a, b)$ iff $[\mathrm{C}\text{:data}]^{\mathfrak{A}} a < [\mathrm{C}\text{:data}]^{\mathfrak{A}} b$.
- $\mathfrak{A} \models [\mathrm{C}\text{:succ}](a, b)$ iff $[\mathrm{C}\text{:data}]^{\mathfrak{A}} b = 1 + [\mathrm{C}\text{:data}]^{\mathfrak{A}} a$.
- $\mathfrak{A} \models [\mathrm{C}\text{:betw-}d\text{-}e](a)$ iff $d \leq [\mathrm{C}\text{:data}]^{\mathfrak{A}} a \leq e$.

Vector Setups

The set of *n-dimensional t-bit vectors* is $\mathbb{B}_t^n$. An *n-dimensional t-bit vector setup* is a predicate signature $\mathrm{V} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{nt} \rangle$ of (nt) distinct unary predicate symbols.

- The *counter setup* $\mathrm{V}(p)$ of V at position $p \in [1, n]$ is $\mathrm{V}(p) = \langle \boldsymbol{u}_{p1}, \boldsymbol{u}_{p2}, \ldots, \boldsymbol{u}_{pt} \rangle$.
- $\mathbf{[}\mathrm{V}\mathbf{:}\mathrm{data}\mathbf{]}^{\mathfrak{A}} a = \left(\mathbf{[}\mathrm{V}(1)\mathbf{:}\mathrm{data}\mathbf{]}^{\mathfrak{A}} a, \mathbf{[}\mathrm{V}(2)\mathbf{:}\mathrm{data}\mathbf{]}^{\mathfrak{A}} a, \ldots, \mathbf{[}\mathrm{V}(n)\mathbf{:}\mathrm{data}\mathbf{]}^{\mathfrak{A}} a \right)$.

Vector Setups

We can define ***small formulas***

- $\mathfrak{A} \models [\mathrm{V}(pq)\text{:eq}](a)$ iff $[\mathrm{V}(p)\text{:data}]^{\mathfrak{A}} a = [\mathrm{V}(q)\text{:data}]^{\mathfrak{A}} a$.
- $\mathfrak{A} \models [\mathrm{V}(pq)\text{:less}](a)$ iff $[\mathrm{V}(p)\text{:data}]^{\mathfrak{A}} a < [\mathrm{V}(q)\text{:data}]^{\mathfrak{A}} a$.
- $\mathfrak{A} \models [\mathrm{V}(pq)\text{:succ}](a)$ iff $[\mathrm{V}(q)\text{:data}]^{\mathfrak{A}} a = 1 + [\mathrm{V}(p)\text{:data}]^{\mathfrak{A}} a$.
- antilexicographic ordering, e.g. $(1, 1, 0) \prec (0, 0, 1)$:
  $\mathfrak{A} \models [\mathrm{V}\text{:less}](a, b)$ iff $[\mathrm{V}\text{:data}]^{\mathfrak{A}} a \prec [\mathrm{V}\text{:data}]^{\mathfrak{A}} b$.

Permutation Setups

The set of permutations of $[1, n]$ is $\mathbb{S}_n$.
Encode an n-permutation $\nu \in \mathbb{S}_n$ by the n-dimensional t-bit vector $(\nu(1), \nu(2), \ldots, \nu(n))$, where t is the bitsize of n.
An *n-permutation setup* $\mathrm{P} = \langle \boldsymbol{u}_{11}, \boldsymbol{u}_{12}, \ldots, \boldsymbol{u}_{nt} \rangle$ is just an n-dimensional t-bit vector.
The ***small formula*** $\mathbf{[}\mathrm{P}\mathbf{:}\mathrm{perm}\mathbf{]} = \mathbf{[}\mathrm{P}\mathbf{:}\mathrm{betw\text{-}1\text{-}}n\mathbf{]} \wedge \mathbf{[}\mathrm{P}\mathbf{:}\mathrm{alldiff}\mathbf{]}$ asserts that the vector setup encodes exactly the permutations.

Agreement

Let $E_1, E_2, \ldots, E_n$ be a sequence of equivalence relations on A. The sequence is in

- *refinement* if it forms a chain under inclusion, that is if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_n$

Agreement

Let $E_1, E_2, \ldots, E_n$ be a sequence of equivalence relations on A.
The sequence is in

- *refinement* if it forms a chain under inclusion, that is if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_n$
- *global agreement* if it can be rearranged into a chain under inclusion, that is if $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(n)}$ for some permutation ν of $[1, n]$

Agreement

Let $E_1, E_2, \dots, E_n$ be a sequence of equivalence relations on A. The sequence is in

- *refinement* if it forms a chain under inclusion, that is if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_n$
- *global agreement* if it can be rearranged into a chain under inclusion, that is if $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(n)}$ for some permutation ν of $[1, n]$
- *local agreement* if for every $a \in A$, the sequence of equivalence classes $E_1[a], E_2[a], \dots, E_n[a]$ can be rearranged into a chain under inclusion, that is if for every $a \in A$ there is some permutation $\nu(a)$ of $[1, n]$ such that $E_{\nu(a)(1)}[a] \subseteq E_{\nu(a)(2)}[a] \subseteq \cdots \subseteq E_{\nu(a)(n)}[a]$

Agreement

Let $E_1, E_2, \ldots, E_n$ be a sequence of equivalence relations on A. The sequence is in

- *refinement* if it forms a chain under inclusion, that is if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_n$
- *global agreement* if it can be rearranged into a chain under inclusion, that is if $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(n)}$ for some permutation ν of $[1, n]$
- *local agreement* if for every $a \in A$, the sequence of equivalence classes $E_1[a], E_2[a], \ldots, E_n[a]$ can be rearranged into a chain under inclusion, that is if for every $a \in A$ there is some permutation $\nu(a)$ of $[1, n]$ such that $E_{\nu(a)(1)}[a] \subseteq E_{\nu(a)(2)}[a] \subseteq \cdots \subseteq E_{\nu(a)(n)}[a]$
- refinement $\implies$ global agreement $\implies$ local agreement

Intuition

Intuitively

- global agreement $=$ refinement $+$ a permutation

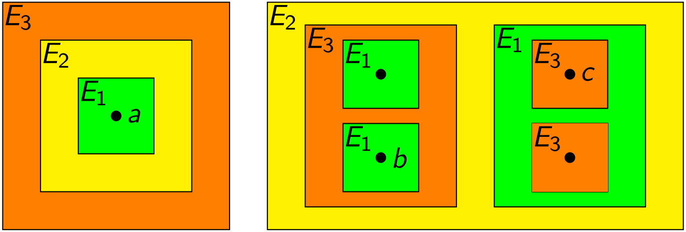
Intuition

Intuitively

- global agreement = refinement + a permutation
- local agreement = refinement + locally agreeing permutations

Example

Example of a sequence E_1, E_2, E_3 in local agreement:



Characterization

Lemma

The sequence E_1, E_2 of two equivalence relations on A is in local agreement iff $E_1 \cup E_2$ is an equivalence relation on A.

Theorem

The sequence $E_1, E_2, \dots, E_n$ of equivalence relations on A is in local agreement iff the union of any nonempty subsequence is an equivalence relation on A, that is for any $m \in [1, n]$ and $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ we have that $E_{i_1} \cup E_{i_2} \cup \cdots \cup E_{i_m}$ is an equivalence relation on A.

Levels

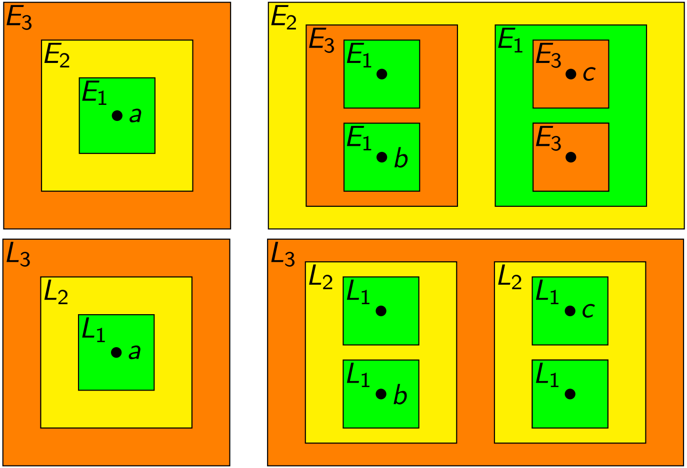
The *level sequence* $L_1, L_2, \ldots, L_n$ of the sequence $E_1, E_2, \ldots, E_n$ of equivalence relations on A in local agreement is defined by:

$$L_m = \bigcap \{E_{i_1} \cup E_{i_2} \cup \cdots \cup E_{i_m} \mid 1 \le i_1 < i_2 < \cdots < i_m \le n\}.$$

Remark

The level sequence is a sequence of equivalence relations on A in refinement.

Example



Permutations

Lemma

Let $E_1, E_2, \ldots, E_n$ be a sequence of equivalence relations on A in local agreement having level sequence $L_1, L_2, \ldots, L_n$. Suppose that $a \in A$ and that ν is any permutation witnessing the local agreement at a:

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

Then $L_k[a] = E_{\nu(k)}[a]$ for any $k \in [1, n]$.

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables
- e, if given, specifies the number of builtin equivalence symbols

Notation

$$\mathcal{L}_p^v e\mathrm{E}_{\mathsf{a}}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables
- e, if given, specifies the number of builtin equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{global}, \mathsf{local}\}$, if given, specifies the agreement condition between the equivalence symbols

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables
- e, if given, specifies the number of builtin equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{global}, \mathsf{local}\}$, if given, specifies the agreement condition between the equivalence symbols
- if p is not given, the signature consists of arbitrary many unary and binary predicate symbols

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables
- e, if given, specifies the number of builtin equivalence symbols
- $\mathsf{a} \in \{\text{refine}, \text{global}, \text{local}\}$, if given, specifies the agreement condition between the equivalence symbols
- if p is not given, the signature consists of arbitrary many unary and binary predicate symbols
- if $p = 0$, the signature consists of only constantly many unary predicate symbols

Notation

$$\mathcal{L}_p^v e\mathrm{E}_\mathsf{a}$$

- $\mathcal{L}$, the *ground logic*, is the first-order logic with formal equality
- v, if given, bounds the number of allowed variables
- e, if given, specifies the number of builtin equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{global}, \mathsf{local}\}$, if given, specifies the agreement condition between the equivalence symbols
- if p is not given, the signature consists of arbitrary many unary and binary predicate symbols
- if $p = 0$, the signature consists of only constantly many unary predicate symbols
- if $p = 1$, the signature consists of arbitrary many unary predicate symbols

Examples

- $\mathcal{L}_0 1\mathrm{E}$ is the logic of a single equivalence
- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}^2 2\mathrm{E}_{\mathsf{local}}$ is the two-variable logic featuring unary and binary predicate symbols in addition to two builtin equivalence symbols in local agreement

Reduction Strategy

To reduce $\mathrm{(FIN\text{-})SAT\text{-}}\mathcal{L}e\mathrm{E}_{\mathsf{local}}$ to $\mathrm{(FIN\text{-})SAT\text{-}}\mathcal{L}e\mathrm{E}_{\mathsf{refine}}$,

- look at the levels
- encode a permutation witnessing the local agreement in a permutation setup
- define formulas that recover the original equivalences from the levels and the permutations
- not every combination of levels and permutations defines local agreement $\implies$ need constraint on permutations

Characteristic Permutations

Consider an $\mathcal{L}e\mathrm{E}_{\mathsf{local}}$-signature Σ containing $\mathrm{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_e \rangle$.
Let $\mathfrak{A}$ be a Σ-structure, $E_i = \boldsymbol{e}_i^{\mathfrak{A}}$ and $a \in A$.
The *characteristic permutation* ν at a is the antilexicographically smallest permutation of $[1, e]$ satisfying:

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

Collect the characteristic permutations in $\mathbf{[\Sigma{:}\mathrm{chperm}]}^{\mathfrak{A}} : A \to \mathbb{S}_e$.

Local Agreement of Permutations

Remark

Let $L_1, L_2, \ldots, L_e$ be the levels of E_i, $a, b \in A$, $\alpha = [\![\Sigma\text{:chperm}]\!]^{\mathfrak{A}} a$ and $\beta = [\![\Sigma\text{:chperm}]\!]^{\mathfrak{A}} b$.
If $(a, b) \in L_k$, then $\alpha(k) = \beta(k)$.
That is, if a and b are connected at level k, then their characteristic permutations agree at position k.

- This doesn't hold in general for any set of witnessing permutations

Levels and Permutations

Let $\mathrm{L} = \langle \boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_e \rangle + \mathrm{P}$ consist of the builtin equivalence symbols $\boldsymbol{I}_i$ (we intend to interpret them as the levels) together with the permutation setup P (intended to encode the characteristic permutations).
The formula

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

encodes the local agreement of permutations.

Recovering

The formulas

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

recover the original equivalences ($i \in [1, e]$).

Remark

Let $\mathfrak{A}$ be an L-structure satisfying $[\mathrm{P}\text{:perm}] \wedge [\mathrm{L}\text{:fixperm}]$ such that the level symbols $\boldsymbol{l}_i$ are interpreted as a sequence of equivalence relations in refinement. Let $L_i = \boldsymbol{l}_i^{\mathfrak{A}}$ and $E_i = [\mathrm{L}\text{:el-}i]^{\mathfrak{A}}$. Then E_i is a sequence of equivalence relations in local agreement and $L_k[a] = E_{\alpha(k)}[a]$ for any $a \in A$ and $\alpha = [\mathrm{P}\text{:data}]^{\mathfrak{A}}$.

Translation

Let $\Sigma' = \Sigma + \mathrm{L}$ and $\mathrm{L}' = \Sigma' - \mathrm{E}$. The translation $\mathsf{ltr}\,\varphi : \mathcal{L}[\Sigma] \to \mathcal{L}[\mathrm{L}']$ is defined by

$$\mathsf{ltr}\,\varphi = \varphi' \wedge [\mathrm{P}\textbf{:}\mathsf{perm}] \wedge [\mathrm{L}\textbf{:}\mathsf{fixperm}],$$

where φ' is obtained from φ by replacing all occurrences of $\boldsymbol{e}_i(x, y)$ by $[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i](x, y)$.

Remark

φ is (finitely) satisfiable over $\mathcal{L}e\mathrm{E}_{\mathsf{local}}$ iff $\mathsf{ltr}\,\varphi$ is (finitely) satisfiable over $\mathcal{L}e\mathrm{E}_{\mathsf{refine}}$.

- if $\mathfrak{A} \models \varphi$, interpret $\boldsymbol{l}_i$ as the levels and encode $[\Sigma\textbf{:}\mathrm{chperm}]^{\mathfrak{A}}$ in the permutation setup P.
- if $\mathfrak{A}' \models \mathsf{ltr}\,\varphi$, interpret $\boldsymbol{e}_i$ as $[\mathrm{L}\textbf{:}\mathsf{el}\text{-}i]$.

Translation

The translation just uses polynomially many new unary predicate symbols (it can "reuse" the builtin equivalences).

Proposition

- *the logic* $\mathcal{L}e\mathrm{E}_{\mathsf{local}}$ *has the finite model property iff the logic* $\mathcal{L}e\mathrm{E}_{\mathsf{refine}}$ *has the finite model property*
- *the corresponding satisfiability problems are polynomial-time equivalent*
- *also works for* $\mathcal{L}_1 e\mathrm{E}_{\mathsf{local}}$ *and* $\mathcal{L}^2 e\mathrm{E}_{\mathsf{local}}$

Monadic Logics

Results

It is known that:

- $\mathcal{L}_1$ has the finite model property [Löwenheim 1915] and its satisfiability problem is $\textsc{NExpTime}$-complete

Results

It is known that:

- $\mathcal{L}_1$ has the finite model property [Löwenheim 1915] and its satisfiability problem is $\mathrm{NE}\textsc{xp}\mathrm{T}\textsc{ime}$-complete
- $\mathcal{L}_0 1\mathrm{E}$ has the finite model property and its satisfiability problem is $\mathrm{PS}\textsc{pace}$-complete

Results

It is known that:

- $\mathcal{L}_1$ has the finite model property [Löwenheim 1915] and its satisfiability problem is $\mathrm{NE}\textsc{xp}\mathrm{T}\textsc{ime}$-complete
- $\mathcal{L}_0 1\mathrm{E}$ has the finite model property and its satisfiability problem is $\mathrm{PS}\textsc{pace}$-complete
- $\mathcal{L}_0 2\mathrm{E}$ lacks the finite model property and both the satisfiability and finite satisfiability are undecidable [Janiczak 1953]

Results

It is known that:

- $\mathcal{L}_1$ has the finite model property [Löwenheim 1915] and its satisfiability problem is $\mathrm{NE}\textsc{xp}\mathrm{T}\textsc{ime}$-complete
- $\mathcal{L}_0 1\mathrm{E}$ has the finite model property and its satisfiability problem is $\mathrm{PS}\textsc{pace}$-complete
- $\mathcal{L}_0 2\mathrm{E}$ lacks the finite model property and both the satisfiability and finite satisfiability are undecidable [Janiczak 1953]

How about $\mathcal{L}_1 1\mathrm{E}$?

Results

It is known that:

- $\mathcal{L}_1$ has the finite model property [Löwenheim 1915] and its satisfiability problem is $\mathrm{NExpTime}$-complete
- $\mathcal{L}_0 1\mathrm{E}$ has the finite model property and its satisfiability problem is PSpace-complete
- $\mathcal{L}_0 2\mathrm{E}$ lacks the finite model property and both the satisfiability and finite satisfiability are undecidable [Janiczak 1953]

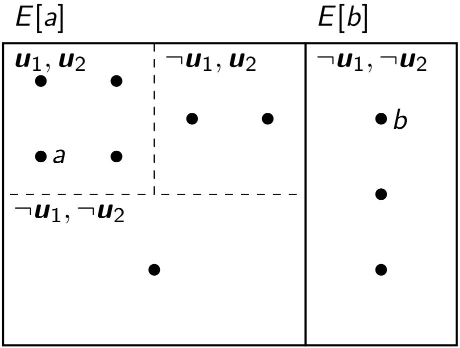
How about $\mathcal{L}_1 1\mathrm{E}$? **We show that:**

- **$\mathcal{L}_1 1\mathrm{E}$ has the finite model property and its satisfiability problem is $\mathrm{N2ExpTime}$-complete**
- **in general, $\mathcal{L}_1 e\mathrm{E}_{\text{refine}}$ has the finite model property and its satisfiability problem is $\mathrm{N}(e+1)\mathrm{ExpTime}$-complete**

Complexity: Cells

Let $\Sigma = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{e} \rangle$ and $\mathfrak{A}$ be a Σ-structure.
A *cell* $C \subseteq A$ is a maximal set of $\boldsymbol{e}$-equivalent elements satisfying the same $\boldsymbol{u}$-predicates.

Example $\Sigma = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{e} \rangle$



Small Cells

Lemma

Let $r \geq 1$ and suppose that $\mathfrak{A}$ is a Σ-structure. Then there is a substructure $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and every $\mathfrak{B}$-cell has cardinality at most r.

Proof Idea.

For every $\mathfrak{A}$-cell, if it has less than r elements select them all, otherwise select any r elements. Consider $\mathfrak{B}$ induced by the selected elements. Win the *r-round Ehrenfeucht-Fraïssé game* as Duplicator: if the challenge is new, choose a new selected element from the same cell. Since the game lasts r rounds, you'll never run out of selected elements. $\square$

Few Isomorphic Classes

Lemma

Let $r \geq 1$ and suppose that $\mathfrak{A}$ is a Σ-structure. Then there is a substructure $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and $\mathfrak{B}$-class is isomorphic to at most $(r-1)$ other $\mathfrak{B}$-classes.

Combining these and doing the math, we get:

Remark

Let $\mathfrak{A}$ be a $\Sigma(u,1)$-structure and let $r \geq 1$. There is some $\mathfrak{B} \subseteq \mathfrak{A}$ such that $\mathfrak{B} \equiv_r \mathfrak{A}$ and $|B| \leq r^2 2^u((r+1)^{2^u} - 1)$.

This is doubly exponential with respect to the size of φ, hence $(\mathrm{FIN\text{-}})\mathrm{SAT}\text{-}\mathcal{L}_1 1\mathrm{E}$ is in $\mathrm{N2E}\textsc{xp}\mathrm{T}\textsc{ime}$.

Hardness: Domino Problem

- Reduce the $\mathrm{N2E{\scriptsize XP}T{\scriptsize IME}}$-complete Square Domino Tiling Problem to $(\mathrm{FIN\text{-}})\mathrm{SAT}\text{-}\mathcal{L}_1 1\mathrm{E}$.
- A *domino system* is a triple $D = (T, H, V)$, where $T = [1, k]$ is a set of *tiles* and $H, V \subseteq T \times T$ are the *horizontal* and *vertical matching relations*.
- A *tiling* of the $m \times m$ square for a domino system D with initial condition $c^0 = \langle t_1^0, t_2^0, \ldots, t_n^0 \rangle$, where $n \leq m$, is a mapping $t : [1, m] \times [1, m] \to T$ such that:
  - $(t(i,j), t(i+1,j)) \in H$ for all $i \in [1, m-1], j \in [1, m]$
  - $(t(i,j), t(i,j+1)) \in V$ for all $i \in [1, m], j \in [1, m-1]$
  - $t(i, 1) = t_i^0$ for all $i \in [1, n]$

Domino Problem

Example $T = [1, 3]$,
$H = \{(1,3),(2,1),(2,2)\}$, $V = \{(2,2),(3,2),(1,2)\}$

2	2
1	3

Theorem

There is a domino system D_0 such that the problem of asking if there exists a tiling for D_0 with initial condition c_0 of length n for the $2^{2^n} \times 2^{2^n}$-square is N2EXPTIME*-complete.*

Hardness

Main issue: given $\Sigma = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{e} \rangle$, how can we define a doubly exponential grid?

Hardness

Main issue: given $\Sigma = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_u, \boldsymbol{e} \rangle$, how can we define a doubly exponential grid?

- Each class can contain exponentially many cells
- If we encode bits in cells, the classes encode doubly exponential numbers

Encoding

- we can ensure that every class contains maximally many 2^u cells
- a cell containing a single element encodes bit 0:

$$[\Sigma\text{:bit-0}](\boldsymbol{x}) = \forall \boldsymbol{y}\, (\boldsymbol{e}(\boldsymbol{y}, \boldsymbol{x}) \wedge [\Sigma\text{:pos-eq}](\boldsymbol{y}, \boldsymbol{x}) \rightarrow \boldsymbol{y} = \boldsymbol{x})$$

- a cell containing more elements encodes bit 1:

$$[\Sigma\text{:bit-1}](\boldsymbol{x}) = \exists \boldsymbol{y}\, (\boldsymbol{e}(\boldsymbol{y}, \boldsymbol{x}) \wedge [\Sigma\text{:pos-eq}](\boldsymbol{y}, \boldsymbol{x}) \wedge \boldsymbol{y} \neq \boldsymbol{x})$$

Data

Let $\mathfrak{A}$ be a Σ-structure and let $E = \boldsymbol{e}^{\mathfrak{A}}$. The equivalence classes of E are $\mathscr{E}E$. The number encoded by the bitstring b is $\underline{\mathrm{b}}$.
With *a bit of work* we can define:

- $[\Sigma\text{:}\mathrm{Data}]^{\mathfrak{A}} : \mathscr{E}E \to \mathbb{B}^{2^u}$ that assigns exponential bitstrings (hence doubly exponential numbers) to the classes of $\mathfrak{A}$
- $\mathfrak{A} \models [\Sigma\text{:Zero}](a)$ iff $\underline{[\Sigma\text{:}\mathrm{Data}]^{\mathfrak{A}} E[a]} = 0$
- $\mathfrak{A} \models [\Sigma\text{:Succ}](a, b)$ iff $\underline{[\Sigma\text{:}\mathrm{Data}]^{\mathfrak{A}} E[b]} = 1 + \underline{[\Sigma\text{:}\mathrm{Data}]^{\mathfrak{A}} E[a]}$
- etc.

Reduction

Given $D_0 = (T, V, H)$, where $T = [1, k]$, and $c^0 = \langle t_1^0, t_1^0, \ldots, t_n^0 \rangle$, consider:

$$\Sigma = \left\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H; \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V; \boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_k^T; \boldsymbol{e} \right\rangle$$

$\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_k^T$ for tiles

$$\Sigma^{HV} = \left\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H; \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V; \boldsymbol{e} \right\rangle \text{ for the full grid}$$

$$\Sigma^{H} = \left\langle \boldsymbol{u}_1^H, \boldsymbol{u}_2^H, \ldots, \boldsymbol{u}_n^H; \boldsymbol{e} \right\rangle \text{ for horizontal matching}$$

$$\Sigma^{V} = \left\langle \boldsymbol{u}_1^V, \boldsymbol{u}_2^V, \ldots, \boldsymbol{u}_n^V; \boldsymbol{e} \right\rangle \text{ for vertical matching}$$

Reduction

small formulas

- $[\Sigma^{HV}$:pos-full$] \wedge [\Sigma^{HV}$:Full$] \wedge [\Sigma^{HV}$:Alldiff$]$ defines a full doubly exponential grid
- $\forall \boldsymbol{x} \left(\bigvee_{1 \leq i \leq k} \left(\boldsymbol{u}_i^T(\boldsymbol{x}) \wedge \bigwedge_{j \in [1,k] \setminus \{i\}} \neg \boldsymbol{u}_j^T(\boldsymbol{x}) \right) \right)$ asserts that every element has a unique type
- $\forall \boldsymbol{x} \forall \boldsymbol{y} \Big(\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \bigwedge_{1 \leq i \leq k} (\boldsymbol{u}_i^T(\boldsymbol{x}) \leftrightarrow \boldsymbol{u}_i^T(\boldsymbol{x})) \Big)$ asserts that the type is the same in each class

Reduction

small formulas

- $\forall \boldsymbol{x}\Big(\textbf{[}\mathrm{D}^{H}\textbf{:}\text{Eq-}(j-1)\textbf{]}(\boldsymbol{x}) \wedge \textbf{[}\mathrm{D}^{V}\textbf{:}\text{Zero}\textbf{]}(\boldsymbol{x}) \rightarrow \boldsymbol{u}^{T}_{t^0_j}(\boldsymbol{x})\Big)$ encodes the inital condition
- $\forall \boldsymbol{x} \forall \boldsymbol{y}\Big(\textbf{[}\mathrm{D}^{H}\textbf{:}\text{Succ}\textbf{]}(\boldsymbol{x}, \boldsymbol{y}) \wedge \textbf{[}\mathrm{D}^{V}\textbf{:}\text{Eq}\textbf{]}(\boldsymbol{x}, \boldsymbol{y}) \rightarrow$ $\bigvee_{(i,j)\in H} \boldsymbol{u}^{T}_{i}(\boldsymbol{x}) \wedge \boldsymbol{u}^{T}_{j}(\boldsymbol{y})\Big)$ encodes the horizontal tiling condition

Summary

- this shows that the satisfiability for $\mathcal{L}_1 1\mathrm{E}$ is $\mathrm{N2E{\small XP}T{\small IME}}$-complete
- we generalize to show the satisfiability for $\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$ is $\mathrm{N}(e+1)\mathrm{E{\small XP}T{\small IME}}$-complete