Satisfiability with Equivalences in Agreement

Krasimir Georgiev

September 12, 2016

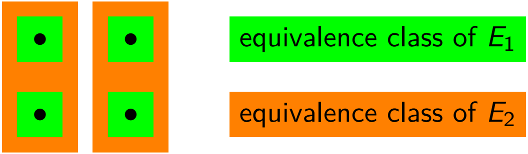
Agreement

A sequence $E_1, E_2, \ldots, E_e$ of equivalence relations on A is in:

Agreement

A sequence $E_1, E_2, \ldots, E_e$ of equivalence relations on A is in:

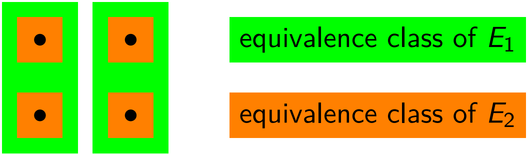
- *refinement* if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_e$
- *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(e)}$ for some permutation ν of $[1, e]$
- *local agreement* if the equivalence classes of any point form a chain, that is for any $a \in A$ there is some permutation ν of $[1, e]$ such that $E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \cdots \subseteq E_{\nu(e)}$



Agreement

A sequence $E_1, E_2, \ldots, E_e$ of equivalence relations on A is in:

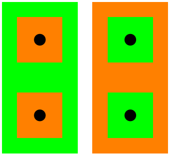
- *refinement* if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_e$
- *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(e)}$ for some permutation ν of $[1, e]$
- *local agreement* if the equivalence classes of any point form a chain, that is for any $a \in A$ there is some permutation ν of $[1, e]$ such that $E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \cdots \subseteq E_{\nu(e)}$



Agreement

A sequence $E_1, E_2, \ldots, E_e$ of equivalence relations on A is in:

- *refinement* if $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_e$
- *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \cdots \subseteq E_{\nu(e)}$ for some permutation ν of $[1, e]$
- *local agreement* if the equivalence classes of any point form a chain, that is for any $a \in A$ there is some permutation ν of $[1, e]$ such that $E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \cdots \subseteq E_{\nu(e)}$



equivalence class of E_1

equivalence class of E_2

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_\mathsf{local}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\text{refine}, \text{local}, \text{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_{\text{local}}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_{\mathsf{local}}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_{\mathsf{local}}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_\mathsf{local}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\text{refine}, \text{local}, \text{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_\mathsf{local}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_\mathsf{local}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\mathsf{refine}, \mathsf{local}, \mathsf{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_\mathsf{local}$ is the two-variable fragment of two equivalences in local agreement

Notation

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

- $\mathcal{L}$ is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- v bounds the number of variables
- e specifies the number of built-in equivalence symbols
- $\mathsf{a} \in \{\text{refine}, \text{local}, \text{global}\}$ specifies an agreement condition
- if $p = 0$ only constantly many additional unary predicate symbols are allowed
- if $p = 1$ only additional unary predicate symbols are allowed

Examples

- $\mathcal{L}_1$ is the monadic fragment
- $\mathcal{L}_0 1\mathrm{E}$ is the fragment of a single equivalence
- $\mathcal{L}^2 2\mathrm{E}_{\mathsf{local}}$ is the two-variable fragment of two equivalences in local agreement

Goal

In this work, we investigate the ***computational complexity*** of the ***satisfiability*** for the ***monadic*** and the ***two-variable fragment*** in the presence of ***equivalences in agreement***.

Reductions

Refinement is the easiest condition to work with, so we define polynomial time reductions to its satisfiability problem:

$$\mathrm{SAT}\text{-}\mathcal{L}_p^v e\mathrm{E}_{\mathsf{global}} \leq_{\mathrm{m}}^{\mathrm{PT{\scriptsize IME}}} \mathrm{SAT}\text{-}\mathcal{L}_p^v e\mathrm{E}_{\mathsf{refine}}$$
$$\mathrm{SAT}\text{-}\mathcal{L}_p^v e\mathrm{E}_{\mathsf{local}} \leq_{\mathrm{m}}^{\mathrm{PT{\scriptsize IME}}} \mathrm{SAT}\text{-}\mathcal{L}_p^v e\mathrm{E}_{\mathsf{refine}}.$$

Central to these reductions is the notion of ***levels***.

Levels

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

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

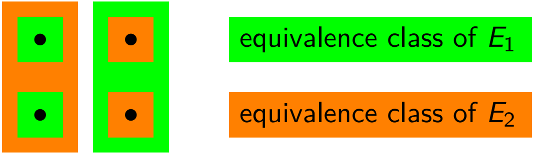
Levels

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

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

Remark

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



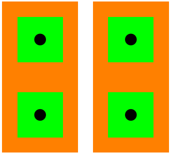
Levels

The *level sequence* $L_1, L_2, \dots, L_e$ of a sequence $E_1, E_2, \dots, E_e$ of equivalence relations on A in local agreement is defined by:

$$L_m = \bigcap \{ E_{i_1} \cup E_{i_2} \cup \dots \cup E_{i_m} \mid 1 \leq i_1 < i_2 < \dots < i_m \leq e \}.$$

Remark

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



equivalence class of L_1

equivalence class of L_2

Monadic fragments

It is known that

- $\mathrm{SAT}\text{-}\mathcal{L}_1$ is in $\mathrm{NExpTime}$ [Löwenheim 1915] and is $\mathrm{NExpTime}$-hard
- $\mathrm{SAT}\text{-}\mathcal{L}_0\mathrm{1E}$ is PSpace-complete
- $\mathrm{SAT}\text{-}\mathcal{L}_0\mathrm{2E}$ is undecidable [Janiczak 1953]

Monadic fragments

It is known that

- $\mathrm{SAT}\text{-}\mathcal{L}_1$ is in $\mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$ [Löwenheim 1915] and is $\mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$-hard
- $\mathrm{SAT}\text{-}\mathcal{L}_0 1\mathrm{E}$ is $\mathrm{PS{\scriptsize PACE}}$-complete
- $\mathrm{SAT}\text{-}\mathcal{L}_0 2\mathrm{E}$ is undecidable [Janiczak 1953]

We show that

- $\mathrm{SAT}\text{-}\mathcal{L}_1 1\mathrm{E}$ is $\mathrm{N2E{\scriptsize XP}T{\scriptsize IME}}$-complete
- In general, $\mathrm{SAT}\text{-}\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$ is $\mathrm{N}(e+1)\mathrm{E{\scriptsize XP}T{\scriptsize IME}}$-complete

Monadic fragments

We show that

- $\mathrm{SAT}\text{-}\mathcal{L}_1 1\mathrm{E}$ is $\mathrm{N2}\textsc{ExpTime}$-complete
- In general, $\mathrm{SAT}\text{-}\mathcal{L}_1 e\mathrm{E}_{\mathsf{refine}}$ is $\mathrm{N}(e+1)\textsc{ExpTime}$-complete

Approach: "work at the quantifier rank level of abstraction"

- for the upper bound, we use Ehrenfeucht-Fraïssé games to bound the size of a minimal model of a satisfiable formula
- for hardness, we reduce a version of the domino tiling problem to satisfiability

Two-variable fragments

Only the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are allowed. It is known that:

- SAT-$\mathcal{L}^2$ is NExpTime-complete [Grädel, Kolaitis, Vardi, 1997]
- SAT-$\mathcal{L}^2$1E is NExpTime-complete [Kieroński, 2005]
- SAT-$\mathcal{L}^2$2E is N2ExpTime-complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014]
- SAT-$\mathcal{L}^2$3E is undecidable [Kieroński, Otto, 2005]

Two-variable fragments

Only the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are allowed. It is known that:

- $\mathrm{SAT}\text{-}\mathcal{L}^2$ is $\mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$-complete [Grädel, Kolaitis, Vardi, 1997]
- $\mathrm{SAT}\text{-}\mathcal{L}^2 1\mathrm{E}$ is $\mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$-complete [Kieroński, 2005]
- $\mathrm{SAT}\text{-}\mathcal{L}^2 2\mathrm{E}$ is $\mathrm{N2E{\scriptsize XP}T{\scriptsize IME}}$-complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014]
- $\mathrm{SAT}\text{-}\mathcal{L}^2 3\mathrm{E}$ is undecidable [Kieroński, Otto, 2005]

We show that

- $\mathrm{SAT}\text{-}\mathcal{L}^2 e\mathrm{E}_{\text{refine}}$ is in $\mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$

Approach: "work at the level of abstraction of types"

Types

- A 1-type π is a maximal consistent set of literals featuring only the variable $\boldsymbol{x}$
- A 2-type τ is a maximal consistent set of literals featuring the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ that contains $(\boldsymbol{x} \neq \boldsymbol{y})$

$$\begin{tikzcd} \mathrm{tp}_{\boldsymbol{x}}\tau \bullet \arrow[r, dash, "\tau", "\tau^{-1}"'] & \bullet\, \mathrm{tp}_{\boldsymbol{y}}\tau \end{tikzcd}$$

- If τ is a 2-type, $\mathrm{tp}_{\boldsymbol{x}}\tau$ is the one-type consisting of the literals from τ featuring only the variable $\boldsymbol{x}$,
- τ^{-1} is the two-type obtained by swapping $\boldsymbol{x}$ and $\boldsymbol{y}$ in the literals of τ
- and $\mathrm{tp}_{\boldsymbol{y}}\tau = \mathrm{tp}_{\boldsymbol{x}}\tau^{-1}$.

Classified signatures

A *classified signature* $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ consists of a predicate signature Σ together with a sequence $\bar{\boldsymbol{m}} = \boldsymbol{m}_1 \boldsymbol{m}_2 \ldots \boldsymbol{m}_m$ of distinct binary predicate symbols from Σ.
We say that the Σ-structure $\mathfrak{A}$ is a structure for $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ if

$$\mathfrak{A} \models \bigwedge_{1 \leq i \leq m} \forall \boldsymbol{x} \exists \boldsymbol{y} (\boldsymbol{m}_i(\boldsymbol{x}, \boldsymbol{y}) \wedge (\boldsymbol{x} \neq \boldsymbol{y}))$$

Type instances

- A *type instance* T over the classified signature $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ is a nonempty set of 2-types that is closed under inversion.
- The type instance of a structure $\mathfrak{A}$ for $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ is defined by

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

- The *type realizability problem* is: given a classified signature $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$ and a type instance T over $\langle \Sigma, \bar{\boldsymbol{m}} \rangle$, is T the type instance of structure?

Approach

- reduce the satisfiability problem for $\mathcal{L}^2 e\mathrm{E}_{\text{refine}}$ to an appropriate type realizability problem in nondeterministic exponential time
- show that the type realizability problem can be decided in nondeterministic polynomial time
- we do this by defining *star-types* to "witness the local environment" around elements in models
- collecting them in "not too big" *certificates* and
- constructing a model from a given certificate