

Satisfiability with Equivalences in Agreement

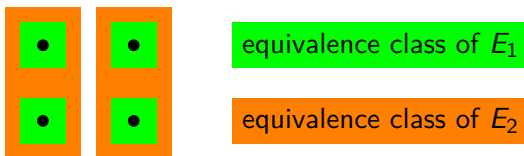
Krasimir Georgiev

September 12, 2016

Agreement

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

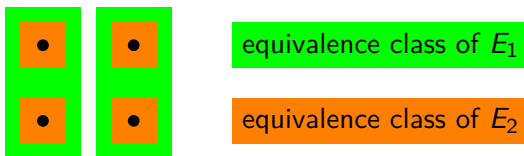
- ▶ *refinement* if $E_1 \subseteq E_2 \subseteq \dots \subseteq E_e$
- ▶ *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \dots \subseteq E_{\nu(e)}$ for some permutation ν of $[1, e]$
- ▶ *local agreement* if the equivalence classes of any point form a chain,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



Agreement

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

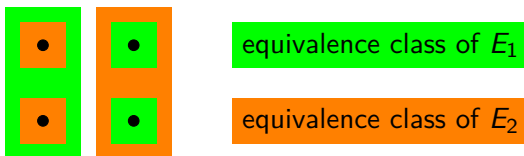
- ▶ *refinement* if $E_1 \subseteq E_2 \subseteq \dots \subseteq E_e$
- ▶ *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \dots \subseteq E_{\nu(e)}$ for some permutation ν of $[1, e]$
- ▶ *local agreement* if the equivalence classes of any point form a chain,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



Agreement

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

- ▶ *refinement* if $E_1 \subseteq E_2 \subseteq \dots \subseteq E_e$
- ▶ *global agreement* if it forms a chain, that is $E_{\nu(1)} \subseteq E_{\nu(2)} \subseteq \dots \subseteq E_{\nu(e)}$ 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 $E_{\nu(1)}[a] \subseteq E_{\nu(2)}[a] \subseteq \dots \subseteq E_{\nu(e)}[a]$



Notation

$$\mathcal{L}_p^{\vee e} E_a$$

- ▶ \mathcal{L} is the first-order predicate logic with equality featuring only unary and binary predicate symbols

Notation

$$\mathcal{L}_p^v e E_a$$

- ▶ \mathcal{L} is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- ▶ v bounds the number of variables

Notation

$$\mathcal{L}_p^{\nu e} E_a$$

- ▶ \mathcal{L} is the first-order predicate logic with equality featuring only unary and binary predicate symbols
- ▶ ν bounds the number of variables
- ▶ e specifies the number of built-in equivalence symbols

Notation

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ specifies an agreement condition

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ specifies an agreement condition
- ▶ if $p = 0$ only constantly many additional unary predicate symbols are allowed

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ 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

Notation

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ 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

Notation

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ 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
- ▶ $\mathcal{L}_0 1E$ is the fragment of a single equivalence

$$\mathcal{L}_p^v e E_a$$

- ▶ \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
- ▶ $a \in \{\text{refine}, \text{local}, \text{global}\}$ 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
- ▶ $\mathcal{L}_0 1E$ is the fragment of a single equivalence
- ▶ $\mathcal{L}^2 2E_{\text{local}}$ is the two-variable fragment of two equivalences in local agreement

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*.

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

$$\begin{aligned}\text{SAT-}\mathcal{L}_p^v\text{eE}_{\text{global}} &\leq_m^{\text{PTIME}} \text{SAT-}\mathcal{L}_p^v\text{eE}_{\text{refine}} \\ \text{SAT-}\mathcal{L}_p^v\text{eE}_{\text{local}} &\leq_m^{\text{PTIME}} \text{SAT-}\mathcal{L}_p^v\text{eE}_{\text{refine}}.\end{aligned}$$

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

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:

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

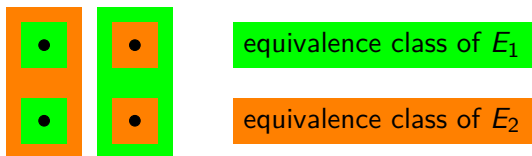
Levels

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.



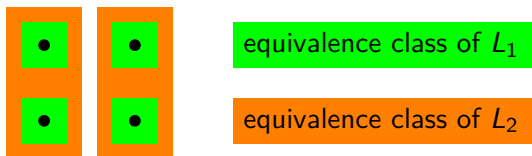
Levels

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.



Monadic fragments

It is known that

- ▶ $\text{SAT-}\mathcal{L}_1$ is in NEXPTIME [Löwenheim 1915] and is NEXPTIME-hard
- ▶ $\text{SAT-}\mathcal{L}_{01E}$ is PSPACE-complete
- ▶ $\text{SAT-}\mathcal{L}_{02E}$ is undecidable [Janiczak 1953]

Monadic fragments

It is known that

- ▶ $\text{SAT-}\mathcal{L}_1$ is in NEXPTIME [Löwenheim 1915] and is NEXPTIME-hard
- ▶ $\text{SAT-}\mathcal{L}_01E$ is PSPACE-complete
- ▶ $\text{SAT-}\mathcal{L}_02E$ is undecidable [Janiczak 1953]

We show that

- ▶ $\text{SAT-}\mathcal{L}_11E$ is $\text{N2EXPTIME-complete}$
- ▶ In general, $\text{SAT-}\mathcal{L}_1eE_{\text{refine}}$ is $\text{N}(e+1)\text{EXPTIME-complete}$

We show that

- ▶ $\text{SAT-}\mathcal{L}_11E$ is $\text{N}2\text{EXP}\text{TIME}$ -complete
- ▶ In general, $\text{SAT-}\mathcal{L}_1eE_{\text{refine}}$ is $\text{N}(e + 1)\text{EXP}\text{TIME}$ -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 x and y are allowed. It is known that:

- ▶ $\text{SAT-}\mathcal{L}^2$ is NEXPTIME -complete [Grädel, Kolaitis, Vardi, 1997]
- ▶ $\text{SAT-}\mathcal{L}^{21\text{E}}$ is NEXPTIME -complete [Kieroński, 2005]
- ▶ $\text{SAT-}\mathcal{L}^{22\text{E}}$ is $\text{N}^2\text{EXPTIME}$ -complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014]
- ▶ $\text{SAT-}\mathcal{L}^{23\text{E}}$ is undecidable [Kieroński, Otto, 2005]

Two-variable fragments

Only the variables x and y are allowed. It is known that:

- ▶ $\text{SAT-}\mathcal{L}^2$ is NEXPTIME -complete [Grädel, Kolaitis, Vardi, 1997]
- ▶ $\text{SAT-}\mathcal{L}^{21\text{E}}$ is NEXPTIME -complete [Kieroński, 2005]
- ▶ $\text{SAT-}\mathcal{L}^{22\text{E}}$ is $\text{N}^2\text{EXPTIME}$ -complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014]
- ▶ $\text{SAT-}\mathcal{L}^{23\text{E}}$ is undecidable [Kieroński, Otto, 2005]

We show that

- ▶ $\text{SAT-}\mathcal{L}^{2\text{eE}}_{\text{refine}}$ is in NEXPTIME

Approach: “work at the level of abstraction of types”

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

$$\text{tp}_{\mathbf{x}}\tau \bullet \xrightarrow[\tau^{-1}]{\tau} \bullet \text{tp}_{\mathbf{y}}\tau$$

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

Classified signatures

A *classified signature* $\langle \Sigma, \bar{\mathbf{m}} \rangle$ consists of a predicate signature Σ together with a nonempty sequence $\bar{\mathbf{m}} = \mathbf{m}_1 \mathbf{m}_2 \dots \mathbf{m}_m$ of distinct binary predicate symbols from Σ .

We say that the Σ -structure \mathfrak{A} is a structure for $\langle \Sigma, \bar{\mathbf{m}} \rangle$ if

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

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

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

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

Approach

- ▶ reduce the satisfiability problem for $\mathcal{L}^2\mathbf{eE}_{\text{refine}}$ 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