

Satisfiability with Equivalences in Agreement, Part 2

Krasimir Georgiev

September 1, 2016

Two-variable Logic

- ▶ The two-variable logic \mathcal{L}^2 is the fragment of first-order logic featuring only the variables \mathbf{x} and \mathbf{y} (with formal equality and restricted to just unary and binary predicate symbols).
- ▶ “resource restriction”
- ▶ connections with modal logic

Satisfiability Results

- ▶ \mathcal{L}^2 has the finite model property [Mortimer, 1975] and its (finite) satisfiability problem is NEXPTIME-complete [Grädel, Kolaitis, Vardi, 1997].

Satisfiability Results

- ▶ \mathcal{L}^2 has the finite model property [Mortimer, 1975] and its (finite) satisfiability problem is NEXPTIME-complete [Grädel, Kolaitis, Vardi, 1997].
- ▶ $\mathcal{L}^2\text{1E}$ has the finite model property and its satisfiability problem is NEXPTIME-complete [Kieroński, 2005].

Satisfiability Results

- ▶ \mathcal{L}^2 has the finite model property [Mortimer, 1975] and its (finite) satisfiability problem is NEXPTIME-complete [Grädel, Kolaitis, Vardi, 1997].
- ▶ $\mathcal{L}^2\text{1E}$ has the finite model property and its satisfiability problem is NEXPTIME-complete [Kieroński, 2005].
- ▶ $\mathcal{L}^2\text{2E}$ lacks the finite model property and both its satisfiability and finite satisfiability are N2EXPTIME-complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014],

Satisfiability Results

- ▶ \mathcal{L}^2 has the finite model property [Mortimer, 1975] and its (finite) satisfiability problem is NEXPTIME-complete [Grädel, Kolaitis, Vardi, 1997].
- ▶ $\mathcal{L}^2 1E$ has the finite model property and its satisfiability problem is NEXPTIME-complete [Kieroński, 2005].
- ▶ $\mathcal{L}^2 2E$ lacks the finite model property and both its satisfiability and finite satisfiability are N2EXPTIME-complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014],
- ▶ Both the satisfiability and finite satisfiability problems for $\mathcal{L}^2 eE$ are undecidable for $e \geq 3$ [Kieroński, Otto, 2005].

Satisfiability Results

- ▶ \mathcal{L}^2 has the finite model property [Mortimer, 1975] and its (finite) satisfiability problem is NEXPTIME-complete [Grädel, Kolaitis, Vardi, 1997].
- ▶ $\mathcal{L}^2 1E$ has the finite model property and its satisfiability problem is NEXPTIME-complete [Kieroński, 2005].
- ▶ $\mathcal{L}^2 2E$ lacks the finite model property and both its satisfiability and finite satisfiability are N2EXPTIME-complete [Kieroński, Michaliszyn, Pratt-Hartmann, Tendera, 2014],
- ▶ Both the satisfiability and finite satisfiability problems for $\mathcal{L}^2 eE$ are undecidable for $e \geq 3$ [Kieroński, Otto, 2005].
- ▶ **We show that $\mathcal{L}^2 eE_{\text{refine}}$ has the finite model property and its satisfiability problem is in NEXPTIME for $e \geq 1$.**

Overview

Two-variable Logic

Types

Type Realizability

Type Realizability with Equivalences in Refinement

Scott Normal Form

Theorem (Scott, 1962)

There is a polynomial-time reduction $\text{sctr} : \mathcal{L}^2 \rightarrow \mathcal{L}^2$ which reduces every sentence φ 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 $m \geq 1$, the formulas α_i are quantifier-free and may use fresh unary predicate symbols, such that φ and $\text{sctr } \varphi$ are satisfiable over the same domains of cardinality at least 2.

Scott Normal Form

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

Idea: if ψ is a subformula of φ of the form $Qx\alpha(x, y)$, where α is quantifier-free, then

$$\varphi \text{ and } \varphi' \wedge \forall y (\mathbf{p}(y) \leftrightarrow Qx\alpha(x, y))$$

are satisfiable over the same domains, where \mathbf{p} is a new unary symbol and φ' is obtained from φ by replacing the subformula ψ by $\mathbf{p}(y)$.

Classified Signatures

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

- ▶ replace $\alpha_i, i \in [1, m]$ with fresh binary predicate *message* symbols \mathbf{m}_i with interpretation $\forall \mathbf{x} \forall \mathbf{y} (\mathbf{m}_i(\mathbf{x}, \mathbf{y}) \leftrightarrow \alpha_i(\mathbf{x}, \mathbf{y}))$
- ▶ A *classified signature* $\langle \Sigma, \bar{\mathbf{m}} \rangle$ consists of a signature Σ together with a sequence of distinct binary predicate symbols $\bar{\mathbf{m}} = \mathbf{m}_1 \mathbf{m}_2 \dots \mathbf{m}_m$ from Σ .
- ▶ A *structure* \mathfrak{A} for $\langle \Sigma, \bar{\mathbf{m}} \rangle$ is a structure for Σ satisfying:

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

Classified Signatures

- ▶ The (finite) *classified satisfiability problem* is: given a classified signature $\langle \Sigma, \bar{\mathbf{m}} \rangle$ and a quantifier-free formula $\alpha(\mathbf{x}, \mathbf{y})$, is there a $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure \mathfrak{A} such that $\mathfrak{A} \models \forall \mathbf{x} \forall \mathbf{y} \alpha(\mathbf{x}, \mathbf{y})$.
- ▶ Scott tells us how (finite) satisfiability reduces to (finite) classified satisfiability.

Types

Let $\Sigma = \langle \mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^n \rangle$ be a predicate signature.

- ▶ A 1-type π over Σ is a maximal consistent set of literals featuring only the variable \mathbf{x} (in model theory known as atomic type).
- ▶ A 2-type τ over Σ is maximal consistent set of literals featuring the variable symbols \mathbf{x} and \mathbf{y} and including $(\mathbf{x} \neq \mathbf{y})$.

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

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

Type Instances

- ▶ A *type instance* T over $\langle \Sigma, \bar{m} \rangle$ is a nonempty set of 2-types that is closed under inversion.
- ▶ The type instance $T[\mathfrak{A}]$ of a $\langle \Sigma, \bar{m} \rangle$ -structure \mathfrak{A} is:

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

where $\text{tp}^{\mathfrak{A}}[a, b]$ is the 2-type realized by (a, b) in \mathfrak{A} .

- ▶ Type instances are typically exponentially bigger than the classified signature.

Type Realizability

- ▶ The *(finite) type realizability problem* is: given a classified signature $\langle \Sigma, \bar{\mathbf{m}} \rangle$ and a type instance T over $\langle \Sigma, \bar{\mathbf{m}} \rangle$, is there a (finite) $\langle \Sigma, \bar{\mathbf{m}} \rangle$ -structure \mathfrak{A} such that $T[\mathfrak{A}] = T$.
- ▶ We aim to show that the type realizability problem for \mathcal{L}^2 is in NPTIME .
- ▶ (Finite) classified satisfiability reduces in nondeterministic exponential time to (finite) type realizability: just guess a type instance consisting of 2-types consistent with $\alpha(x, y)$.

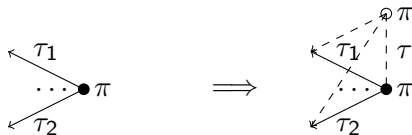
Kings and Workers

- ▶ Let T be a type instance over $\langle \Sigma, \bar{m} \rangle$. The 1-types of T are $\Pi_T = (\text{tp}_x \upharpoonright T)$.
- ▶ A 1-type $\kappa \in \Pi_T$ is a *king type* if no $\tau \in T$ connects κ with itself. The set of king types is K_T .
- ▶ The remaining 1-types are the *worker types*.
- ▶ In models of T , king types are realized uniquely, while worker types are realized by at least two elements.

Worker Copies

If \mathfrak{A} is a model for T , any worker element can be copied:

$$\pi \in W_T, \tau \in T, \text{tp}_x \tau = \text{tp}_y \tau = \pi$$



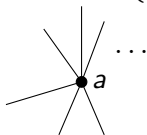
This doesn't work for kings!

$$T[\pi] = \begin{cases} \Pi_T & \text{if } \pi \text{ is a worker type} \\ \Pi_T \setminus \{\pi\} & \text{otherwise, if it is a king type} \end{cases}$$

Star-types

The *star-type* of $a \in A$ is:

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



Gives reason why the existential condition is satisfied locally at a .



image © 2014 Asia Citro, reproduced with permission from

<http://www.funathomewithkids.com/2014/07/magic-foaming-treasure-stars.html>

Star-types

A *star-type* σ over T is a nonempty subset $\sigma \subseteq T$ satisfying:

(σx) If $\tau, \tau' \in \sigma$, then $\text{tp}_x \tau = \text{tp}_x \tau'$. Denote $\text{tp}_x \tau$ for any $\tau \in \sigma$ by $\pi = \text{tp}_x \sigma$.

$(\sigma \pi y)$ If $\pi' \in T[\pi]$, then some $\tau \in \sigma$ has $\text{tp}_y \tau = \pi'$.

$(\sigma \kappa y)$ If $\kappa' \in T[\pi] \cap K_T$ and if $\tau, \tau' \in \sigma$ have $\text{tp}_y \tau = \text{tp}_y \tau' = \kappa'$, then $\tau = \tau'$.

(σm) If $m \in \bar{m}$, then some $\tau \in \sigma$ has $m(x, y) \in \tau$.

Then:

$(\sigma \kappa y')$ If $\kappa' \in T[\pi] \cap K_T$, then a unique $\tau \in \sigma$ has $\text{tp}_y \tau = \kappa$.

Certificates

A *certificate* \mathcal{S} for \mathbb{T} is a nonempty set of star-types satisfying:

- $(\mathcal{S}\tau)$ If $\tau \in \mathbb{T}$, then some $\sigma \in \mathcal{S}$ has $\tau \in \sigma$, that is there is a star-type containing each 2-type.
- $(\mathcal{S}\kappa)$ If $\kappa \in K_{\mathbb{T}}$ and if $\sigma, \sigma' \in \mathcal{S}$ have $\text{tp}_{\mathbf{x}}\sigma = \text{tp}_{\mathbf{x}}\sigma' = \kappa$, then $\sigma = \sigma'$.

Then:

- $(\mathcal{S}\pi)$ If $\pi \in \Pi_{\mathbb{T}}$, then some $\sigma \in \mathcal{S}$ has $\text{tp}_{\mathbf{x}}\sigma = \pi$.
- $(\mathcal{S}\kappa')$ If $\kappa \in K_{\mathbb{T}}$, then a unique $\sigma \in \mathcal{S}$ has $\text{tp}_{\mathbf{x}}\sigma = \pi$.

Certificate Extraction

A star-type for every $\tau \in \mathbb{T}$ is sufficient!

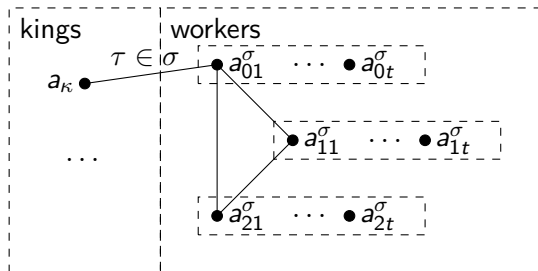
- ▶ Let \mathfrak{A} be a model for \mathbb{T}
- ▶ For every $\tau \in \mathbb{T}$ choose $a_\tau, b_\tau \in A$ such that $\text{tp}^{\mathfrak{A}}[a_\tau, b_\tau] = \tau$
- ▶ $\mathcal{S} = \left\{ \text{stp}^{\mathfrak{A}}[a_\tau] \mid \tau \in \mathbb{T} \right\}$ is a *polynomial* certificate

Certificate Expansion

Theorem

*Let S be a certificate for the type instance T and let $t \geq |T|$.
Then T has a finite model in which every worker type is realized at least t times.*

Construction



- ▶ single element for each king
- ▶ 3 blocks of t elements for each worker star-type
- ▶ king-to-element determined by the star-type of the element
- ▶ worker-to-worker between consecutive blocks
- ▶ completion to a full structure

Summary

- ▶ Type realizability for \mathcal{L}^2 is in NPTIME
- ▶ \mathcal{L}^2 has the finite model property and its satisfiability problem is in NEXPTIME

Strategy

- ▶ Consider the two-variable logic with a single builtin equivalence symbol $\mathcal{L}^2\text{E}$
- ▶ Equivalence classes are structures for the *simpler* \mathcal{L}^2
- ▶ Exploit the previous result to ensure *classes are “consistent”* and figure out how to *glue them together*
- ▶ Make sure the argument is suitable for induction to get to $\mathcal{L}^2\text{eE}_{\text{refine}}$

Galaxies and Cosmos

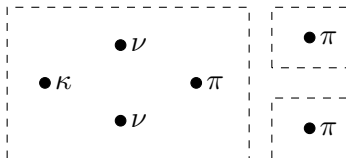
Let \mathfrak{A} be a \mathcal{L}^21E -structure for the type instance T .

- ▶ classes of \mathfrak{A} are the *galaxies* of \mathfrak{A}
- ▶ \mathfrak{A} is *the cosmos*
- ▶ $\tau \in T$ is *galactic* if $\mathbf{e}(\mathbf{x}, \mathbf{y}) \in \tau$, $T^g \subseteq T$
- ▶ otherwise τ is *cosmic*, $T^c \subseteq T$

Noble and Peasant Types

- ▶ $\nu \in \Pi_T$ is *noble* if no cosmic τ has $\text{tp}_x\tau = \text{tp}_y\tau = \nu$; the set of noble types is N_T
- ▶ $\pi \in \Pi_T$ is *peasant* if it is not noble; the set of peasant types is Π_T
- ▶ kings are noble
- ▶ peasants are workers
- ▶ a galaxy is *noble* if it realizes a noble type
- ▶ a galaxy is *peasant* if it realizes only peasant types

Example



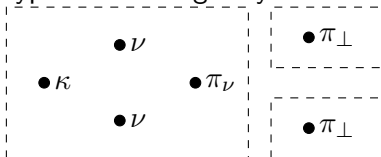
Noble galaxies may contain peasants!

Noble Distinguishability

- ▶ A structure is *nobly distinguished* if every noble galaxy realizes only noble types

Noble Distinguishability

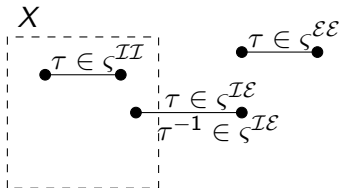
- ▶ A structure is *nobly distinguished* if every noble galaxy realizes only noble types
- ▶ Reduction: tag peasants in a noble galaxy with some noble type from that galaxy



Cosmic Spectrums

- ▶ Let \mathfrak{A} be a nobly distinguished model for \mathbb{T} having at least 2 galaxies
- ▶ The cosmic spectrum of a galaxy $X \subset A$ is

$$\text{csp}^{\mathfrak{A}}[X] = (\varsigma^{\mathcal{II}}, \varsigma^{\mathcal{IE}}, \varsigma^{\mathcal{EI}}, \varsigma^{\mathcal{EE}})$$



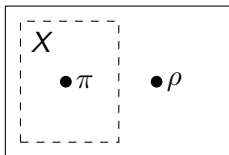
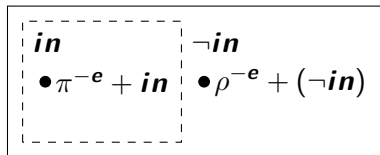
Cosmic Spectrums

A cosmic spectrum $\varsigma = (\varsigma^{\mathcal{II}}, \varsigma^{\mathcal{IE}}, \varsigma^{\mathcal{EI}}, \varsigma^{\mathcal{EE}})$ over T is a tuple satisfying:

- $(\varsigma^{\mathcal{II}})$ The set of *internal types* $\varsigma^{\mathcal{II}} \subseteq T^g$ is a set of galactic types that is closed under inversion.
- $(\varsigma^{\mathcal{IE}})$ The set of *boundary types* $\varsigma^{\mathcal{IE}} \subseteq T^c$ is a nonempty set of cosmic types.
- $(\varsigma^{\mathcal{EI}})$ The set of *inverted boundary types* is: $\varsigma^{\mathcal{EI}} = \{\tau^{-1} \mid \tau \in \varsigma^{\mathcal{IE}}\}$.
- $(\varsigma^{\mathcal{EE}})$ The set of *external types* $\varsigma^{\mathcal{EE}} \subseteq T$ is a set of 2-types that is closed under inversion.
- (ς^T) We require that $T = \varsigma^{\mathcal{II}} \cup \varsigma^{\mathcal{IE}} \cup \varsigma^{\mathcal{EI}} \cup \varsigma^{\mathcal{EE}}$.
- (ς^{NP}) The (nonempty) set $\text{Tp}_x \varsigma = (\text{tp}_x \upharpoonright \varsigma^{\mathcal{IE}})$ is the set of *internal 1-types* of ς . We require that $\text{Tp}_x \varsigma \subseteq N_T$ or $\text{Tp}_x \varsigma \subseteq P_T$.

Locally consistent cosmic spectrums

- ▶ The *spectral type instance* T^ς of the $\mathcal{L}^2 eE_{\text{refine}}$ -cosmic spectrum ς is $T^\varsigma = T_{II}^\varsigma \cup T_{IE}^\varsigma \cup T_{EI}^\varsigma \cup T_{EE}^\varsigma$, where $T_{\mathcal{X}\mathcal{Y}}^\varsigma = \{\tau_{\mathcal{X}\mathcal{Y}} \mid \tau \in \varsigma^{\mathcal{X}\mathcal{Y}}\}$, where $\tau_{\mathcal{X}\mathcal{Y}}$ works by “removing e and tagging the ends with a new predicate symbol *in*”.
- ▶ ς is *locally consistent* if T^ς is realizable over *the simpler* $\mathcal{L}^2(e-1)E_{\text{refine}}$

 $\mathfrak{A} : \mathcal{L}^2 eE_{\text{refine}}$

 \Rightarrow
 $\mathfrak{A}' : \mathcal{L}^2(e-1)E_{\text{refine}}$


Certificates

A *certificate* \mathcal{S} for the type instance T is a nonempty set of locally consistent cosmic spectra satisfying:

$(\mathcal{S}T^c)$ If $\tau \in T^c$ then some $\varsigma \in \mathcal{S}$ has $\tau \in \varsigma^{\mathcal{IE}}$.

$(\mathcal{S}T^g)$ If $\tau \in T^g$ then some $\varsigma \in \mathcal{S}$ has $\tau \in \varsigma^{\mathcal{II}}$.

$(\mathcal{S}\nu)$ If $\nu \in N_T$ and $\varsigma, \varsigma' \in \mathcal{S}$ have $\nu \in \text{Tp}_x \varsigma$ and $\nu \in \text{Tp}_x \varsigma'$, then $\varsigma' = \varsigma$.

Analogue

$$\mathcal{L}^2 \iff \mathcal{L}^2 eE_{\text{refine}}$$

models \iff nobly distinguished models

elements \iff galaxies

kings \iff nobles

workers \iff peasants

star-types \iff locally consistent cosmic spectrums

certificates \iff certificates

Certificate Expansion

Theorem

Let S be a certificate for the type instance T over the $\mathcal{L}^2\text{eE}_{\text{refine}}$ -classified signature $\langle \Sigma, \bar{\mathbf{m}} \rangle$. Then T has a finite model in which each worker type is realized at least t times.

- ▶ Type realizability for $\mathcal{L}^2\text{eE}_{\text{refine}}$ is in NPTIME
- ▶ Satisfiability for $\mathcal{L}^2\text{eE}_{\text{refine}}$ is in NEXPTIME