Extensions of two-variable logic

Reijo Jaakkola

Tampere University

December 11, 2020

Background

Theorem (Mortimer 75, Grädel et al. 97)

FO^2 *has the finite model property and its satisfiability problem is* NexpTime*-complete.*

Background

Theorem (Mortimer 75, Grädel et al. 97)

FO^2 *has the finite model property and its satisfiability problem is* NEXPTIME*-complete.*

Dana Scott proved already in 1962 that equality-free FO^2 is decidable by reducing its satisfiability problem to the Gödel-Kalmár-Schütte class without equality $[\exists^*\forall^2\exists^*, \mathit{all}]$.

Question

FO^2 does not cope well with relations of arity higher than two. For instance, it can't even express the property that a ternary relation is non-empty.

Question

FO^2 does not cope well with relations of arity higher than two. For instance, it can't even express the property that a ternary relation is non-empty.

How can we extend the expressive power of FO^2 while preserving decidability?

This talk

In this talk the problem of finding tame extensions of FO^2 will be approached using an algebraic framework.

This talk

In this talk the problem of finding tame extensions of FO^2 will be approached using an algebraic framework. We will focus on extensions of *equality-free* FO^2.

Relational operators

Definition

If $k \in \mathbb{Z}_+$, then a k-ary AD-relation over a set A is a pair (X, k), where $X \subseteq A^k$.

Relational operators

Definition

If $k \in \mathbb{Z}_+$, then a k-ary AD-relation over a set A is a pair (X, k), where $X \subseteq A^k$.

If A is a set, then $\mathrm{AD}(A)$ denotes the set of AD-relations over A.

Relational operators

Definition

If $k \in \mathbb{Z}_+$, then a k-ary AD-relation over a set A is a pair (X, k), where $X \subseteq A^k$.

If A is a set, then $\mathrm{AD}(A)$ denotes the set of AD-relations over A.

Definition

A k-ary relational operator F is a mapping which associates to every set A a function F_A

$$F_A : \mathrm{AD}(A)^k \to \mathrm{AD}(A)$$

Algebraic way of defining logics

Definition

Let $\mathcal{F}$ be a set of operators and let σ be a relational vocabulary. The set of terms $\mathrm{GRA}(\mathcal{F})[\sigma]$ is defined by the following grammar.

$$\mathcal{T} ::= R \mid F(\mathcal{T}, ..., \mathcal{T}),$$

where $R \in \sigma$ and $F \in \mathcal{F}$.

Algebraic way of defining logics

Definition

Let $\mathcal{F}$ be a set of operators and let σ be a relational vocabulary. The set of terms $\mathrm{GRA}(\mathcal{F})[\sigma]$ is defined by the following grammar.

$$\mathcal{T} ::= R \mid F(\mathcal{T}, ..., \mathcal{T}),$$

where $R \in \sigma$ and $F \in \mathcal{F}$.

Definition

Given a model $\mathfrak{A}$ of vocabulary σ and term $\mathcal{T} \in \mathrm{GRA}(\mathcal{F})[\sigma]$, its interpretation $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is defined recursively as follows.

1. $[\![R]\!]_{\mathfrak{A}} := R^{\mathfrak{A}}$
2. $[\![F(\mathcal{T}_1, ..., \mathcal{T}_n)]\!]_{\mathfrak{A}} := F_A([\![\mathcal{T}_1]\!]_{\mathfrak{A}}, ..., [\![\mathcal{T}_n]\!]_{\mathfrak{A}})$

Connection with standard syntax

To compare the expressive power of terms and formulas, we note that each first-order formula $\varphi(v_{i_1}, \ldots, v_{i_k})$, where $i_1 \leq \ldots \leq i_k$, defines over each model $\mathfrak{A}$ an AD-relation

$$[\![\varphi]\!]_{\mathfrak{A}} = (\{(a_1, \ldots, a_k) \in A^k \mid \mathfrak{A} \models \varphi(a_1, \ldots, a_k)\}, k).$$

Connection with standard syntax

To compare the expressive power of terms and formulas, we note that each first-order formula $\varphi(v_{i_1}, \ldots, v_{i_k})$, where $i_1 \leq \ldots \leq i_k$, defines over each model $\mathfrak{A}$ an AD-relation

$$[\![\varphi]\!]_{\mathfrak{A}} = (\{(a_1, \ldots, a_k) \in A^k \mid \mathfrak{A} \models \varphi(a_1, \ldots, a_k)\}, k).$$

For example the formula $R(v_1, v_2)$ defines the AD-relation $(R^{\mathfrak{A}}, 2)$ and $R(v_2, v_1)$ defines the AD-relation $((R^{\mathfrak{A}})^{-1}, 2)$.

Relevant operators

We start by introducing a sufficient set of relational operators for characterization (roughly) the equality-free FO^2.

Relevant operators

We start by introducing a sufficient set of relational operators for characterization (roughly) the equality-free FO^2.

First we need some operators for expressing atomic formulas.

Relevant operators

We start by introducing a sufficient set of relational operators for characterization (roughly) the equality-free FO^2.

First we need some operators for expressing atomic formulas. Given an AD-relation (X, k) over A, where $k \geq 2$, we define

$$s((X,k)) = (\{(a_1, ..., a_k, a_{k-1}) \in A^k \mid (a_1, ..., a_{k-1}, a_k) \in X\}, k)$$

$$I((X,k)) = (\{(a_1, ..., a_{k-1}) \in A^k \mid (a_1, ..., a_{k-1}, a_{k-1}) \in X\}, k-1)$$

Relevant operators

We start by introducing a sufficient set of relational operators for characterization (roughly) the equality-free FO^2.

First we need some operators for expressing atomic formulas. Given an AD-relation (X, k) over A, where $k \geq 2$, we define

$$s((X,k)) = (\{(a_1,\dots,a_k,a_{k-1}) \in A^k \mid (a_1,\dots,a_{k-1},a_k) \in X\}, k)$$

$$I((X,k)) = (\{(a_1,\dots,a_{k-1}) \in A^k \mid (a_1,\dots,a_{k-1},a_{k-1}) \in X\}, k-1)$$

Now $R(v_2, v_1)$ can be expressed as sR and $R(v_1, v_1)$ as IR.

Relevant operators

We will also need operators for boolean combinations.

Relevant operators

We will also need operators for boolean combinations. Given AD-relations (X, k) and (Y, ℓ), where $k = \ell$, we define

$$\cap((X, k), (Y, \ell)) = (X \cap Y, k).$$

Relevant operators

We will also need operators for boolean combinations. Given AD-relations (X, k) and (Y, ℓ), where $k = \ell$, we define

$$\cap((X, k), (Y, \ell)) = (X \cap Y, k).$$

On the other hand, if $\ell = 1$, we define

$$C((X, k), (Y, \ell)) = (\{\bar{a} \in X \mid a_k \in Y\}, k).$$

Relevant operators

We will also need operators for boolean combinations. Given AD-relations (X, k) and (Y, ℓ), where $k = \ell$, we define

$$\cap((X, k), (Y, \ell)) = (X \cap Y, k).$$

On the other hand, if $\ell = 1$, we define

$$C((X, k), (Y, \ell)) = (\{\bar{a} \in X \mid a_k \in Y\}, k).$$

Now $R(v_1, v_2) \wedge S(v_1, v_2)$ can be expressed as $(R \cap S)$, and $R(v_1, v_2) \wedge S(v_2)$ can be expressed as $C(R, S)$.

Relevant operators

Finally we need operators for quantification.

Relevant operators

Finally we need operators for quantification. Given an AD-relation (X, k) over A, where $k \geq 1$, we define

$$\exists_1((X, k)) = (\{a \in A \mid \text{There exists } \overline{b} \in A^{k-1} \text{ so that } a\overline{b} \in X\}, 1)$$

and we define $\exists_0((X, k))$ to be $(\{\varnothing\}, 0)$ if and only if X is non-empty.

Algebraic characterization of FO^2

Theorem

$\mathrm{GRA}(s, I, \neg, \cap, C, \exists_1, \exists_0)$ *and* FO^2 *are sententially equiexpressive over vocabularies with at most binary relation symbols.*

Algebraic characterization of FO^2

Theorem

$\mathrm{GRA}(s, I, \neg, \cap, C, \exists_1, \exists_0)$ and FO^2 are sententially equiexpressive over vocabularies with at most binary relation symbols.

Using the fact that the satisfiability problem for two-variable fluted logic is NEXPTIME-hard, we obtain the following complexity result.

Theorem

The satisfiability problem for $\mathrm{GRA}(s, I, \neg, \cap, C, \exists_1, \exists_0)$ is NEXPTIME-hard.

Extending FO^2

There are at least three ways of extending the previous algebra in such a way that we can still prove a corresponding upper bound.

Extending FO^2

There are at least three ways of extending the previous algebra in such a way that we can still prove a corresponding upper bound.

The first option is to add a cyclic permutation operator.

Extending FO^2

There are at least three ways of extending the previous algebra in such a way that we can still prove a corresponding upper bound.

The first option is to add a cyclic permutation operator. Given an AD-relation (X, k) over A, where $k \geq 2$, we define

$$p((X, k)) := (\{(a_2, ..., a_k, a_1) \in A^k \mid (a_1, ..., a_k) \in X\}, k).$$

If $k \leq 1$, then $p((X, k)) = (X, k)$.

Extending FO^2

There are at least three ways of extending the previous algebra in such a way that we can still prove a corresponding upper bound.

The first option is to add a cyclic permutation operator. Given an AD-relation (X, k) over A, where $k \geq 2$, we define

$$p((X,k)) := (\{(a_2, ..., a_k, a_1) \in A^k \mid (a_1, ..., a_k) \in X\}, k).$$

If $k \leq 1$, then $p((X,k)) = (X,k)$.

Using the cyclic permutation (together with s and I), we can define arbitrary atomic formulas.

Extending FO^2

There are at least three ways of extending the previous algebra in such a way that we can still prove a corresponding upper bound.

The first option is to add a cyclic permutation operator. Given an AD-relation (X, k) over A, where $k \geq 2$, we define

$$p((X,k)) := (\{(a_2, \ldots, a_k, a_1) \in A^k \mid (a_1, \ldots, a_k) \in X\}, k).$$

If $k \leq 1$, then $p((X,k)) = (X,k)$.

Using the cyclic permutation (together with s and I), we can define arbitrary atomic formulas. The resulting logic is (roughly) equivalent to the equality-free uniform one-dimensional logic, which was introduced by Hella and Kuusisto.

Extending FO^2

The second option is to replace the operators $\exists_1$ and $\exists_0$ with the projection operator $\exists$.

Extending FO^2

The second option is to replace the operators $\exists_1$ and $\exists_0$ with the projection operator $\exists$. Given an AD-relation (X, k) over A, where $k \geq 1$, we define

$$\exists((X, k)) = (\{\bar{a} \in A^{k-1} \mid \bar{a}b \in X, \text{ for some } b \in A\}, k-1).$$

If $k = 0$, then $\exists((X, k)) = (X, k)$.

Extending FO^2

The second option is to replace the operators $\exists_1$ and $\exists_0$ with the projection operator $\exists$. Given an AD-relation (X, k) over A, where $k \geq 1$, we define

$$\exists((X, k)) = (\{\bar{a} \in A^{k-1} \mid \bar{a}b \in X, \text{ for some } b \in A\}, k-1).$$

If $k = 0$, then $\exists((X, k)) = (X, k)$.

The resulting logic has access to arbitrary quantifier alternations.

Extending FO^2

The second option is to replace the operators $\exists_1$ and $\exists_0$ with the projection operator $\exists$. Given an AD-relation (X, k) over A, where $k \geq 1$, we define

$$\exists((X,k)) = (\{\overline{a} \in A^{k-1} \mid \overline{a}b \in X, \text{ for some } b \in A\}, k-1).$$

If $k = 0$, then $\exists((X,k)) = (X,k)$.

The resulting logic has access to arbitrary quantifier alternations. Thus, it can for instance express statements such as $\forall x \exists y \forall z \exists w (R(x,y,z,w) \wedge P(z) \wedge P(w))$.

Extending FO^2

The third option is to replace the operators $\cap$ and C with the suffix-intersection operator $\dot{\cap}$.

Extending FO^2

The third option is to replace the operators $\cap$ and C with the suffix-intersection operator $\dot{\cap}$. Given two AD-relations (X, k) and (Y, ℓ) over A, where $k \geq \ell$, we define

$$\dot{\cap}((X,k),(Y,\ell)) = (\{\bar{a} \in X \mid (a_{k-\ell+1}, \dots, a_k) \in Y\}, k).$$

If $k < \ell$, then $\dot{\cap}((X,k),(Y,\ell)) = \dot{\cap}((Y,\ell),(X,k))$.

Extending FO^2

The third option is to replace the operators $\cap$ and C with the suffix-intersection operator $\dot{\cap}$. Given two AD-relations (X, k) and (Y, ℓ) over A, where $k \geq \ell$, we define

$$\dot{\cap}((X,k),(Y,\ell)) = (\{\bar{a} \in X \mid (a_{k-\ell+1}, \ldots, a_k) \in Y\}, k).$$

If $k < \ell$, then $\dot{\cap}((X,k),(Y,\ell)) = \dot{\cap}((Y,\ell),(X,k))$.

Using the suffix intersection operator, we can express statements such as $\forall x \exists y \exists z (S(x,y,z) \wedge R(y,z) \wedge P(z))$.

Extending FO^2

The third option is to replace the operators $\cap$ and C with the suffix-intersection operator $\dot{\cap}$. Given two AD-relations (X, k) and (Y, ℓ) over A, where $k \geq \ell$, we define

$$\dot{\cap}((X, k), (Y, \ell)) = (\{\overline{a} \in X \mid (a_{k-\ell+1}, \dots, a_k) \in Y\}, k).$$

If $k < \ell$, then $\dot{\cap}((X, k), (Y, \ell)) = \dot{\cap}((Y, \ell), (X, k))$.

Using the suffix intersection operator, we can express statements such as $\forall x \exists y \exists z (S(x, y, z) \wedge R(y, z) \wedge P(z))$.

The resulting logic can be seen as one-dimensional fragment of fluted logic.

Further extensions

Can we have add both p and $\dot{\cap}$, while preserving decidability?

Further extensions

Can we have add both p and $\dot{\cap}$, while preserving decidability? No.

Theorem

The satisfiability problem for $\mathrm{GRA}(p, \neg, \dot{\cap}, \exists_1, \exists_0)$ *is* Π^0_1*-complete.*

Further extensions

Can we have add both p and $\dot{\cap}$, while preserving decidability? No.

Theorem

The satisfiability problem for $\mathrm{GRA}(p, \neg, \dot{\cap}, \exists_1, \exists_0)$ *is* Π^0_1*-complete.*

How about p and $\exists$?

Further extensions

Can we have add both p and $\dot{\cap}$, while preserving decidability? No.

Theorem

The satisfiability problem for $\mathrm{GRA}(p, \neg, \dot{\cap}, \exists_1, \exists_0)$ *is* Π^0_1*-complete.*

How about p and $\exists$? No.

Theorem

The satisfiability problem for $\mathrm{GRA}(p, \neg, \cap, \exists)$ *is* Π^0_1*-complete.*

Further extensions

How about $\dot{\cap}$ and $\exists$?

Further extensions

How about $\dot{\cap}$ and $\exists$? Yes, but it will cost.

Theorem (Pratt-Hartmann et al. 2019)

The satisfiability problem for $\mathrm{GRA}(\neg, \dot{\cap}, \exists)$ *is Tower-complete.*

Further extensions

How about $\dot{\cap}$ and $\exists$? Yes, but it will cost.

Theorem (Pratt-Hartmann et al. 2019)

The satisfiability problem for $\mathrm{GRA}(\neg, \dot{\cap}, \exists)$ *is Tower-complete.*

Here Tower is the class of problems solvable by a Turing machine (deterministic or not) in time $F_3(p(n))$, where p is an elementary function and $F_3(x)$ is roughly speaking $\mathit{tower}(x, x)$.

Constructing models of bounded size

For all of the extensions of FO^2 mentioned here, one can prove that they have the bounded model property: if $\mathcal{T} \in \mathrm{GRA}(\mathcal{F})$ has a model, then it has a model of size at most $2^{(|\mathcal{T}|)}$.

Constructing models of bounded size

For all of the extensions of FO^2 mentioned here, one can prove that they have the bounded model property: if $\mathcal{T} \in \mathrm{GRA}(\mathcal{F})$ has a model, then it has a model of size at most $2^{(|\mathcal{T}|)}$.

To demonstrate the ideas involved in these types of constructions, we will sketch a proof of this property for $\mathrm{GRA}(p, s, I, \neg, C, \cap, \exists_1, \exists_0)$.

Scott normal form

We say that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, C, \cap, \exists_1, \exists_0)$ is in scott normal form, if it has the following form

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall},$$

where $\mathcal{T}_i^{\exists}, \mathcal{T}_j^{\forall} \in \mathrm{GRA}(s, I, \neg, C, \cap)$.

Scott normal form

We say that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, C, \cap, \exists_1, \exists_0)$ is in scott normal form, if it has the following form

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall},$$

where $\mathcal{T}_i^{\exists}, \mathcal{T}_j^{\forall} \in \mathrm{GRA}(s, I, \neg, C, \cap)$.

Lemma

There exists a nondeterministic polynomial time procedure which translates each $\mathrm{GRA}(p, s, I, \neg, C, \cap, \exists_1, \exists_0)$ term $\mathcal{T}$ to a $\mathrm{GRA}(p, s, I, \neg, C, \cap, \exists_1, \exists_0)$ term $\mathcal{T}'$ in normal form that is equisatisfiable with $\mathcal{T}$ in the following sense. If $\mathfrak{A} \models \mathcal{T}$, then there exists an extension of $\mathfrak{A}'$ so that $\mathfrak{A}' \models \mathcal{T}'$, and vice versa, if $\mathfrak{A} \models \mathcal{T}'$, then $\mathfrak{A} \models \mathcal{T}$.

Types and tables

Definition

A 1-type π over a vocabulary σ is a maximally consistent set of unary terms of $\mathrm{GRA}(I, \neg)[\sigma]$.

Types and tables

Definition

A 1-type π over a vocabulary σ is a maximally consistent set of unary terms of $\mathrm{GRA}(I, \neg)[\sigma]$.

Definition

A k-table over a vocabulary σ is a maximally consistent set of k-ary terms of $\mathrm{GRA}(p, s, I)[\sigma]$ and their negations.

Types and tables

If $\mathfrak{A}$ is a model of vocabulary σ and $(a_1, ..., a_k) \in A^k$, then we use $tp_{\mathfrak{A}}(a_1, ..., a_k)$ to denote the k-table that the tuple realizes.

Types and tables

If $\mathfrak{A}$ is a model of vocabulary σ and $(a_1, ..., a_k) \in A^k$, then we use $tp_{\mathfrak{A}}(a_1, ..., a_k)$ to denote the k-table that the tuple realizes.

Observation: if $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, \cap, C)[\sigma]$ is a k-ary term and $\mathfrak{A}$ is a model of vocabulary σ, then whether or not a tuple $(a_1, ..., a_k)$ belongs to the interpretation of $\mathcal{T}$ depends only on $tp_{\mathfrak{A}}(a_1, ..., a_k)$ and $tp_{\mathfrak{A}}(a_i)$.

Constructing the model

Suppose that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, \cap, C, \exists_1, \exists_0)$ is a term in normal form:

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall}$$

Let $\mathfrak{A}$ be a model of $\mathcal{T}$.

Constructing the model

Suppose that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, \cap, C, \exists_1, \exists_0)$ is a term in normal form:

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall}$$

Let $\mathfrak{A}$ be a model of $\mathcal{T}$. Now, for every 1-type π that is realized in $\mathfrak{A}$, we choose some representative $a \in A$ so that $tp_{\mathfrak{A}}(a) = \pi$.

Constructing the model

Suppose that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, \cap, C, \exists_1, \exists_0)$ is a term in normal form:

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall}$$

Let $\mathfrak{A}$ be a model of $\mathcal{T}$. Now, for every 1-type π that is realized in $\mathfrak{A}$, we choose some representative $a \in A$ so that $tp_{\mathfrak{A}}(a) = \pi$.
Then, for every $i \in I$ and π we choose some set $W_{\pi,i} = \{c_1, ..., c_k\} \subseteq A$ so that $(a, c_1, ..., c_k) \in [\![\mathcal{T}_i^{\exists}]\!]_{\mathfrak{A}}$.

Constructing the model

Suppose that $\mathcal{T} \in \mathrm{GRA}(p, s, I, \neg, \cap, C, \exists_1, \exists_0)$ is a term in normal form:

$$\bigcap_{i \in I} \forall_0 \exists_1 \mathcal{T}_i^{\exists} \cap \bigcap_{j \in J} \forall_0 \mathcal{T}_i^{\forall}$$

Let $\mathfrak{A}$ be a model of $\mathcal{T}$. Now, for every 1-type π that is realized in $\mathfrak{A}$, we choose some representative $a \in A$ so that $tp_{\mathfrak{A}}(a) = \pi$. Then, for every $i \in I$ and π we choose some set $W_{\pi,i} = \{c_1, ..., c_k\} \subseteq A$ so that $(a, c_1, ..., c_k) \in [\![\mathcal{T}_i^{\exists}]\!]_{\mathfrak{A}}$. As the domain of the new model $\mathfrak{B}$, we will take the set

$$B = \bigcup W_{\pi,i,j},$$

where $j \in \{0, 1, 2\}$ and all the sets $W_{\pi,i,j}$ are pairwise disjoint.

Assigning witnesses

We start the model construction by making sure that all existential requirements are satisfied.

Assigning witnesses

We start the model construction by making sure that all existential requirements are satisfied.

So let $i \in I$ and $b \in W_{\pi', i', j}$.

Assigning witnesses

We start the model construction by making sure that all existential requirements are satisfied.

So let $i \in I$ and $b \in W_{\pi',i',j}$. If a is the element associated to $\pi = tp_{\mathfrak{B}}(b)$, then we know that the elements of $W_{\pi,i} = (c_1, ..., c_k)$ form a witness for a.

Assigning witnesses

We start the model construction by making sure that all existential requirements are satisfied.

So let $i \in I$ and $b \in W_{\pi',i',j}$. If a is the element associated to $\pi = tp_{\mathfrak{B}}(b)$, then we know that the elements of $W_{\pi,i} = (c_1, ..., c_k)$ form a witness for a. So, if $W_{\pi,i,j+1 \mod 3} = (d_1, ..., d_k)$, where $tp_{\mathfrak{B}}(d_i) = tp_{\mathfrak{A}}(c_i)$, for every $1 \leq i \leq k$, then we define $tp_{\mathfrak{B}}(b, d_1, ..., d_k) = tp_{\mathfrak{A}}(a, c_1, ..., c_k)$.

Completing the model

Since the term $\mathcal{T}$ makes also global requirements, we have to specify the k-tables for each k-tuple.

Completing the model

Since the term $\mathcal{T}$ makes also global requirements, we have to specify the k-tables for each k-tuple.

Let $\overline{b} \in B^k$. By construction, there exists some $\overline{a} \in A^k$ so that $tp_{\mathfrak{B}}(b_i) = tp_{\mathfrak{A}}(a_i)$, for every $1 \leq i \leq k$.

Completing the model

Since the term $\mathcal{T}$ makes also global requirements, we have to specify the k-tables for each k-tuple.

Let $\overline{b} \in B^k$. By construction, there exists some $\overline{a} \in A^k$ so that $tp_{\mathfrak{B}}(b_i) = tp_{\mathfrak{A}}(a_i)$, for every $1 \leq i \leq k$. We then define $tp_{\mathfrak{B}}(\overline{b}) = tp_{\mathfrak{A}}(\overline{a})$.

Completing the model

Since the term $\mathcal{T}$ makes also global requirements, we have to specify the k-tables for each k-tuple.

Let $\overline{b} \in B^k$. By construction, there exists some $\overline{a} \in A^k$ so that $tp_{\mathfrak{B}}(b_i) = tp_{\mathfrak{A}}(a_i)$, for every $1 \leq i \leq k$. We then define $tp_{\mathfrak{B}}(\overline{b}) = tp_{\mathfrak{A}}(\overline{a})$.

This completes the construction and the resulting model $\mathfrak{B}$ will be a model of $\mathcal{T}$.

Imposing syntactical restrictions to guarded fragments

Guarded fragment GF can be given a nice algebraic characterization with the difference operator.

Imposing syntactical restrictions to guarded fragments

Guarded fragment GF can be given a nice algebraic characterization with the difference operator. Given AD-relations (X, k) and (Y, k), we define

$$\setminus((X, k), (Y, k)) = (X \setminus Y, k).$$

Imposing syntactical restrictions to guarded fragments

Guarded fragment GF can be given a nice algebraic characterization with the difference operator. Given AD-relations (X, k) and (Y, k), we define

$$\backslash((X, k), (Y, k)) = (X \backslash Y, k).$$

Theorem

$\mathrm{GRA}(p, s, I, \backslash, \dot{\cap}, \exists)$ *is sententially equivalent with equality-free* GF *and its satisfiability problem is* $2\textsc{ExpTime}$*-complete.*

Imposing syntactical restrictions to guarded fragments

Dropping p or replacing $\exists$ with $\exists_1$ and $\exists_0$ will lead to an easier satisfiability problem.

Imposing syntactical restrictions to guarded fragments

Dropping p or replacing $\exists$ with $\exists_1$ and $\exists_0$ will lead to an easier satisfiability problem.

Theorem

The satisfiability problem for $\mathrm{GRA}(s, I, \setminus, \dot{\cap}, \exists)$ *is* $\textsc{ExpTime}$*-complete.*

Theorem (Kieronski, 2019)

The satisfiability problem for $\mathrm{GRA}(p, s, I, \setminus, \dot{\cap}, \exists_1, \exists_0)$ *is* $\textsc{NexpTime}$*-complete.*

Summary

Imposing a combination of one-dimensionality, uniformity and restricted permutations of variables leads to decidable extensions of FO^2.

Summary

Imposing a combination of one-dimensionality, uniformity and restricted permutations of variables leads to decidable extensions of FO^2. Furthermore, more liberal restrictions seem to easily lead to undecidability.

Summary

Imposing a combination of one-dimensionality, uniformity and restricted permutations of variables leads to decidable extensions of FO^2. Furthermore, more liberal restrictions seem to easily lead to undecidability.

The algebras $\mathrm{GRA}(p, s, I, \neg, \cap, C, \exists_1, \exists_0)$ and $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists_1, \exists_0)$ remain decidable even in the presence of equality. Does the same hold for $\mathrm{GRA}(s, I, \neg, \cap, C, \exists)$?