Tampere University

Reijo Jaakkola

ALGEBRAIC FRAGMENTS OF FIRST-ORDER LOGIC

Faculty of Information Technology and Communication Sciences
Master's Thesis
Mathematics
April 2021

TIIVISTELMÄ

Reijo Jaakkola: Algebraic fragments of first-order logic
Pro gradu -tutkielma
Tampereen yliopisto
Matematiikan ja tilastotieteen tutkinto-ohjelma
Huhtikuu 2021

Tämän tutkielman tarkoitus on käsitellä algebrallista lähestymistapaa ensimmäisen kertaluvun logiikan fragmenttien luokitteluun niiden ratkeavuuden perusteella. Tutkielman alussa esittelemme relaatio-operaattorin käsitteen, ja näytämme, kuinka sen avulla voidaan määritellä algebrallisesti logiikoita. Osoitamme myös, kuinka ensimmäisen kertaluvun logiikan voi määritellä tätä kautta algebrallisesti käyttämällä vain äärellisen montaa relaatio-operaattoria. Tämän jälkeen siirrymme tarkastelemaan ensimmäisen kertaluvun logiikan fragmentteja. Osoitamme, että monet jo aikaisemmin tunnetut ensimmäisen kertaluvun logiikan fragmentit voidaan myös määritellä algebrallisesti ja jopa niin, että fragmentin ja sitä vastaavan algebran toteutuvuusongelman laskennallinen vaativuus on täsmälleen sama.

Seuraavaksi tarkastelemme erilaisten algebrallisesti määrättyjen fragmenttien toteutuvuusongelman kompleksisuutta. Aloitamme tarkastelemalla erään ensimmäisen kertaluvun logiikan algebrallisen karakterisoinnin fragmenttien toteutuvuusongelmien laskennallista vaativuutta. Tarkemmin sanottuna tarkastelemme, kuinka algebrallisen fragmentin toteutuvuusongelman laskennallinen vaativuus muuttuu, mikäli poistamme joitakin relaatio-operaattoreita sen syntaktista. Saamme luokiteltua melkein täydellisesti näin syntyvät fragmentit niiden laskennallisen vaativuuden mukaan.

Tämän jälkeen tutkimme järjestetyn logiikan ja sen laajennusten toteutuvuusongelmien laskennallista vaativuutta. Järjestetyssä logiikassa rajoitetaan muuttujien permutointia ja järjestystä, jossa niitä voidaan kvantifioida. Osoittautuu, että tutkielmassa käytetty algebrallinen lähestymistapa soveltuu poikkeuksellisen hyvin tämän kaltaisten fragmenttien tutkimiseen, sillä näiden logiikoiden syntaksi on luonnollisempaa esittää siten, että kaavoissa ei esiinny ollenkaan muuttujia. Onnistumme määrittämään järjestetyn logiikan toteutuvuusongelman tarkan kompleksisuuden, jonka lisäksi paikanamme tarkan rajan sille, kuinka paljon järjestetyn logiikan syntaktisia rajoituksia voidaan nostaa sillä tavalla, että logiikka pysyy ratkeavana.

Tutkielman lopussa tarkastelemme yleisellä tasolla ensimmäisen kertaluvun logiikan toteutuvuusongelmaa. Aloitamme osoittamalla, että annetun ensimmäisen kertaluvun logiikan fragmentin – eli sen rekursiivinen osajoukon – ratkeavuuden selvittämminen on Σ^0_3-täydellinen ongelma. Tämän jälkeen tarkastelemme ilmaisuvoiman ja ratkeavuuden välistä suhdetta ensimmäisen kertaluvun logiikan fragmenttien tapauksessa. Osoitamme esimerkiksi, että ei ole olemassa ilmaisuvoiman suhteen maksimaalista ratkeavaa ensimmäisen kertaluvun logiikan fragmenttia.

Avainsanat: ensimmäisen kertaluvun logiikka, laskennallinen vaativuus, toteutuvuusongelma

Contents

1 Introduction

An important invariant of a logic $\mathcal{L}$ is the complexity of its satisfiability problem, i.e. the problem of determining whether a given formula of $\mathcal{L}$ is satisfiable. A lot of work has been done towards understanding the complexity of the satisfiability problem in the case where $\mathcal{L}$ is a fragment of first-order logic FO, where by a fragment we mean any recursive subset of FO. One motivation for this research comes from the fact that many of the logics that are used in computer science applications, such as description logics, can be seen as fragments of FO, which means that results on the complexity of fragments of FO will imply results on the complexities of these logics.

Although a lot of effort has been put on trying to understand the complexity of various fragments of FO, one could still argue that the field is currently missing a unifying framework for these types of results. Initially, such a framework was provided by the program of classifying quantifier prefixes based on whether the satisfiability problem is decidable for the corresponding fragment of FO (see [3] for the results of this program). After this major program was completed in the 1980's, it became unclear as to where the field should move on. Of course the more ambitious goal of classifying in general fragments of FO based on whether they are decidable or not is hopeless, since there are simply too many fragments to consider, and most of them are probably of no interest to anybody.

In the end, the research started to move further towards questions related to logics that are actually used in computer science applications. For example, a lot of research has been done towards understanding how fragments of FO behave over restricted classes of structures, such as word structures or trees, which are important from the point of view of applications. There has also been a lot of research towards understanding how the complexity of a fragment of FO changes, if we allow the formulas to use built-in predicates, such as linear orders. For a taste of these types of results, see e.g. [14], [20] and [2].

One can't consider the more application oriented research to be fully satisfactory, since it does not seem to try to study more general questions, such as why certain fragments of FO behave better than others, although one should mention that there has been a considerable amount of research, which was initiated by the article [26], towards understanding why modal logics behave very well computationally[1]. However, no one has done a similar analysis concerning the question of why, for example, the two-variable logic is decidable (although the recently introduced uniform one-dimensional logic seems to provide a partial answer, see [7]).

I believe that the reason for the lack of research around more general questions concerning decidability questions is because it would appear that these questions can't be answered completely. However, I would argue that in many cases we can give *sufficient* answers towards these questions, i.e. answers of the form "if a logic has these and these properties, then its satisfiability problem is probably undecidable". I would even argue that such sufficient answers could also be of importance from the point of view of applications. When we are trying to formalize something, we often have an idea of what we would like to express, but we might not know whether there is a computationally nice logic for the job. Knowing such sufficient answers would help, since they could be used to demonstrate when we are asking for too much.

I would thus argue that there are good motivations for studying fragments of FO from a more general perspective. To support the study of these types of questions, we would ideally like to have a similar classification task as the one mentioned above, which concerned the type quantifier prefixes the formulas were allowed to have. Such questions allow us to focus our research and

[1]For more details on this topic, I recommend the illuminating article [18]

they help us in discovering new and possibly interesting fragments of FO. However, it also important to note that we do not need a single all encompassing task of classifying fragments of FO. Each classification will necessarily focus on one aspect, and a lot can be learned from comparing the different classifications.

The purpose of this thesis is to study one approach towards providing classifications for the fragments of FO. The main idea is to view logics as being generated by a finite set of algebraic operators, such as projection and intersection. If the initial set of algebraic operators is finite, this opens up the possibility of giving a complete classification for the complexities of the different logics obtained by using different combinations of the operators.

The idea of viewing logics as algebras is certainly not new. Out of all of the algebraic systems that have been studied in the literature, the closest approach to our system is Quine's predicate functor logic [24, 25, 22], since it also comes with a finite set of algbraic operators (for comparison, the cylindrical algebras of Tarski often require infinitely many individual operators). However, it was only pointed out quite recently in [16] that one could take an algebraic characterization of a logic $\mathcal{L}$ as a starting point for a fine-grained classification of the fragments of $\mathcal{L}$. This idea was then studied further in the preprints [11, 12, 10] to which this thesis is based on.

Next we will go through the structure of this thesis. The general approach of viewing logics as algebras will be presented in chapter 2. In chapter 3 we will present several algebraic characterizations for FO. The first two characterizations bear some degree of similarity with Codd's theorem, while the last characterization is based on cylindric algebras. In chapter 4 we will give algebraic characterizations for guarded fragment, two-variable logic and fluted logic. In all of the cases we are able to establish that the complexity of the satisfiability problem is the same for both the fragment and its algebraic characterization.

In chapter 5 we will study the fragments of one particular algebraic characterization of FO. In particular, we will study the complexity of the satisfiability problem for maximal fragments of this algebra, i.e. the fragments which are obtained by dropping *one* of the algebraic operators from its syntax. With one exception, we are able to determine the exact complexity of the satisfiability problem for these fragments.

Having collected a somewhat large list of operators to study, it also makes sense to start with a rather weak fragment of FO and then study what happens to its satisfiability problem if we add certain operators into its syntax. This approach will be investigated in chapter 6 where we will study ordered first-order logic, which turns out to be a fragment of fluted logic. We will start by showing that its satisfiability problem is Pspace-complete. We will then investigate two natural extensions of the ordered first-order logic, which are obtained from the ordered first-order logic by lifting some of its syntactical restrictions.

The last chapter of this thesis contains some general remarks on the satisfiability problem. First we will prove that the problem of determining whether the satisfiability problem for a fragment of FO is decidable is in general an undecidable problem, and we will in fact determine the exact complexity of this problem. After this we will prove two results concerning the relationship of decidability and expressive power for fragments of FO. The tone of the chapter is somewhat informal (in the sense that we rely on Church thesis) and it can be read independently from the rest of this thesis.

The work presented in chapters from 2 to 5 is almost entirely joint work Antti Kuusisto. Most of the results in these chapters are essentially contained in the preprints [16, 11, 12]. The Pspace upper bound on the complexity of ordered logic is due to Andreas Herzig [8] and the proof given in chapter 6 is a simplification of his original argument. The remaining results in chapter 6 are my own and most of them are contained in the preprint [10]. I have not been able to find references for the results proved in chapter 7, but I suspect that they are folklore. Concerning prerequisites, the reader is assumed to be familiar with FO and basic computational complexity theory. Concerning the latter part, the book [15] should cover all of the necessary background material.

I wish to thank Antti Kuusisto for suggesting the topic of my master's thesis and for acting as my research collaborator and thesis advisor. I cannot overstate how much I have learned from our many discussions on various topics. I also wish to thank him and Professor Lauri Hella for their useful comments on this thesis. Finally, I wish to acknowledge the financial support of the Academy of Finland project, *Theory of Computational Logics*, grant numbers 324435 and 328987.

2 Preliminaries

2.1 Conventions

We let VAR $= \{v_1, v_2, \dots\}$ denote the countably infinite set of exactly all variables used in first-order logic FO. We also use metavariables (e.g., $x, y, z, x_1, x_2 \dots$) to refer to symbols in VAR. When using the six metavariables x, y, z, u, v, w, we always assume $(x, y, z, u, v, w) = (v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}, v_{i_5}, v_{i_6})$ for some $i_1 < i_2 < i_3 < i_4 < i_5 < i_6$. We let $\hat{\tau}$ denote the full relational vocabulary containing countably infinitely many relation symbols of each arity $k \geq 0$. Throughout this work we assume that every vocabulary is a subset of $\hat{\tau}$.

The notion of a model is defined as usual in model theory. For simplicity, we restrict attention to relational models, i.e., vocabularies of models do not contain function or constant symbols. The domain of a model $\mathfrak{A}$ is denoted by A, the domain of $\mathfrak{B}$ by B et cetera. We say that a model $\mathfrak{A}$ is a model of vocabulary τ, if the set of relation symbols its interpreters is τ.

If τ is a vocabulary, then we use FO$[\tau]$ to denote the set of first-order formulas over the vocabulary τ. We use FO to denote the set of first-order formulas over $\hat{\tau}$. By an FO-formula $\varphi(x_1, \dots, x_k)$ we refer to a formula whose free variables are exactly $x_1, \dots, x_k$. An FO-formula φ (without a list of variables) may or may not have free variables. The set of free variables of φ is denoted by Free(φ). Given a formula $\varphi(v_{i_1}, ..., v_{i_k})$, a model $\mathfrak{A}$ and a tuple $(a_1, ..., a_k) \in A^k$, we use $\mathfrak{A} \models \varphi(a_1, ..., a_k)$ to denote that φ is true on $\mathfrak{A}$ with respect to the assignment that maps the variable v_{i_j} to the element a_j.

2.2 Arity definite relations

Let A be an arbitrary set. As usual, a k**-tuple** $\overline{a} = (a_1, ..., a_k)$ over A is an element of A^k. Given two tuples $\overline{a} = (a_1, ..., a_k)$ and $\overline{b} = (b_1, ..., b_\ell)$, we will use $\overline{ab}$ to denote the tuple $(a_1, ..., a_k, b_1, ..., b_\ell)$, i.e. the concatenation of the tuples $\overline{a}$ and $\overline{b}$. When $k = 0$, we let ϵ denote the unique 0**-tuple** in $A^k = A^0$. Note that $A^0 = B^0 = \varnothing^0 = \{\epsilon\}$ for all sets A and B. Note also that $\varnothing^k = \varnothing$ for all positive integers k.

Definition 2.1. If k is a non-negative integer, then a k**-ary AD-relation** over a set A is a pair (R, k), where $R \subseteq A^k$. Here 'AD' stands for *arity definite*. We call $(\varnothing, k)$ the **empty** k**-ary AD-relation**.

Example 2.2. To motivate the usage of AD-relations instead of ordinary relations, consider the relations A and A^2, i.e. the full unary and binary relations over the set A. Now the complement of both A and A^2 is just the empty set $\varnothing$. This creates some ambiguity as to what is the complement of an empty set. To solve this, we keep track of the arities for the relations, since then the complement of $(A, 1)$ is $(\varnothing, 1)$ and the complement of $(A^2, 2)$ is $(\varnothing, 2)$.

Given a set A and a non-negative integer k, we will use AD$_k(A)$ to denote the set of all k-ary AD-relations over A. Furthermore we will define

$$\mathrm{AD}(A) := \bigcup_k \mathrm{AD}_k(A).$$

For a non-negative integer k, we will use $\top_k$ (respectively, $\bot_k$) to denote an operator that maps any set A to the AD-relation $\top_k(A) := (A^k, k)$ (respectively, $\bot_k(A) := (\varnothing, k)$). We may also write $\top_k^A$ for $\top_k(A)$ and $\top_0$ simply for $\top_0(A)$, and furthermore we may write $\bot_k^A$ or $\bot_k$ for $\bot_k(A)$. Note that the operators $\top_k$ and $\bot_k$ are proper classes, and furthermore note that $\bot_k^{\emptyset} = \top_k^{\emptyset}$ iff $k \neq 0$. When $T = (R, k)$ is a k-ary AD-relation, we will use $rel(T)$ to denote R and write $ar(T) = k$ to refer to the arity of T.

Definition 2.3. **An AD-structure** is a tuple $(A, T_1, ..., T_k)$, where each T_i is an AD-relation over A. We say that a mapping $g : A \to B$ is an isomorphismn between the two AD-structures $(A, T_1, ..., T_k)$ and $(B, S_1, ..., S_k)$ if $ar(T_i) = ar(S_i)$ for every i and g is an isomorphism between the structures

$$(A, rel(T_1), ..., rel(T_k))$$

and

$$(B, rel(S_1), ..., rel(S_k)).$$

Now, let $k \geq 0$ and consider a FO-formula $\varphi(v_{i_1}, \ldots, v_{i_k})$ where $i_1 < \cdots < i_k$. The formula $\varphi(v_{i_1}, \ldots, v_{i_k})$ **defines** the following AD-relation

$$\big(\{(a_1, \ldots, a_k) \in A^k \mid \mathfrak{A} \models \varphi(a_1, \ldots, a_k)\}, k\big).$$

in the model $\mathfrak{A}$. Note that we make crucial use of the linear ordering of the subindices of the variables $v_{i_1}, \ldots, v_{i_k}$. We will use $[\![\varphi]\!]_{\mathfrak{A}}$ to denote the AD-relation defined by φ in $\mathfrak{A}$.

Example 2.4. Let $\mathfrak{A} = (\{0, 1, 2\}, R^{\mathfrak{A}})$ be a model of vocabulary $\{R\}$, where we interpreted R as $R^{\mathfrak{A}} = \{(2, 0, 1), (1, 1, 2), (2, 0, 0)\}$. The AD-relation defined by the formula $R(v_3, v_1, v_2)$ is

$$\big(\{(0, 1, 2), (1, 2, 1), (0, 0, 2)\}, 3\big).$$

Notice that since we make use of the linear ordering of the variables based on their sub-indices, formulas such as $R(v_1, v_2, v_3)$ and $R(v_4, v_5, v_6)$ define the same AD-relation, while they are not strictly speaking equivalent because they have different variables.

2.3 Relation operators

In this thesis we are going to need the concept of a relational algebra. To this end we define first the concept of a relation operator, which was first introduced in [11], where it was called arity-regular relation operator.

Definition 2.5. A k-ary **relation operator** F is a mapping which associates to each set A a function $F^A : \mathrm{AD}(A)^k \to \mathrm{AD}(A)$ and which satisfies the following requirements.

1. The operator F is isomorphism invariant in the sense that if the two AD-structures $(A, T_1, ..., T_k)$ and $(B, S_1, ..., S_k)$ are isomorphic via g, then also the AD-structures $(A, F^A(T_1, ..., T_k))$ and $(B, F^B(S_1, ..., S_k))$ are likewise isomorphic via g.

2. There exists a function $\sharp : \mathbb{N}^k \to \mathbb{N}$ so that for every AD-structure $(A, T_1, ..., T_k)$ we have that the arity of the AD-relation $F^A(T_1, ..., T_k)$ is $\sharp(ar(T_1), ..., ar(T_k))$. In other words the arity of the output AD-relation is determined by the sequence of arities of the input AD-relations.

Remark. By definition a relation operator is a proper class. We could avoid this by fixing a large enough set X and then agree that the relation operators are defined for every subset of X.

The concept of relation operator is quite general, and hence we will give two examples to demonstrate their rich structure.

Example 2.6. Consider the mapping $\mathrm{H}_A : \mathrm{AD}(A)^2 \to \mathrm{AD}(A)$, which is defined by

$$\mathrm{H}_A((T_1, T_2)) = \begin{cases} \top_0^A & \text{, if } |rel(T_1)| = |rel(T_2)| \\ \bot_0^A & \text{otherwise} \end{cases}$$

Then the mapping $A \mapsto \mathrm{H}_A$ is a relation operator, which is rather similar to the well-known Härtig guantifier, the only difference being that we do not put any restrictions on the arities of T_1 and T_2. This example demonstrates that relation operators can be seen as a generalization of generalized quantifiers.

Example 2.7. Fix $i, j \in \mathbb{Z}_+$ and consider the mapping $\sigma_A^{i,j} : \mathrm{AD}(A) \to \mathrm{AD}(A)$, which is defined by

$$\sigma_A^{i,j}(T_1) = \begin{cases} (\{\overline{a} \in rel(T_1) \mid a_i = a_j\}, ar(T_1)) & \text{, if } i, j \leq ar(T_1) \\ \bot_0^A & \text{otherwise} \end{cases}$$

Then the mapping $A \mapsto \sigma_A^{i,j}$ is a relation operator, which is similar to the selection operator from database theory. It is easy to see that other operators from database theory - such as the projection $\pi^{i_1,\ldots,i_k}$ - can be seen as relation operators.

We will conclude this section by defining relation operators that will be used later in this thesis. First we have the **difference operator** $\setminus$, which we define as the mapping $A \mapsto \setminus_A$, where $\setminus_A : \mathrm{AD}(A)^2 \to \mathrm{AD}(A)$ is defined by

$$\setminus_A(T_1, T_2) = (\{\overline{a} \in rel(T_1) \mid \overline{a} \notin rel(T_2)\}, ar(T_1))$$

Note that if $ar(T_1) \neq ar(T_2)$, then $\setminus_A(T_1, T_2) = T_1$.

Next we define the **suffix intersection operator** $\dot{\cap}$ as the mapping $A \mapsto \dot{\cap}_A$, where $\dot{\cap}_A : \mathrm{AD}(A)^2 \to \mathrm{AD}(A)$ is defined as follows. Given $T_1, T_2 \in \mathrm{AD}(A)$, let $m := \max\{ar(T_1), ar(T_2)\}$. We then define that

$$\begin{aligned} \dot{\cap}_A(T_1, T_2) = \big(\{(a_1, \ldots, a_m) \in A^m \mid\ & (a_{m-ar(T_1)+1}, \ldots, a_m) \in rel(T_1) \\ & \text{and } (a_{m-ar(T_2)+1}, \ldots, a_m) \in rel(T_2)\},\ m\big), \end{aligned}$$

We note that $\dot{\cap}$ can be seen as a special case of the so called semijoin operator in database theory.

As the final two operators we define two restrictions of the suffix intersection operator. The first one is the **one-dimensional intersection operator** C and we define it as the mapping $A \mapsto C_A$, where $C_A : \mathrm{AD}(A)^2 \to \mathrm{AD}(A)$ is defined by

$$C_A(T_1, T_2) = (\{(a_1, ..., a_{ar(T_1)}) \in rel(T_1) \mid a_{ar(T_1)} \in rel(T_2)\}, ar(T_1)),$$

if $ar(T_2) = 1 \leq ar(T_1)$. In the case where $ar(T_2) = 0$ and $ar(T_1) \geq 0$, we define $C_A(T_1, T_2)$ to be T_1 if $T_2 = \top_0^A$, and otherwise $\bot_{ar(T_1)}^A$. In the other cases we define $C_A(T_1, T_2)$ to be $\bot_0^A$.

The second one is the more usual **intersection operator** $\cap$, which is defined as the mapping $A \mapsto \cap_A$, where $\cap_A : \mathrm{AD}(A)^2 \to \mathrm{AD}$ is defined by setting that $\cap_A(T_1, T_2) = \bot_0$, if $ar(T_1) \neq ar(T_2)$, and otherwise

$$\cap_A(T_1, T_2) = (\{\overline{a} \in rel(T_1) \mid \overline{a} \in rel(T_2)\}, ar(T_1)).$$

2.4 General relational algebra

Let τ be a vocabulary and let $\mathcal{F}$ be a collection of relation operators. In the rest of this thesis we will use the convention that if F is a relation operator, then we will also use F to denote a symbol (which will be interpreted according to the relation operator it refers to).

We now define the language $\mathrm{GRA}(\mathcal{F})[\tau]$ as follows, where $R \in \tau$ and $\mathrm{F} \in \mathcal{F}$:

$$\mathcal{T} ::= \bot \mid \top \mid R \mid \mathrm{F}\,\underbrace{(\mathcal{T}, ..., \mathcal{T})}_{ar(\mathrm{F})\text{ times}}.$$

Here we note that we sometimes use the infix notation instead of the prefix notation, if the infix notation is more conventional. For example, instead of writing $\cap(\mathcal{T}, \mathcal{T})$ - where $\cap$ is the relation operator introduced in the previous section - we will write $(\mathcal{T} \cap \mathcal{T})$.

As for FO, we use $\mathrm{GRA}(\mathcal{F})$ to denote $\mathrm{GRA}(\mathcal{F})[\hat{\tau}]$. Elements of $\mathrm{GRA}(\mathcal{F})$ will be referred to as **terms**. If $\mathcal{F}$ is finite $\{F_1, ..., F_n\}$ (which it will be in the rest of this thesis), we will use the shorthand notation $\mathrm{GRA}(F_1, ..., F_n)[\tau]$ for $\mathrm{GRA}(\{F_1, ..., F_n\})[\tau]$.

Definition 2.8. Given a model $\mathfrak{A}$ of vocabulary τ and term $\mathcal{T} \in \mathrm{GRA}(\mathcal{F})[\tau]$, its **interpretation** $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is defined recursively as follows:

1. $[\![\bot]\!]_{\mathfrak{A}} := \bot_0^A$.
2. $[\![\top]\!]_{\mathfrak{A}} := \top_0^A$.
3. $[\![R]\!]_{\mathfrak{A}} := (R^{\mathfrak{A}}, ar(R))$.
4. $[\![\mathrm{F}(\mathcal{T}, ..., \mathcal{T})]\!]_{\mathfrak{A}} := \mathrm{F}_A([\![\mathcal{T}]\!]_{\mathfrak{A}}, ..., [\![\mathcal{T}]\!]_{\mathfrak{A}})$.

Note that the interpretation of every term $\mathcal{T}$ over a model $\mathfrak{A}$ is an AD-relation over the set A. The arity of this AD-relation (for some fixed model $\mathfrak{A}$) is called the arity of the term $\mathcal{T}$ and we will denote it by $ar(\mathcal{T})$. Note that by definition the arity of the output relation is independent of the underlying model, and hence $ar(\mathcal{T})$ is well-defined.

Since the interpretation of every term $\mathcal{T}$ over a model $\mathfrak{A}$ is an AD-relation over the set A, we can compare the expressive power of relational algebras and fragments of first-order logic.

Definition 2.9. Let $\mathcal{F}$ be a set of relation operators. We say that $\mathcal{T} \in \mathrm{GRA}(\mathcal{F})$ and $\varphi \in \mathrm{FO}$ are equivalent, if for every suitable model, i.e. of an appropriate vocabulary, $\mathfrak{A}$ we have that $[\![\varphi]\!]_{\mathfrak{A}} = [\![\mathcal{T}]\!]_{\mathfrak{A}}$.

Definition 2.10. Let $\mathcal{F}$ be a set of relation operators, $S_1 \subseteq \mathrm{GRA}(\mathcal{F})$ and $S_2 \subseteq \mathrm{FO}$. We say that S_1 and S_2 are **equivalent**, if for every term $\mathcal{T} \in S_1$ there exists an equivalent formula $\varphi \in S_2$, and conversely for every $\varphi \in S_2$ there exists an equivalent term $\mathcal{T} \in S_1$. We say that S_1 and S_2 are **sententially equivalent**, if for every 0-ary term $\mathcal{T} \in S_1$ there exists an equivalent sentence $\varphi \in S_2$, and conversely for every sentence $\varphi \in S_2$ there exists an equivalent 0-ary term $\mathcal{T} \in S_1$.

Of course the above definitions could have been formulated more generally for any logic, but since we will focus on studying fragments of first-order logic, this level of generality is enough for our purposes.

The main concern of this thesis is to study relational algebras from the point of view of whether they are decidable. This problem will be defined as follows.

Definition 2.11. Let $\mathcal{G} \subseteq \mathrm{GRA}(F_1, ..., F_n)$. The satisfiability problem for $\mathcal{G}$ takes as an input a term $\mathcal{T} \in \mathcal{G}$ and returns **True** if and only if there exists a model $\mathfrak{A}$ so that $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is not the empty AD-relation.

3 Algebraic characterizations of first-order logic

In this chapter we will give an algebraic characterization of first-order logic by giving a finite set of relation operators $\mathcal{F}$ so that GRA($\mathcal{F}$) is equivalent with FO. As a by-product of this characterization we will also obtain algebraic characterizations for the quantifier-free and equality-free fragments of first-order logic. We will also discuss alternative sets of relation operators which would suffice to capture the expressive power of FO. To conclude this chapter we will give an algebraic characterization of first-order logic using cylindric algebras.

3.1 Characterizing first-order logic as a relational algebra

Consider the set of relation operators $\{e, p, s, \neg, I, J, \exists\}$, where

- $ar(e) = 0$
- $ar(p) = ar(s) = ar(\neg) = ar(I) = ar(\exists) = 1$
- $ar(J) = 2$.

Let A be a set. We then define the interpretation for the operators as follows.

1. $e_A = (\{(a, a) \mid a \in A\}, 2)$.
2. p_A is defined as the mapping for which we have that if $k < 2$, then $p_A((X, k)) = (X, k)$, and otherwise
$$p_A((X, k)) = (\{(a_2, ..., a_k, a_1) \mid (a_1, ..., a_k) \in X\}, k).$$
3. s_A is defined as the mapping for which we have that if $k < 2$, then $s_A((X, k)) = (X, k)$, and otherwise
$$s_A((X, k)) = (\{(a_1, ..., a_{k-2}, a_k, a_{k-1}) \mid (a_1, ..., a_k) \in X\}, k).$$
4. $\neg_A$ is defined as the mapping
$$\neg_A((X, k)) = (A^k \backslash X, k).$$
5. I_A is defined as the mapping for which we have that if $k < 2$, then $I_A((X, k)) = (X, k)$, and otherwise
$$I_A((X, k)) = (\{(a_1, ..., a_{k-1}) \mid (a_1, ..., a_{k-1}, a_{k-1}) \in X, k - 1).$$
6. J_A is defined as the mapping
$$J_A((X, k), (Y, l)) = (X \times Y, k + l).$$
7. $\exists_A$ is defined as the mapping for which we have that if $k < 1$, then $\exists_A((X, k)) = (X, k)$, and otherwise
$$\exists_A((X, k)) = (\{(a_1, ..., a_{k-1}) \mid \text{There exists } b \in A \text{ so that } (a_1, ..., a_{k-1}, b) \in X\}, k - 1).$$

For the rest of this chapter we will assume the convention that GRA denotes the language GRA$(e, p, s, \neg, I, J, \exists)$. The purpose of this section is to show that GRA provides an algebraic characterization of FO in the sense that the two formalism are equivalent.[1]. Before proving this equivalence, it might be useful to see how formulas written in the standard syntax of first-order logic translate to terms in GRA.

Example 3.1. The atomic formula $R(v_3, v_1, v_2)$ can be expressed as the term pR. Observe that although p permutes all the tuples in the relation by moving the first element behind the last element, on the level of atomic formulas this is reflected by making the last variable the first variable.

Example 3.2. The formula $\exists v_1 R(v_1, v_2)$ can be expressed as the term $\exists sR$.

Example 3.3. The formula $R(v_1, v_2) \wedge R(v_2, v_3)$ can be expressed as the term $pIpJ(R, R)$.

The above examples already give a hint as to how one might show that FO is contained in GRA. In particular, they demonstrate that the purpose of the operators p, s and I is to permute the variables in the formulas and identify them with each other. In fact, the most involved part in the proof of the equivalence of GRA and FO is to show that we can form arbitrary permutations of elements of tuples just by using p and s.

Theorem 3.4. GRA *is equivalent with* FO.

Proof. It is straightforward to prove that GRA is contained in FO and so we will focus on proving that FO is contained in GRA. So, let φ be an arbitrary FO-formula. If φ is either $x = x$ or $x = y$, then it can be translated to $\exists e$ and e respectively.

Consider then the case where φ is an atom $R(v_{i_1}, ..., v_{i_k})$. Consider first the case where no variable symbol gets repeated in the tuple $(v_{i_1}, \ldots, v_{i_k})$ and furthermore $i_1 < \cdots < i_k$. Then the term R is clearly equivalent to φ. Consider then the cases where the tuple $(v_{i_1}, \ldots, v_{i_k})$ may contain repetitions and the variables might not be linearly ordered (i.e., $i_1 < \cdots < i_k$ does not necessarily hold). We start by observing that we can permute relations arbitrarily by using the operators p and s; for the sake of completeness, we present here the following steps that prove this basic group-theoretic fact:

- Consider a tuple $(a_1, \ldots, a_i, \ldots, a_\ell)$ of the relation R (in some model $\mathfrak{A}$). We will first describe how we can move any of the elements a_i an arbitrary number n of steps to the right, while keeping the rest of the tuple in the same order as it was.

  1. Repeatedly apply p to the term R, making a_i the rightmost element of the tuple.
  2. After this we apply the *composed* function ps (s first and after this p) precisely n times.
  3. Finally we apply p repeatedly to put the tuple into the desired order.

- We can also move an element a_i to the left in a similar way. Intuitively, we will move a_i to the right and even past the rightmost end of the tuple. Formally, we do this by moving a_i n steps to the left by performing the steps described above so that in step 2, we will apply the composed function ps precisely $\ell - n - 1$ times.

[1]Similar characterization was also provided in the recent preprint [12]

This demonstrates that we can move an arbitrary element anywhere in the tuple, and thereby it is clear that with p and s we can permute an arbitrary tuple in all possible ways.

Since we can permute tuples in every way, we can also deal with the possible repetitions of variables in $R(v_{i_1}, \dots, v_{i_k})$. Indeed, we just need to bring any two elements to the end of a tuple and then apply I. For instance, consider the atom $R(v_1, v_2, v_1)$ (which defines a *binary* relation). We observe that $R(v_1, v_2, v_1)$ is equivalent to the term $pIpR$. Now it should be clear how we can use p, s and I to find an equivalent term for the atom $R(v_{i_1}, \dots, v_{i_k})$.

Suppose then that we already have equivalent terms $\mathcal{T}$ and $\mathcal{P}$ for the formulas φ and ψ. We will now show how to translate $\neg\varphi$, $\varphi \wedge \psi$ and $\exists v_i \varphi$. First, it is clear that $\neg\varphi$ is equivalent with $\neg\mathcal{T}$. The translation of $\varphi \wedge \psi$ can be done in two steps. Suppose that the free variables of φ are $\{v_{i_1}, ..., v_{i_k}\}$ and that the free variables of ψ are $\{v_{i_{k+1}}, ..., v_{i_\ell}\}$. We start by considering the term $J(\mathcal{T}, \mathcal{P})$ which is equivalent with the formula $\varphi(v_1, ..., v_k) \wedge \psi(v_{k+1}, ..., v_\ell)$; notice that we introduced fresh variables. We will now deal with the possible overlapping of variables in the sets $\{v_{i_1}, ..., v_{i_k}\}$ and $\{v_{i_{k+1}}, ..., v_{i_\ell}\}$. This can be done by repeatedly applying p, s and I to $J(\mathcal{T}, \mathcal{P})$ in the very same way as in the case of atoms.

Finally, we will consider the case of $\exists v_i \varphi$. We start by repeatedly applying p to the term $\mathcal{T}$ that correspondence to φ to bring the element to be projected away to the right end of the tuple. Then we will apply $\exists$. After this we will apply repeatedly p to put the term into the final wanted form. □

Using the proof of theorem 3.4 it is easy to show that the quantifier-free FO is characterized by the algebra $GRA(e, p, s, \neg, I, J)$,

Theorem 3.5. *$GRA(e, p, s, \neg, I, J)$ is equivalent with quantifier-free* FO.

Proof. The proof of theorem 3.4 shows that quantifier-free FO can be translated to the algebra GRA$(e, p, s, \neg, I, J)$. For the converse direction the only non-trivial case is to show that if $\mathcal{T}$ can be translated to quantifier-free FO, then also $I\mathcal{T}$ can be translated to quantifier-free FO. Suppose that $\mathcal{T}$ was translated to a formula $\psi(v_1, ..., v_k)$ of quantifier-free FO. First, if $k < 2$, then $I\mathcal{T}$ is equivalent to ψ. Suppose then that $k \geq 2$. Now $I\mathcal{T}$ is equivalent with $\exists v_k(v_{k-1} = v_k \wedge \psi(v_1, ..., v_k))$, which is equivalent with $\psi^*(v_1, ..., v_{k-1})$, where $\psi^*(v_1, ..., v_{k-1})$ is obtained by substituting v_k with v_{k-1}. □

It is also not hard to show that the equality-free FO can be characterized by the algebra $GRA(p, s, \neg, I, J, \exists)$.

Theorem 3.6. *$GRA(p, s, \neg, I, J, \exists)$ is equivalent with equality-free* FO.

3.2 Alternative algebraic characterization for first-order logic

In this section we discuss a slightly different set of algebraic operators that could be used to give algebraic characterization of first-order logic. These operators are closer to the ones that were used in the preprint [11][2].

Consider two relation operators u and $\hat{I}$, for which we have that $ar(u) = 0$ and $ar(\hat{I}) = 2$. Let A be a set. The interpretations for these two operators are defined as follows.

1. $u_A = (A, 1)$.

[2]Similar set of operators was also used in the preprint [16]

2. If $k < 2$, then $\hat{I}_A((X,k)) = (X,k)$, and otherwise

$$\hat{I}_A((X,k)) = (\{(a_1,...,a_{k-1},a_k) \in X \mid a_{k-1} = a_k\}, k).$$

Consider then the algebra $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$. First of all, it is not hard to see that this algebra can be translated to FO. On the other hand, it is not hard to see that $\mathrm{GRA}(e,p,s,I,\neg,J,\exists)$ can be translated to $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$, since e is equivalent with $\hat{I}J(u,u)$, and if $\mathcal{T} \in \mathrm{GRA}(e,p,s,I,\neg,J,\exists)$ is equivalent with $\mathcal{T}' \in \mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$, then $I\mathcal{T}$ is either equivalent with $\mathcal{T}'$ or $\exists\hat{I}\mathcal{T}'$, depending on whether $ar(\mathcal{T}) \geq 2$. Thus based on theorem 3.4 we can conclude the following.

Theorem 3.7. $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$ *is equivalent with* FO.

We will conclude this section by comparing the two algebras $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$ and $\mathrm{GRA}(e,p,s,I,\neg,J,\exists)$[3]. I believe that the latter characterization is more natural than the former one. The main reason for this is that with the latter we can obtain some important fragments of FO from $\mathrm{GRA}(e,p,s,I,\neg,J,\exists)$ by simply dropping some of the operators, while with the former this is no longer the case. As a concrete example, consider the theorems 3.5 and 3.6. We will argue that there does not exist corresponding results for the algebra $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$.

First, if we drop $\exists$ from the algebra $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$, then the resulting algebra is strictly weaker than the quantifier-free FO. To see this, consider the vocabulary $\{R\}$, where R is a binary relation. We claim that there does not exists a term $\mathcal{T} \in \mathrm{GRA}(u,p,s,\hat{I},\neg,J)[\{R\}]$ which is equivalent with $R(x,x)$. Aiming for a contradiction, suppose that there is such a term $\mathcal{T} \in \mathrm{GRA}(u,p,s,\hat{I},\neg,J)[\{R\}]$. If $\mathcal{T}$ does not contain any occurrences of R, then by induction over the structure of $\mathcal{T}$ it is easy to show that over any model $\mathfrak{A}$ we have that $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is either $\bot_1^A$ or $\top_1^A$, i.e. no such term is equivalent to $R(x,x)$. Suppose then that $\mathcal{T}$ contains at least one occurrence of R. Since none of the operators from $\{p,s,\hat{I},\neg,J\}$ decrease the arity of the input term(s), the arity of $\mathcal{T}$ must be greater than two. But clearly no such term can be equivalent to the formula $R(x,x)$. On the other hand it is clear that if we do not drop $\exists$ from the algebra, then it is certainly not equivalent with the quantifier-free FO, since it can then express statements such as $\exists x P(x)$.

Secondly, the equality-free FO cannot be obtained from $\mathrm{GRA}(u,p,s,\hat{I},\neg,J,\exists)$ by dropping some of the operators. If the operator $\hat{I}$ is dropped, then the satisfiability problem for the resulting algebra is NP-complete [11], implying that this fragment is much weaker than the equality-free FO, for which the satisfiability problem is undecidable. On the other hand, if the operator $\hat{I}$ is not dropped, then the resulting algebra is clearly not equivalent with the equality-free FO, since it can then express statements such as $x = y$.

3.3 Cylindric algebra for first-order logic

In this section we characterize FO using cylindric algebras. Our algebras resemble closely the original cylindric algebras of Tarski, with the difference that they employ only a finite algebraic signature. For a background on the cylindric algebras, we refer the reader to [9].

Definition 3.8. Let A be a set. **A cylindric relational algebra over** A is a tuple

$$C(A) = (\mathcal{P}(A^{\mathbb{Z}}), A^{\mathbb{Z}}, \overline{p}^A, p^A, s^A, I^A, \cap^A, \neg^A, \exists^A),$$

which is defined as follows.

[3]Similar comparison was also done in [11].

1. Let $pred : \mathbb{Z} \to \mathbb{Z}$ be the mapping $n \mapsto n - 1$. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$\overline{p}^A(X) := \{f \circ pred \mid f \in X\}.$$

2. Let $suc : \mathbb{Z} \to \mathbb{Z}$ be the mapping $n \mapsto n + 1$. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$p^A(X) := \{f \circ suc \mid f \in X\}.$$

3. Let $swap : \mathbb{Z} \to \mathbb{Z}$ be the mapping which maps 0 to 1, 1 to 0 and fixes everything else. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$s^A(X) := \{f \circ swap \mid f \in X\}.$$

4. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$I^A(X) := \{f \in X \mid f(0) = f(1)\}.$$

5. For every $X, Y \subseteq A^{\mathbb{Z}}$ we have that

$$\cap^A(X, Y) := X \cap Y.$$

6. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$\neg^A(X) := A^{\mathbb{Z}} \backslash X.$$

7. For every $X \subseteq A^{\mathbb{Z}}$ we have that

$$\exists^A(X) := \{f \in A^{\mathbb{Z}} \mid \text{There exists } g \in X \text{ so that } f(n) = g(n), \text{ for every } n \in \mathbb{Z}\backslash\{0\}\}.$$

Let τ be a vocabulary. We define the language CRA$[\tau]$ using the grammar

$$\mathcal{T} ::= \top \mid R \mid \overline{p}\mathcal{T} \mid p\mathcal{T} \mid s\mathcal{T} \mid I\mathcal{T} \mid (\mathcal{T} \cap \mathcal{T}) \mid \neg\mathcal{T} \mid \exists\mathcal{T},$$

where $R \in \tau$. As in the case of FO and GRA we use CRA to denote CRA$[\hat{\tau}]$ (recall that $\hat{\tau}$ denoted the full relational vocabulary).

Definition 3.9. Given a model $\mathfrak{A}$ of vocabulary τ and a term $\mathcal{T} \in$ CRA$[\tau]$, its **interpretation** $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is defined by setting that

$$[\![\top]\!]_{\mathfrak{A}} := A^{\mathbb{Z}}$$

and that for every $R \in \tau$ we have that

$$[\![R]\!]_{\mathfrak{A}} := \{f \in A^{\mathbb{Z}} \mid (f(0), ..., f(ar(R) - 1)) \in R^{\mathfrak{A}}\}$$

and then extending this inductively to other terms using the operators of $C(A)$.

Definition 3.10. Let $\varphi(v_{i_1}, ..., v_{i_k}) \in$ FO$[\tau]$ and $\mathcal{T} \in$ CRA$[\tau]$. We say that φ is equivalent to $\mathcal{T}$, if in every model $\mathfrak{A}$ of vocabulary τ we have that

$$\mathfrak{A} \models \varphi(a_1, ..., a_k) \iff \text{There exists } f \in [\![\mathcal{T}]\!]_{\mathfrak{A}}, \text{ so that } f(v_{i_j} - 1) = a_j, \text{ for every } 1 \leq j \leq k.$$

Notice that the equivalence of a formula and a term depends on what variables occur in the formula. For instance, although $R(v_1, v_2)$ and $R(v_3, v_4)$ define the same AD-relations, the term R is equivalent only with $R(v_1, v_2)$. Note that the term that is equivalent with $R(v_3, v_4)$ is the term ppR.

Example 3.11. To demonstrate the expressive power of CRA, we will show how formulas of FO translate into CRA.

1. The formula $R(v_1, v_2) \wedge R(v_3, v_4)$ is equivalent with the CRA$[\{R\}]$-term $R \cap ppR$.
2. The formula $R(v_2, v_1) \wedge P(v_2)$ is equivalent with the CRA$[\{P, R\}]$-term $sR \cap pP$.
3. The sentence $\exists v_1 \exists v_2 (R(v_1, v_2) \wedge R(v_2, v_1))$, is equivalent with the CRA$[\{R\}]$-term $\exists s \exists s (R \cap sR)$.

Perhaps unsurprisingly, it can be shown that CRA is equivalent with FO in the sense that for every $\mathcal{T} \in$ CRA there exists an equivalent formula $\varphi \in$ FO, and conversely, for every formula $\varphi \in$ FO there exists an equivalent term $\mathcal{T} \in$ CRA. We omit the proof for this theorem, since it is essentially the same as the proof of 3.4.

Theorem 3.12. *For every first-order formula there exists an equivalent term in* CRA*, and conversely for every term in* CRA *there exists an equivalent first-order formula.*

We conclude this section with a discussion on the differences of GRA and CRA. Perhaps the most important difference between the two algebras is that they give a rather different interpretations for the existential quantifier. This difference can be demonstrated by comparing the translations from GRA and CRA to FO. Suppose that we have translated $\mathcal{T}$ to $\psi(v_{i_1}, ..., v_{i_k})$, where $\mathcal{T}$ is either a term of GRA or CRA. In the former, a term of the form $\exists \mathcal{T}$ could be then translated to $\exists v_{i_k} \psi(v_{i_1}, ..., v_{i_k})$, while in the latter, a term of the same form would be translated to $\exists v_0 \psi(v_{i_1}, ..., v_{i_k})$ (note that v_0 might not even occur in ψ). Intuitively, in CRA we are always quantifying the same variable, which is v_0.

Now one can ask which of these two interpretations for existential quantifier is the more natural one. In my opinion, at least for the purpose of this thesis, the interpretation provided by GRA is more natural, since it gives more expressive power to the existential quantifier. For example, in chapter 6 we will study the so-called ordered first-order logic, which can be seen as the algebra GRA$(\neg, \cap, \exists)$, where $\cap$ is the intersection operator (see chapter 6 for a formal definition of the operator $\cap$). This fragment is already rather expressive, since the corresponding satisfiability problem turns out to be PSPACE-complete. If we were to consider the fragment of CRA which contains only the operators $\neg$, $\cap$ and $\exists$, then it is clear from the above discussion that it is a rather weak fragment of FO.

4 Algebraic characterizations for fragments of first-order logic

In this section we will present algebraic characterizations for well-known fragments of first-order logic. More precisely, we will show the following characterizations.

1. Guarded fragment is sententially equiexpressive with the algebra $\mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$.
2. Two-variable logic is equiexpressive with the algebra $\mathrm{GRA}(e, s, \neg, \dot{\cap}, \exists)$ over vocabularies with at most binary relations.
3. Fluted logic is equiexpressive with the algebra $\mathrm{GRA}(\neg, \dot{\cap}, \exists)$.

In all of the cases we are also able to show that the complexity of the decidability problem is the same both for the fragment as well as the algebra which characterizes it.

4.1 Guarded fragment

In this section we consider $\mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ and show that it is sententially equiexpressive with GF. Recall that GF is the logic that has all atoms $R(x_1, \ldots, x_k)$, $x = y$ and $x = x$, is closed under $\neg$ and $\wedge$, but existential quantification is restricted to patterns $\exists x_1 \ldots \exists x_k(\alpha \wedge \psi)$ where α is an atomic formula (a guard) having (at least) all the free variables of $\psi \in \mathrm{GF}$.

We start by defining that a **term guard** of a term $\mathcal{T}$ is a tuple $(\mathcal{S}, i_1, ..., i_k)$, where $k = ar(\mathcal{T})$, with the following properties.

1. $\mathcal{S}$ is a term of the algebra $\mathrm{GRA}(e, p, s)$.
2. $1 \le i_1 < ... < i_k \le m$, where $m = ar(\mathcal{S})$.
3. For every model $\mathfrak{A}$ and a tuple $(a_1, ..., a_k) \in [\![\mathcal{T}]\!]_{\mathfrak{A}}$ there exists a tuple $(b_1, ..., b_m) \in [\![\mathcal{S}]\!]_{\mathfrak{A}}$ so that $b_{i_j} = a_j$, for every $1 \le j \le k$.

Thus $\mathcal{S}$ is e or a relation symbol to which we can also possibly apply p and s for some number of times. To make the internalization of the concept of term guard easier, I recommend that the reader should first take a look at the proof of theorem 4.3.

The following lemma will be used below when translating algebraic terms to formulas of the guarded fragment.

Lemma 4.1. *Every term $\mathcal{T} \in \mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ has a term guard. Furthermore the term guard can be computed from $\mathcal{T}$ in polynomial time.*

Proof. We will define inductively a mapping which maps each term $\mathcal{T} \in \mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ to a term guard for it. Before stating the formal proof, we will first consider a concrete example that might clarify the intuition behind the proof.

Consider the term $p\exists R$, where R is a quaternary relational operator. We will construct step-by-step a term guard for this term. First it is clear that $(R, 1, 2, 3, 4)$ is already a term guard for R. To obtain a term guard for $\exists R$, we will just need to remove the last index from the term guard, i.e. $(R, 1, 2, 3)$ is a term guard for $\exists R$. We can then use $(R, 1, 2, 3)$ as a starting point for constructing a term guard for $p\exists R$ as follows. If a tuple (a, b, c) belongs to the interpretation of $p\exists R$, then (c, a, b) belonged to the interpretation of $\exists R$. Since $(R, 1, 2, 3)$ was a term guard for $\exists R$, there is some element d so that (c, a, b, d) belongs to the interpretation of R. Now (a, b, d, c) belongs to the interpretation of pR, and we see that the original tuple (a, b, c) occurs as a subtuple of this tuple, in the sense that if we remove d then the resulting tuple is (a, b, c). Hence we can conclude that $(pR, 1, 2, 4)$ is a term guard for $p\exists R$.

Now we will proceed with the definition of the mapping. We start by defining that e will be mapped to $(e, 1, 2)$ and that every relation symbol R will be mapped to $(R, 1, ..., ar(R))$. Suppose that we have mapped a k-ary term $\mathcal{T}$ to the term guard $(\mathcal{S}, i_1, ..., i_k)$. Using this term guard as a starting point, we wish to construct term guards for the terms $p\mathcal{T}$ and $s\mathcal{T}$. Consider first the term $p\mathcal{T}$. Denoting by $\mathcal{G}$ the term obtained from $\mathcal{S}$ by applying the operator p i_1-times to it, it is straightforward to verify that $(\mathcal{G}, i_2 - i_1, ..., i_k - i_1, m)$ is a term guard for the term $p\mathcal{T}$, where m was the arity of $\mathcal{S}$. Consider then the term $s\mathcal{T}$. Now we apply p and s to $\mathcal{S}$ as follows. First we apply p $(i_1 - 1)$-times to $\mathcal{S}$. Let the resulting term be $\mathcal{S}'$. Then we apply ps (i.e., first s then p) to $\mathcal{S}'$ $(i_2 - i_1 - 1)$-times, after which we apply s one more time. Denoting the resulting term by $\mathcal{G}$, it is straightforward to verify that $(\mathcal{G}, 1, 2, ..., i_k - (i_2 - 2))$ is a term guard for $s\mathcal{T}$.

Other cases are much simpler. Suppose that we have mapped a k-ary term $\mathcal{T}$ to the term guard $(\mathcal{S}, i_1, ..., i_k)$ and the ℓ-ary term $\mathcal{P}$ to the term guard $(\mathcal{S}', j_1, ..., j_\ell)$. Suppose furthermore that $k \geq \ell$. Then $\mathcal{T} \dot{\cap} \mathcal{P}$ and $\mathcal{P} \dot{\cap} \mathcal{T}$ will both be mapped to $(\mathcal{S}, i_1, ..., i_k)$. If $k \neq \ell$, then $(\mathcal{T} \backslash \mathcal{P})$ and $(\mathcal{P} \backslash \mathcal{T})$ will be mapped to $(\mathcal{S}, i_1, ..., i_k)$ and $(\mathcal{S}', j_1, ..., j_\ell)$ respectively, while if $k = \ell$, then both of them will be mapped to $(\mathcal{S}, i_1, ..., i_k)$. Note here that if the arities of $\mathcal{T}$ and $\mathcal{P}$ differ, then by definition $\mathcal{T} \backslash \mathcal{P}$ and $\mathcal{P} \backslash \mathcal{T}$ are equivalent with $\mathcal{T}$ and $\mathcal{P}$ respectively. Finally $\exists \mathcal{T}$ will be mapped to $(\mathcal{S}, i_1, ..., i_{k-1})$.

This completes the definition of the mapping. Since the mapping is clearly computable in polynomial time, the claim follows. □

We will also make use of the following lemma which implies that we can use I in $\mathrm{GRA}(e, p, s, \backslash, \dot{\cap}, \exists)$.

Lemma 4.2. *The operator I can be expressed with $e, \exists$ and $\dot{\cap}$, i.e., any term $I(\mathcal{T})$ has an equivalent term built from $\mathcal{T}$ by using e, $\exists$ and $\dot{\cap}$.*

Proof. Consider the term $I(\mathcal{T})$. If $ar(\mathcal{T}) > 1$, then $I(\mathcal{T})$ is expressed as the term $\exists(e \dot{\cap} \mathcal{T})$. If $ar(\mathcal{T})$ is 1 or 0, then $I(\mathcal{T})$ is equivalent to $\mathcal{T}$. □

We can now prove the following.

Theorem 4.3. $\mathrm{GRA}(e, p, s, \backslash, \dot{\cap}, \exists)$ *and* GF *are sententially equiexpressive.*

Proof. We will first show that for every (possibly open) formula $\exists x_1 \ldots \exists x_k\, \psi$ of GF, there exists an equivalent term $\mathcal{T}$ of $\mathrm{GRA}(e, p, s, \backslash, \dot{\cap}, \exists)$. Let us begin by showing this for a formula $\varphi := \exists x_1 \ldots \exists x_k\, \psi$ where ψ is of the form

$$\alpha(y_1, \ldots, y_n) \wedge \beta(z_1, \ldots, z_m)$$

where $\alpha(y_1, \ldots, y_n)$ is an atom, $\beta(z_1, \ldots, z_m)$ is quantifier-free and we have $\{z_1, \ldots, z_m\} \subseteq \{y_1, \ldots, y_n\}$ and $\{x_1, \ldots, x_k\} \subseteq \{y_1, \ldots, y_n\}$.

Now consider a conjunction $\alpha \wedge \rho$ where $\alpha = \alpha(y_1, \ldots, y_n)$ is an atomic formula and ρ an arbitrary atom whose set of variables is a subset of $\{y_1, \ldots, y_n\}$. We call such a conjunction an *α-guarded atom*. For each α-guarded atom, we can find an equivalent term as follows. First, recall from the proof of Theorem 3.4 that for any atom of vocabulary τ there exists an equivalent term in $\mathrm{GRA}(p, s, I)$. Since we can use I in $\mathrm{GRA}(e, p, s, \backslash, \dot{\cap}, \exists)$, it follows that there exists a term

$\mathcal{T}_\rho \in \mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ which is equivalent with ρ. Concerning α, we note that since $x = y$ and $x = x$ are equivalent to the terms e and $\exists e$, there also exists a term $\mathcal{T}_\alpha \in \mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ which is equivalent with α. Now, the term $\mathcal{T} \dot{\cap} \mathcal{S}$ is most likely not equivalent with $\alpha \wedge \rho$, as the variables in $\alpha \wedge \rho$ can be unfavourably ordered instead of matching each other in the desired way. However, we can first permute $\mathcal{T}_\alpha$ to match $\mathcal{T}_\rho$ at the last coordinates of tuples, since one can use p and s to generate arbitrary permutations, and then combine the terms with $\dot{\cap}$, and finally we can permute the obtained term to the final desired form. In this way we obtain an equivalent term for an arbitrary α-guarded atom.

Now recall the original formula $\alpha(y_1, \ldots, y_n) \wedge \beta(z_1, \ldots, z_m)$ from above. For each atom γ which occurs in β, let $\mathcal{T}_\gamma^\alpha$ denote the term equivalent to the α-guarded atom $\alpha \wedge \gamma$. The formula β is a Boolean combination composed from atoms by using $\neg$ and $\wedge$. We let $\mathcal{T}_\beta$ denote the term obtained from β by replacing each atom γ by the corresponding term $\mathcal{T}_\gamma^\alpha$, each $\wedge$ by $\dot{\cap}$ and each $\neg$ by relative complementation with respect to $\mathcal{T}_\alpha$ (i.e., formulas $\neg\,\eta$ will be replaced by the terms $\mathcal{T}_\alpha \setminus \eta^*$ where η^* denotes the translation of η). It is easy to show that $\mathcal{T}_\beta$ is indeed equivalent to $\alpha(y_1, \ldots, y_n) \wedge \beta(z_1, \ldots, z_m)$. Thus we can apply p and $\exists$ in a suitable way to the term $\mathcal{T}_\beta$ to get a term equivalent with the initial formula $\varphi = \exists x_1 \ldots \exists x_k(\alpha(y_1, \ldots, y_n) \wedge \beta(z_1, \ldots, z_m))$.

Thus we managed to translate φ. To get the full translation, we can just keep repeating the procedure described above. The only difference is that in addition to atoms, the above formula β can now contain formulas of the form $\exists x_1 \ldots \exists x_r(\delta \wedge \eta)$. Proceeding inductively, we get a term equivalent to $\exists x_1 \ldots \exists x_r(\delta \wedge \eta)$ by the induction hypothesis, and otherwise we proceed similarly as described above. This concludes the argument for translating formulas to terms.

Let us then consider how to translate terms into equivalent formulas of GF. The proof proceeds by induction. Since GF is closed under Boolean operators, the only non-trivial case is the translation of the projection operator $\exists$. So suppose that we have translated $\mathcal{T}$ to $\psi(x_1, \ldots, x_k)$. By lemma 4.1, we can find a term guard $(\mathcal{S}, i_1, ..., i_k)$ for $\mathcal{T}$. Since $\mathcal{S} \in \mathrm{GRA}(e, p, s)$, we can translate $\mathcal{S}$ to an atomic formula $\alpha(v_1, ..., v_m)$. Recalling the definition of term guard, we see that $\mathcal{T}$ is equivalent to $\exists\overline{v}(\alpha(v_1, ..., v_m) \wedge \psi(v_{i_1}, ..., v_{i_k}))$, where $\overline{v}$ contains those variables from $\{v_1, ..., v_m\}$ that are different from $\{v_{i_1}, ..., v_{i_k}\}$. Hence we can conclude that $\exists\mathcal{T}$ is equivalent to $\exists v_{i_1}\exists\overline{v}(\alpha(v_1, ..., v_m) \wedge \psi(v_{i_1}, ..., v_{i_k}))$.

□

It is easy to see that our translations in the above proof are computable in polynomial time. Since the satisfiability problem for GF is 2ExpTime-complete [5], we have the following corollary.

Corollary 4.4. *The satisfiability problem for* $\mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ *is 2ExpTime-complete.*

We note that in [17] the authors define, using a semijoin operator, a Codd-style relational algebra *with an infinite signature* which they then prove to be sententially equiexpressive with GF. Since $\dot{\cap}$ can be seen as a special case of the semijoin operator, our characterization of GF bears some similarity to their characterization. However our proofs differ considerably. One of the main differences is that the translation we give from $\mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$ to GF is polynomial, while the corresponding translation given in [17] is in general exponential.

Our proof also gives something slightly stronger than just sentential equivalence: it shows that each (possibly open) formula $\exists x_1 ... \exists x_k \psi$ of GF is already equivalent to a term in the algebra $\mathrm{GRA}(e, p, s, \setminus, \dot{\cap}, \exists)$. One can't hope for a much stronger result, since using lemma 4.1 it is easy to see that our algebra is not equivalent to GF. Indeed, if there were a term equivalent with the GF formula $\neg R(x, y)$, then clearly that term could not have a term guard.

4.2 Two-variable logic

Here we establish that $\mathrm{GRA}(e, s, \neg, \dot{\cap}, \exists)$ is equivalent to the two-variable fragment of first order logic FO^2 when we restrict attention to vocabularies that consist of relation symbols of arity at most two. By $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ we denote the set of terms of $\mathrm{GRA}(e, s, \neg, \dot{\cap}, \exists)$ that use at most binary relation symbols, but the arity of the terms themselves is not restricted. However it is easy to see that the terms of $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ have arity at most two, since none of the operators increase the arity of the input terms.

Theorem 4.5. $\mathrm{GRA}(e, s, \neg, \dot{\cap}, \exists)$ *and* FO^2 *are equivalent over vocabularies with at most binary relation symbols.*

Proof. As $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ contains only terms of arity at most two, it is easy to see that $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ translates into FO^2.

We then consider the converse translation. We assume that FO^2 is built using $\neg$, $\wedge$ and $\exists$ and treat other connectives and $\forall$ as abbreviations in the usual way. In what follows we let $v \in \{x, y\}$ denote a generic variable.

Atoms of the form $P(v)$ (respectively $v = v$) translate to P (respectively $\exists e$). Relation symbols of arity 0 translate to themselves and

1. $R(x, y)$ translates to R,
2. $R(y, x)$ translates to sR,
3. $R(v, v)$ translates to $\exists(R \dot{\cap} e)$,
4. $x = y$ and $y = x$ translate to the term e.

Now suppose we have translated ψ to $\mathcal{T}$. Then $\neg\psi$ translates to $\neg\mathcal{T}$. If ψ has one free variable v, then $\exists v\psi$ translates to $\exists\mathcal{T}$. If ψ has two free variables, then we either translate $\exists v\psi$ to $\exists\mathcal{T}$ when v is x and to $\exists s\mathcal{T}$ when v is y.

Consider now a formula $\psi \wedge \chi$ and suppose that we have translated ψ to $\mathcal{T}$ and χ to $\mathcal{S}$. We have now different cases based on how the variables of ψ and χ overlap. First of all, if at least one of ψ and χ is a sentence, we simply translate $\psi \wedge \chi$ to $(\mathcal{T} \dot{\cap} \mathcal{S})$. Consider than the case where $\mathrm{Free}(\psi) \cap \mathrm{Free}(\chi) = \emptyset$. Without loss of generality we assume that $\mathrm{Free}(\psi) = \{x\}$ and $\mathrm{Free}(\chi) = \{y\}$. Now we translate $\psi \wedge \chi$ to the term

$$s(s((e\dot{\cup}\neg e)\dot{\cap}\mathcal{S})\dot{\cap}\mathcal{T}),$$

where $(e\dot{\cup}\neg e)$ is shorthand for $\neg(\neg e\dot{\cap}e)$. Finally we consider the case where $\mathrm{Free}(\psi)\cap\mathrm{Free}(\chi) \neq \emptyset$. Now $\psi(x) \wedge \chi(x), \psi(y) \wedge \chi(y), \psi(x, y) \wedge \chi(x, y), \psi(y) \wedge \chi(x, y)$ and $\psi(x, y) \wedge \chi(y)$ are all translated to $\mathcal{T} \dot{\cap} \mathcal{S}$, while $\psi(x, y) \wedge \chi(x)$ and $\psi(x) \wedge \chi(x, y)$ are translated to $s(s\mathcal{T} \dot{\cap} \mathcal{S})$ and $s(\mathcal{T} \dot{\cap} s\mathcal{S})$ respectively. □

Our above translation of FO^2 to $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ is clearly polynomial, and thus we obtain the following corollary from the fact that the complexity of the satisfiability problem for FO^2 is NEXPTIME-complete [6].

Corollary 4.6. *The satisfiability problem of* $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ *is NEXPTIME-complete.*

We note here that while the algebra $\mathrm{GRA}_2(e, s, \neg, \dot{\cap}, \exists)$ is equivalent with FO^2, the algebra $\mathrm{GRA}_2(s, I, \neg, \dot{\cap}, \exists)$ is no longer equivalent with equality-free FO^2. To see this, consider the vocabulary $\{P\}$, where P is unary. A straightforward induction reveals that for every term $\mathcal{T}$ in the algebra $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)[\{P\}]$ we have that $ar(\mathcal{T}) \leq 1$. Hence no term of $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)[\{P\}]$ is equivalent to the formula $P(x) \wedge P(y)$. However, one can still show that $\mathrm{GRA}_2(s, I, \neg, \dot{\cap}, \exists)$ is *sententially* equivalent with equality-free FO^2.

Theorem 4.7. $\mathrm{GRA}_2(s, I, \neg, \dot{\cap}, \exists)$ *and equality-free* FO^2 *are sententially equivalent.*

Proof. Again, $\mathrm{GRA}_2(s, I, \neg, \dot{\cap}, \exists)$ contains only terms of arity at most two, and hence it is easy to translate it into equality-free FO^2. For the converse direction we note that the only reason why we can't use the proof of theorem 4.5 is that we can't translate formulas of the form $\psi(x) \wedge \chi(y)$ (or $\psi(x) \vee \chi(y)$). However, if we are translating a *sentence* φ of FO^2 to $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)$, then we can replace φ with an equivalent sentence φ^* which does not contain subformulas of type $\psi(x) \wedge \chi(y)$ as follows.

Consider any subformula $\exists x \eta(x, y)$ of φ, where $\eta(x, y)$ is quantifier-free. Put η into disjunctive normal form and distribute $\exists x$ over the disjunctions. Then distribute $\exists x$ also over the conjunctions as follows. Consider a conjunction $\alpha_i(x, y) \wedge \beta_i(y) \wedge \gamma_i$ where α_i, β_i and γ_i are conjunctions of literals; the formula γ_i contains the nullary relation symbols and $\alpha_i(x, y)$ contains the literals of type $\pi(x, y)$ and $\pi'(x)$. We distributed $\exists x$ into $\alpha_i(x, y) \wedge \beta_i(y) \wedge \gamma_i$ so that we obtain the formula $\exists x \alpha_i(x, y) \wedge \beta_i(y) \wedge \gamma_i$. Thereby the formula $\exists x \eta(x, y)$ gets modified into the formula $\bigvee_{i=1}^{n}(\exists x \alpha_i(x, y) \wedge \beta_i(y) \wedge \gamma_i)$ which is of the right form and does not have x as a free variable. Next we can repeat this process for other existential quantifiers in the formula (by treating the subformulas with one free variable in the way that atoms with one free variable were treated in the translation step for $\eta(x, y)$ described above). Having started from the sentence φ, we ultimately get a sentence that does not have subformulas of the form $\psi(x) \wedge \chi(y)$ or of the form $\psi(x) \vee \chi(y)$ but is nevertheless equivalent to φ. □

Clearly our translation from equality-free FO^2 to $\mathrm{GRA}_2(s, I, \neg, \dot{\cap}, \exists)$ is not polynomial, and in fact it is not too hard to show that any such translation necessarily leads to an exponential blow-up in the size of terms. The first observation to make is that if we restrict our attention to vocabularies that contain only unary relation symbols, then the terms of $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)$ translate to sentences of the same length in the so-called Bernays-Schönfinkel-Ramsey fragment BSR of FO. Recall that BSR consists of sentences of the form $\exists y_1 \ldots \exists y_n \forall x_1 \ldots \forall x_m \psi$, where ψ is quantifier-free (possibly with equality).

Lemma 4.8. *Let τ be a vocabulary that consists only of unary relation symbols. For every* 0*-ary term $\mathcal{T} \in \mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)[\tau]$ there exists an equivalent sentence φ in BSR of the same length.*

Proof. First we note that if τ is a vocabulary that consists solely of unary relation symbols, then we can eliminate s and I from the terms of $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)[\tau]$, since the arity of each term is at most one. For the same reason we can replace instances of $\dot{\cap}$ with the operator C. Thus we can restrict attention to terms of the algebra $\mathrm{GRA}(\neg, C, \exists)[\tau]$.

Next we note the following equivalences. If $ar(\mathcal{S}) = 0$, then the term $\exists C(\mathcal{P}, \mathcal{S})$ is either equivalent to $C(\exists \mathcal{P}, \mathcal{S})$ or $C(\mathcal{P}, \mathcal{S})$. Similarly, if we use $\forall$ as a short hand for $\neg\exists\neg$, then $\forall C(\mathcal{P}, \mathcal{S})$ is either equivalent to $C(\forall \mathcal{P}, \mathcal{S})$ or $C(\mathcal{P}, \mathcal{S})$. In the case where $ar(\mathcal{S}) = 1$, we note that if $ar(\mathcal{P}) = 0$, then $C(\mathcal{P}, \mathcal{S})$ is equivalent with $\perp$.

Applying the above equivalences to $\mathcal{T}$, together with the fact that $\exists\neg\mathcal{P}$ and $\forall\neg\mathcal{P}$ are equivalent with $\neg\forall\mathcal{P}$ and $\neg\exists\mathcal{P}$ respectively, we can push all the operators $\exists$ and $\forall$ to the bottom of the term, which will convert $\mathcal{T}$ into an equivalent term which is a boolean combination of terms of the form $\exists\mathcal{P}$ and $\forall\mathcal{P}$, where $\mathcal{P}$ is a unary term in $\mathrm{GRA}(\neg, C)[\tau]$. It is easy to translate such terms into sentences of BSR of the same length. □

Now, there are couple of different ways to show that there does not exists a polynomial translation from sentences of equality-free FO^2 to sentences of BSR over unary vocabularies. One way to prove this is to show that over such vocabularies FO^2 can enforce models of exponential size[1]. Now it is well-known that each sentence $\exists y_1 ... \exists y_n \forall x_1 ... \forall x_m \psi$ of BSR has a model of size at most $\max\{1, n\}$, see for example Proposition 6.2.17 in [3]. From this it follows immediately that there can be no polynomial translation from equality-free FO^2 to BSR.

However, here we prefer to use the following result of [27].

Proposition 4.9. *For every $n \geq 1$, if φ is a sentence of* BSR *that is equivalent with the sentence*

$$\varphi_n := \forall x \exists y \bigwedge_{i=1}^{n} (P_i(x) \leftrightarrow Q_i(y)),$$

then φ contains at least 2^n existential quantifiers.

Proof. Let $n \geq 1$ and suppose that φ is a sentence of BSR that is equivalent with φ_n. Aiming for a contradiction, suppose that φ contains less than 2^n existential quantifiers. Now, using a set A of size 2^n as a domain, we will construct a specific model $\mathfrak{A}$ of φ_n as follows. First, let us fix two bijections $f, g : A \to \{0,1\}^n$ so that $f(a) \neq g(a)$, for every $a \in A$. Then, for every P_i and Q_i we define that

$$P_i^{\mathfrak{A}} = \{a \in A \mid v_i = 1, \text{ when } f(a) = (v_1, ..., v_n)\}$$

and

$$Q_i^{\mathfrak{A}} = \{a \in A \mid v_i = 1, \text{ when } g(a) = (v_1, ..., v_n)\}.$$

Clearly $\mathfrak{A}$ is a model of φ_n. On the other hand, it is easy to see that if we remove any element from $\mathfrak{A}$, then the resulting model is no longer a model of φ_n. Now, since φ is equivalent with φ_n, we have that $\mathfrak{A} \models \varphi$. Since φ contains less than 2^n leading existential quantifiers, there exists an element $a \in A$ so that if we remove it from the model $\mathfrak{A}$, the resulting model will still satisfy φ. This is a contradiction since the resulting model can't satisfy φ_n. □

Corollary 4.10. *There exists a family of sentences $\{\varphi_n \mid n \in \mathbb{Z}_+\}$ of equality-free* FO^2 *so that for every $n \geq 1$, the length of φ_n is linear with respect to n, but every term of* $\mathrm{GRA}(s, I, \neg, \dot{\cap}, \exists)$ *that is equivalent with φ_n has size at least 2^n.*

4.3 Fluted logic

The fluted logic FL is a fragment of FO that was originally considered by Quine in relation to his work on the predicate functor logic [23], but it has also received some attention recently. For the (somewhat unfortunate) history of FL, we recommend the introduction of the article [21].

Definition 4.11. Let $\overline{v}_\omega = (v_1, v_2, ...)$ and let τ be a vocabulary. For every $k \in \mathbb{N}$, we define sets $\mathrm{FL}^k[\tau]$ as follows.

1. Let R be an n-ary relation symbol in τ and consider the gap-free subsequence

$$(v_{k-n+1}, \ldots, v_k)$$

of $\overline{v}_\omega$ containing precisely n variables. Then $R(v_{k-n+1}, \ldots, v_k) \in \mathrm{FL}^k[\tau]$.

[1] See for example the proof of Theorem 6.2.13 in [3].

2. For every $\varphi, \psi \in \mathrm{FL}^k$, we have that $\neg\varphi, (\varphi \wedge \psi) \in \mathrm{FL}^k[\tau]$.

3. If $\varphi \in \mathrm{FL}^{k+1}$, then $\exists v_{k+1}\varphi \in \mathrm{FL}^k[\tau]$.

Finally, we define the set of fluted formulas over τ to be $\mathrm{FL}[\tau] := \bigcup_k \mathrm{FL}^k[\tau]$.

We note here that remarkably the complexity of the decision problem for the fluted logic is non-elementary. For the proof of the following theorem and the precise definition of the related complexity class, see [21].

Theorem 4.12. *The satisfiability problem of* FL *is* Tower*-complete.*

It turns out that fluted logic has a very clean algebraic characterization.

Theorem 4.13. FL *and* $\mathrm{GRA}(\neg, \dot{\cap}, \exists)$ *are equiexpressive.*

Proof. We first translate formulas to algebraic terms. Atomic formulas $R(v_{k-n+1}, \ldots, v_k)$ are translated to R. Note that when R has arity 0, then $R(v_{k-0+1}, v_k)$ of course denotes the formula R (which translates to the term R). Suppose then that $\neg\varphi, (\varphi \wedge \psi) \in \mathrm{FL}^k$ and that we have translated φ to $\mathcal{T}$ and ψ to $\mathcal{S}$. We translate $\neg\varphi$ to $\neg\mathcal{T}$. Now, observe that if $\alpha \in \mathrm{FL}^k$, then the free variables of α form some suffix of the sequence $(v_1, \ldots, v_k)$. Thus we can translate $(\varphi \wedge \psi)$ to $(\mathcal{T} \dot{\cap} \mathcal{S})$. Finally, if $\exists v_{k+1}\varphi \in \mathrm{FL}^k$ and φ translates to $\mathcal{T}$, then we can translate $\exists v_{k+1}\varphi$ to $\exists\mathcal{T}$.

We then translate algebraic terms into fluted logic. An easy way to describe the translation is by giving a family of translations $f_{v_m,\ldots,v_k}$ where $(v_m, \ldots, v_k)$ is a suffix of $(v_1, \ldots, v_k)$. (It is also possible that $f_{v_m,\ldots,v_k} = f_{v_{k+1},v_k}$ which happens precisely when translating a term of arity zero.) The translations are as follows.

1. $f_{v_m,\ldots,v_k}(R) := R(v_m, \ldots, v_k)$ for $ar(R) = k - m + 1$. (When $ar(R) = 0$, then R translates to R.)

2. $f_{v_m,\ldots,v_k}(\neg\mathcal{T}) := \neg f_{v_m,\ldots,v_k}(\mathcal{T})$ for $ar(\mathcal{T}) = k - m + 1$.

3. $f_{v_m,\ldots,v_k}(\mathcal{T} \dot{\cap} \mathcal{S}) := f_{v_n,\ldots,v_k}(\mathcal{T}) \wedge f_{v_\ell,\ldots,v_k}(\mathcal{S})$, where $ar(\mathcal{T} \dot{\cap} \mathcal{S}) = k - m + 1$, $ar(\mathcal{T}) = k - n + 1$ and $ar(\mathcal{S}) = k - l + 1$.

4. $f_{v_m,\ldots,v_k}(\exists\mathcal{T}) = \exists v_{k+1} f_{v_m,\ldots,v_{k+1}}(\mathcal{T})$ for $ar(\exists\mathcal{T}) = k - m + 1$.

□

5 Fragments of $\mathrm{GRA}(e, p, s, \neg, I, J, \exists)$

In this section we will study the complexity of the decidability problem for fragments of $\mathrm{GRA}(e, p, s, \neg, I, J, \exists)$ obtained by removing one of the operators. For most of the resulting fragments we are able to determine whether they are decidable, the principal open case being the fragment $\mathrm{GRA}(e, s, \neg, I, J, \exists)$. In all of the cases where we are able to determine whether a fragment is decidable or not, we are also able to obtain tight complexity bounds.

5.1 Decidable fragments

Our first result concerns the algebra $\mathrm{GRA}(e, p, s, I, J, \exists)$. This fragment is trivially decidable, since every term in it is satisfiable. Nevertheless, this system has the following interesting property concerning conjunctive queries[1].

Theorem 5.1. $\mathrm{GRA}(e, p, s, I, J, \exists)$ *is equiexpressive with the set of conjunctive queries with equality, and* $\mathrm{GRA}(p, s, I, J, \exists)$ *is equiexpressive with the set of conjunctive queries.*

Proof. This follows immediately from the proof of theorem 3.4. □

Now, the next results concern the algebra $\mathrm{GRA}(e, p, s, I, \neg, J)$. By theorem 3.5, we know that the algebra is equivalent with the quantifier-free fragment of first-order logic. Furthermore the translations are clearly computable in polynomial time. From the well-known folklore result that the satisfiability problem for quantifier-free first-order logic is NP-complete, we can conclude that it also NP-complete for the above algebra.

Corollary 5.2. *The satisfiability problem for* $\mathrm{GRA}(e, p, s, I, \neg, J)$ *is NP-complete.*

We can sharpen this by showing that the satisfiability problem is NP-hard already for a small fragment of the algebra.

Theorem 5.3. *The satisfiability problem for* $\mathrm{GRA}(I, \neg, J)$ *is NP-complete.*

Proof. The upper bound follows from the above corollary. For the lower bound we give a reduction from the boolean satisfiability problem. Let φ be a sentence of propositional logic and let $\{p_1, ..., p_n\}$ be the set of propositional symbols in φ. Let $\{P_1, ..., P_n\}$ be a set of unary relation symbols. We translate φ to an equisatisfiable term $\mathcal{T}$ of $\mathrm{GRA}(I, \neg, J)$ as follows. Every p_i translates to P_i. If ψ is translated to $\mathcal{T}$, then $\neg\psi$ is translated to $\mathcal{T}$. If ψ is translated to $\mathcal{T}$ and χ is translated to $\mathcal{P}$, then $(\psi \wedge \chi)$ is translated to $J(\mathcal{T}, \mathcal{P})$. Now let $\mathcal{T}$ be the final term resulting from this translation. Then it is easy to verify that φ is equisatisfiable with the term $I^{ar(\mathcal{T})}\mathcal{T}$, where I^n denotes the sequence $\underbrace{I...I}_{n \text{ times}}$. □

We will next consider the join-free fragment $\mathrm{GRA}(e, p, s, I, \neg, \exists)$, which turns out to be quite tame.

[1] A conjunctive query is a formula of the form

$$\exists v_{i_1} ... \exists v_{i_k} \bigwedge_{j \in J} \alpha_j,$$

where each α_j is an atomic formula.

Theorem 5.4. *Satisfiability of* GRA$(e, p, s, I, \neg, \exists)$ *can be checked by a finite automaton.*

Proof. Observe that every term in the algebra is built either by starting from the constant e or from some relation symbol R, and then using the remaining operators $p, s, I, \neg$ and $\exists$. We consider the two cases separately.

Consider first terms that are built up starting from some relation symbol R, say one which has arity k. Define two $\{R\}$-models $\mathfrak{A}$ and $\mathfrak{B}$, both having the same singleton domain $A = \{a\}$ but with $R^{\mathfrak{A}} = \top_k^A$ and $R^{\mathfrak{B}} = \bot_k^A$. In any model $\mathfrak{M}$ with a singleton domain, every n-ary term $\mathcal{T}$ of GRA can receive only two interpretations, $\bot_n^M$ or $\top_n^M$. Using this observation and induction over the structure of terms $\mathcal{T}$ formed from R with $p, s, I, \neg, \exists$, we can show that $\mathcal{T}^{\mathfrak{A}} = \bot_n^A$ iff $(\neg\mathcal{T})^{\mathfrak{B}} = \bot_n^A$. This implies in particular that every such term $\mathcal{T}$ is satisfiable, since if $\mathcal{T}^{\mathfrak{A}} = \bot_n^A$, then $(\neg\mathcal{T})^{\mathfrak{B}} = \bot_n^A$, and thus $\mathcal{T}^{\mathfrak{B}} = \top_n^A$.

Consider then the terms that are built up from e by using $p, s, I, \neg$ and $\exists$. If $\mathcal{T}$ is such a term, then it is of the following form

$$f_1 ... f_n g h_1 ... h_m(e),$$

where $\{h_1, ..., h_m\} \subseteq \{p, s, \neg\}$, while $\{f_1, ..., f_n, g\} \subseteq \{p, s, \neg, I, \exists\}$. Note that if I and $\exists$ do not occur in $\mathcal{T}$, then it is always satisfiable. Thus we can assume that $g \in \{I, \exists\}$. Now we have two cases based on whether g is I or $\exists$.

First, if $g = I$, then we see that $gh_1...h_m(e)$ is $\top_1^A$ (over any model $\mathfrak{A}$) if $h_1...h_m$ has an even number of negations, and otherwise it is $\bot_1^A$. Thus $f_1...f_n g h_1...h_m(e)$ is satisfiable if $h_1...h_m$ and $f_1...f_n$ have an even number of negations, or if $h_1...h_m$ and $f_1...f_n$ have an odd number of negations.

Suppose then that $g = \exists$. Since over models of size one $Ih_1...h_m(e)$ is equivalent with $gh_1...h_m(e)$, this case can be handled similarly as the previous case. On the other hand, if we restrict attention to models $\mathfrak{A}$ of size at least two, then $\exists h_1...h_m(e)$ is always $\top_1^A$, and hence over such models $f_1...f_n g h_1...h_m(e)$ is satisfiable if and only if $f_1...f_n$ contains an even number of negations.

Using the above observations, it is routine to construct a finite automaton for solving the satisfiability problem of GRA$(e, p, s, I, \neg, \exists)$. □

As the final result of this section we show that GRA without I is NP-complete. We start by defining a certain fragment $\mathcal{F}$ of FO as follows.

1. $x_1 = x_2 \in \mathcal{F}$, for all variables x_1 and x_2.
2. $R(x_1, ..., x_n) \in \mathcal{F}$, for all relation symbols R and variables $x_1, ..., x_n$.
3. If $\varphi, \psi \in \mathcal{F}$ and $\mathrm{Free}(\varphi) \cap \mathrm{Free}(\psi) = \emptyset$, then $(\varphi \wedge \psi) \in \mathcal{F}$.
4. If $\varphi \in \mathcal{F}$, then $\neg\varphi \in \mathcal{F}$ and $\exists x \varphi \in \mathcal{F}$, for any variable x.

We start by showing that the satisfiability problem for $\mathcal{F}$ is in NP. The main idea in our proof is to reduce nondeterministically the satisfiability problem of $\mathcal{F}$ to the satisfiability problem of the *Herbrand fragment*. This latter fragment of FO consists of sentences of the form

$$Qx_1 ... Qx_n \bigwedge_i \eta_i,$$

where each η_i is either an atomic formula or a negation of such. It is known that the satisfiability problem for the Herbrand fragment *without equality* is P-complete, see Theorem 8.2.6 in [3]. In the same book the authors also state (without a proof) that the same holds for the full Herbrand fragment. Since I was unable to find a proof of this result in the literature, I will prove it here.

Lemma 5.5. *Checking whether a sentence in the Herbrand fragment has a model of size one can be done in polynomial time.*

Proof. Over models of size one, every sentence of the Herbrand fragment is equivalent to a sentence of the form $\exists x \bigwedge_i \eta_i$, where each η_i is either an atomic formula or a negation of such. To see this, note first that over models of size one we can replace each universal quantifier with an existential quantifier. After this we can replace the prefix $\exists x_1 ... \exists x_n$ with $\exists x$, since $x_i = x_j$ holds for every $1 \leq i, j \leq n$ over models of size one. Now, if the sentence contains the formula $\neg x = x$, then it is unsatisfiable. Otherwise such a sentence is satisfiable iff it does not contain two complementary formulas η_i and $\neg \eta_i$, which can be easily verified in polynomial time. □

Proposition 5.6. *Satisfiability problem for the Herbrand fragment is P-complete.*

Proof. The lower bound follows straight from the corresponding lower bound the Herbrand fragment without equality, so we just have to prove the upper bound. Let φ be a sentence in the Herbrand fragment. If x_i is a variable appearing in φ, then we say that it is existentially quantified, if the quantifier binding it is existential. Similarly we say that such a variable is universally quantified, if the quantifier binding it is universal. We will assume that each variable is quantified precisely ones in the sentence φ. Furthermore, for notational convenience, we assume that if φ contains an atomic formula $x_i = x_j$ (or a negation of such), then the quantifier binding x_j appears in the scope of the quantifier binding x_i.

Suppose first that φ contains a formula $\neg x_i = x_j$, so that one of the following cases holds.

1. x_i is existentially quantified and x_j is universally quantified.

2. x_i and x_j are both universally quantified.

3. x_i and x_j are the same variable.

Then clearly φ is not satisfiable and we can reject it. Suppose then that φ contains a formula $x_i = x_j$ so that either

1. x_i is existentially quantified and x_j is universally quantified, or

2. x_i and x_j are both universally quantified.

Then φ is clearly satisfiable iff it has a model of size one, which can be checked in polynomial time by lemma 5.5.

In the remaining cases we can eliminate all the equalities from φ as follows. First suppose that φ contains at least one instance of a formula $\neg x_i = x_j$ so that either

1. x_i is universally quantified and x_j is existentially quantified, or

2. x_i and x_j are both existentially quantified.

Then we will introduce a fresh binary relation symbol E, and replace all such formulas $\neg x_i = x_j$ with the formula $\neg E(x_i, x_j)$. If φ^* denotes the resulting formula, then clearly φ is equisatisfiable with $\forall x E(x, x) \wedge \varphi^*$, which can be easily transformed to a sentence in the Herbrand fragment.

Suppose then that φ contains at least one instance of a formula $x_i = x_j$ so that either

1. x_i is universally quantified and x_j is existentially quantified, or

2. x_i and x_j are both existentially quantified.

In both cases we can remove the existential quantifier which is binding the variable x_j from the sentence φ, and replace x_j with x_i in every atomic formula that appears in φ. The remaining equalities are then of the form $x_i = x_i$, and they can be discarded from the sentence.

Thus we have managed to eliminate all the equalities from φ while preserving its satisfiability. Since the resulting sentence belongs to the Herbrand fragment without equality, its satisfiability can be checked in polynomial time. $\square$

Lemma 5.7. *The satisfiability problem of $\mathcal{F}$ is in NP.*

Proof. Let $\chi \in \mathcal{F}$ be a sentence. Start by transforming χ into negation normal form, thus obtaining a sentence χ'. Now note that in $\mathcal{F}$, the formula $\forall x(\varphi \vee \psi)$ is equivalent to either $(\varphi \vee \psi)$, $(\forall x\varphi \vee \psi)$ or $(\varphi \vee \forall x\psi)$ since $\mathrm{Free}(\varphi) \cap \mathrm{Free}(\psi) = \emptyset$. Similarly, $\exists x(\varphi \wedge \psi)$ is equivalent to $(\varphi \wedge \psi)$, $(\exists x\varphi \wedge \psi)$ or $(\varphi \wedge \exists x\psi)$. Thus we can push all quantifiers past all binary connectives in the formula χ' in polynomial time, getting a formula χ''.

Consider then the following trick. Let C denote the set of all conjunctions obtained from χ'' as follows: we begin from the syntax tree of χ'' and keep eliminating disjunctions by always keeping exactly one of the two disjuncts. Clearly χ'' is satisfiable iff some $\beta \in C$ is satisfiable. Starting from χ'', we nondeterministically guess some $\beta \in C$ (without constructing C).

Now β is a conjunction of sentences $Qx_1 ... Qx_k\eta$, where $Q_i \in \{\exists, \forall\}$ and η is either an atomic formula or a negation of such. Putting β in prenex normal form, we get a sentence of Herbrand fragment; thus by proposition 5.6 it can be checked in polynomial time whether it is satisfiable. $\square$

Lemma 5.8. *Terms of* $\mathrm{GRA}(e, p, s, \neg, J, \exists)$ *translate to equisatisfiable formulas of $\mathcal{F}$ in polynomial time.*

Proof. We use induction on the structure of terms $\mathcal{T}$ of $\mathrm{GRA}(u, p, s, \neg, J, \exists)$. We translate every k-ary term to a formula $\chi(v_1, \ldots, v_k)$, so the free variables are precisely $v_1, \ldots, v_k$. For the base case we note that e is equivalent to $x_1 = x_2$ and R to $R(v_1, \ldots, v_k)$. If $\mathcal{T}$ is equivalent to $\varphi(v_1, \ldots, v_k)$, then $\neg\mathcal{T}$ is equivalent to $\neg\varphi(v_1, \ldots, v_k)$ and $\exists\mathcal{T}$ to $\exists v_k \varphi(v_1, \ldots, v_k)$. Suppose then that $\mathcal{T}$ is equivalent to $\varphi(v_1, \ldots, v_k)$. We translate $s\mathcal{T}$ to the variant of $\varphi(v_1, \ldots, v_k)$ that swaps v_1 and v_2 and $p\mathcal{T}$ to $\varphi(v_k, v_1, \ldots, v_{k-1})$. Finally, suppose that $\mathcal{T}$ translates to $\varphi(v_1, \ldots, v_k)$ and $\mathcal{P}$ to $\psi(v_1, \ldots, v_\ell)$. Now $J(\mathcal{T}, \mathcal{P})$ is translated to $\varphi(v_1, \ldots, v_k) \wedge \psi(v_{k+1}, \ldots, v_{k+\ell})$. $\square$

From lemmas 5.7 and 5.8 it follows that the satisfiability problem for $\mathrm{GRA}(e, p, s, \neg, J, \exists)$ is in NP. We will next show that already a small fragment of it is NP-hard.

Lemma 5.9. *The satisfiability problem of* $\mathrm{GRA}(\neg, J, \exists)$ *is NP-hard.*

Proof. We give a reduction from the boolean satisfiability problem. Let φ be a sentence of propositional logic and let $\{p_1, ..., p_n\}$ be the set of propositional symbols in φ. Let $\{P_1, ..., P_n\}$ be a set of unary relation symbols. We translate φ to an equisatisfiable term $\mathcal{T}$ of $\mathrm{GRA}(\neg, J, \exists)$ as follows. Every p_i translates to $\exists P_i$. If ψ is translated to $\mathcal{T}$, then $\neg\psi$ is translated to $\mathcal{T}$. If ψ is translated to $\mathcal{T}$ and χ is translated to $\mathcal{P}$, then $(\psi \wedge \chi)$ is translated to $J(\mathcal{T}, \mathcal{P})$. It is clear that the resulting term is equisatisfiable with $\mathcal{T}$. $\square$

Corollary 5.10. *The satisfiability problem of* $\mathrm{GRA}(e, p, s, \neg, J, \exists)$ *is NP-complete.*

5.2 Undecidable fragments

In this section we identify undecidable subsystems of GRA. We first quickly observe that since by theorem 3.6 GRA without e has the same expressive power as equality-free FO, we can conclude the following.

Theorem 5.11. *The satisfiability problem of* $\mathrm{GRA}(p, s, I, \neg, J, \exists)$ *is* Π^0_1*-complete.*

We then consider GRA without s. The rest of this section will be dedicated towards proving that the satisfiability problem of $\mathrm{GRA}(p, I, \neg, J, \exists)$ is Π^0_1-complete (note that this gives an alternative proof for the lower bound of Proposition 5.11). As usual, we will prove this by reducing the tiling problem to this problem.

We will start by recalling the **tiling problem** of $\mathbb{N}\times\mathbb{N}$. A **tile** is a function $t : \{R, L, T, B\} \to C$ where C is a countably infinite set, whose members are called colors. We will use t_X to denote $t(X)$. Intuitively, t_R, t_L, t_T and t_B correspond to the colors of the right, left, top and bottom edges of a tile. Now, let $\mathbb{T}$ be a finite set of tiles. A $\mathbb{T}$**-tiling** of $\mathbb{N} \times \mathbb{N}$ is a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{T}$ such that for all $i, j \in \mathbb{N}$, we have $t_R = t'_L$ when $f(i, j) = t$ and $f(i + 1, j) = t'$, and similarly, $t_T = t'_B$ when $f(i, j) = t$ and $f(i, j + 1) = t'$. Intuitively, the right color of each tile in the tiling equals the left color of its right neighbour, and analogously for top and bottom colors. The tiling problem for the grid $\mathbb{N} \times \mathbb{N}$ asks, with the input of a finite set $\mathbb{T}$ of tiles, if there exists a $\mathbb{T}$-tiling of $\mathbb{N} \times \mathbb{N}$. It is well known that this problem is Π^0_1-complete. We will show that the satisfiability problem for $\mathrm{GRA}(p, I, \neg, J, \exists)$ is undecidable by reducing the tiling problem to it.

Define the **standard grid** $\mathfrak{G}_{\mathbb{N}} := (\mathbb{N} \times \mathbb{N}, R, U)$ where $R = \{((i, j), (i + 1, j)) \mid i, j \in \mathbb{N}\}$ and $U = \{((i, j), (i, j + 1)) \mid i, j \in \mathbb{N}\}$. If $\mathfrak{G}$ is a structure of the vocabulary $\{R, U\}$ with binary relation symbols R and U, then $\mathfrak{G}$ is **grid-like** if there is a homomorphism $\tau : \mathfrak{G}_{\mathbb{N}} \to \mathfrak{G}$. Consider now the extended vocabulary $\{R, U, L, D\}$ where L and D are binary relation symbols. Define

$$\begin{aligned}
\varphi_{inverses} &:= \forall x\forall y(R(x, y) \leftrightarrow L(y, x)) \wedge \forall x\forall y(U(x, y) \leftrightarrow D(y, x)),\\
\varphi_{successor} &:= \forall x(\exists y R(x, y) \wedge \exists y U(x, y)),\\
\varphi_{cycle} &:= \forall x\forall y\forall z\forall u[(L(y, x) \wedge U(x, z) \wedge R(z, u)) \to D(u, y)].
\end{aligned}$$

Furthermore we define $\Gamma := \varphi_{inverses} \wedge \varphi_{successor} \wedge \varphi_{cycle}$. The intended model of Γ should be thought of as the standard grid $\mathfrak{G}_{\mathbb{N}}$ extended with two additional binary relations, L pointing left and D pointing down.

Lemma 5.12. *Let* $\mathfrak{G}$ *be a structure of the vocabulary* $\{R, U, L, D\}$*. Suppose* $\mathfrak{G}$ *satisfies* Γ*. Then there exists a homomorphism from* $\mathfrak{G}_{\mathbb{N}}$ *to* $\mathfrak{G} \upharpoonright \{R, U\}$*, i.e,. to the restriction of* $\mathfrak{G}$ *to the vocabulary* $\{R, U\}$*.*

Proof. As $\mathfrak{G}$ satisfies $\varphi_{inverses}$ and φ_{cycle}, it is easy to see that $\mathfrak{G}$ satisfies the sentence $\varphi_{grid\text{–}like} := \forall x\forall y\forall z\forall u[(R(x, y) \wedge U(x, z) \wedge R(z, u)) \to U(y, u)]$. Using this sentence and $\varphi_{successor}$, it is easy to inductively construct a homomorphism from $\mathfrak{G}_{\mathbb{N}}$ to $\mathfrak{G} \upharpoonright \{R, U\}$. □

The sentence $\varphi_{grid\text{–}like}$ used above reveals the *key trick* in our argument towards proving undecidability of $\mathrm{GRA}(p, I, \neg, J, \exists)$. It would have been much more natural to use $\varphi_{grid\text{–}like}$ rather than φ_{cycle}, since we could have then replace $\Gamma = \varphi_{inverses} \wedge \varphi_{successor} \wedge \varphi_{cycle}$ in the statement of Lemma 5.12 with the sentence $\varphi_{successor} \wedge \varphi_{grid\text{–}like}$, as the proof of the lemma demonstrates. But it seems that $\varphi_{grid\text{–}like}$ can't be expressed in the algebra $\mathrm{GRA}(p, I, \neg, J, \exists)$, because of the order in which the variables occur in the atoms. To solve this problem we have to use φ_{cycle} instead of $\varphi_{grid\text{–}like}$. By extending the vocabulary, we can formulate φ_{cycle} so that the variables in it occur in a cyclic order. The proof of Theorem 5.14 shows that using this cyclicity, we can express φ_{cycle} in $\mathrm{GRA}(p, I, \neg, J, \exists)$ even though it lacks the swap operator s. We note here that essentially the same trick was also employed in the recent preprint [10], where it was used to prove that the extension of one-dimensional fluted logic with p is also undecidable.

Fix a set of tiles $\mathbb{T}$. We will represent the tiles $t \in \mathbb{T}$ by unary relation symbols P_t. Let $\varphi_{\mathbb{T}}$ be the conjunction of the following four sentences:

$$\forall x \bigvee_{t \in \mathbb{T}} P_t(x)$$

$$\bigwedge_{t \neq t'} \forall x \neg (P_t(x) \wedge P_{t'}(x))$$

$$\bigwedge_{t_R \neq t'_L} \forall x \forall y \neg (P_t(x) \wedge R(x, y) \wedge P_{t'}(y))$$

$$\bigwedge_{t_T \neq t'_B} \forall x \forall y \neg (P_t(x) \wedge U(x, y) \wedge P_{t'}(y))$$

Clearly $\varphi_{\mathbb{T}}$ expresses that $\mathbb{N} \times \mathbb{N}$ is $\mathbb{T}$-tilable:

Lemma 5.13. *$\mathbb{N} \times \mathbb{N}$ is $\mathbb{T}$-tilable iff $\varphi_{\mathbb{T}} \wedge \Gamma$ is satisfiable.*

Proof. Assume that there is a model $\mathfrak{G}$ so that $\mathfrak{G} \models \varphi_{\mathbb{T}} \wedge \Gamma$. By Lemma 5.12, there exists a homomorphism $\tau : \mathfrak{G}_{\mathbb{N}} \to \mathfrak{G} \upharpoonright \{R, U\}$. We can now define a tiling T of $\mathbb{N} \times \mathbb{N}$ by setting that $T((i, j)) = t$ if $\tau((i, j)) \in P_t$. Since $\mathfrak{G} \models \varphi_{\mathbb{T}}$ and τ is homomorphism, the tiling is well-defined and correct. For the converse direction, suppose that there is a tiling T of $\mathbb{N} \times \mathbb{N}$ using $\mathbb{T}$. Now we can expand $\mathfrak{G}_{\mathbb{N}} = (\mathbb{N} \times \mathbb{N}, R, U)$ to $\mathfrak{G}'_{\mathbb{N}} = (\mathbb{N} \times \mathbb{N}, R, U, L, D, (P_t)_{t \in \mathbb{T}})$ in the obvious way. Clearly $\mathfrak{G}'_{\mathbb{N}} \models \varphi_{\mathbb{T}} \wedge \Gamma$. □

We are now ready to prove the main theorem of this section.

Theorem 5.14. *The satisfiability problem of* $\mathrm{GRA}(p, I, \neg, J, \exists)$ *is Π^0_1-complete.*

Proof. The upper bound follows from the fact that $\mathrm{GRA}(p, I, \neg, J, \exists)$ is contained in FO. For the lower bound, it suffices to show that $\varphi_{\mathbb{T}}$ and each sentence in Γ can be expressed in $\mathrm{GRA}(p, I, \neg, J, \exists)$, since on the level of sentences J correspondence to an ordinary conjunction.

Now, note that $\varphi_{inverses}$ is equivalent to the conjunction of the sentences

$$\forall x \forall y (R(x, y) \to L(y, x)) \wedge \forall x \forall y (L(y, x) \to R(x, y)),$$
$$\forall x \forall y (U(x, y) \to D(y, x)) \wedge \forall x \forall y (D(y, x) \to U(x, y)).$$

We will start by expressing the first sentence in $\mathrm{GRA}(p, I, \neg, J, \exists)$. Consider the formula $R(x, y) \to L(y, x)$. We will start with the formula $\psi := R(x, y) \to L(z, u)$ which can be expressed by the term $\mathcal{T} = \neg J(R, \neg L)$. Now, to make ψ equivalent to $R(x, y) \to L(y, x)$, we could first write $x = u \wedge y = z \wedge \psi$ and then existentially quantify u and z away. On the algebraic side, this can be achieved by transitioning from $\mathcal{T}$ first to $Ip\,(\mathcal{T})$ and after this to $IpIp\,(\mathcal{T})$. This term is then equivalent to $R(x, y) \to L(y, x)$. Thus the sentence $\forall x \forall y (R(x, y) \to L(y, x))$ is equivalent to $\neg \exists \exists \neg IpIp\mathcal{T}$.

Consider now the formula $\varphi_{cycle} = \ \forall x \forall y \forall z \forall u [(L(y, x) \wedge U(x, z) \wedge R(z, u)) \to D(u, y)]$. In the quantifier-free part, the variables occur in a cyclic fashion, but with repetitions. We will start by translating the repetition-free variant $(L(v_1, v_2) \wedge U(v_3, v_4) \wedge R(v_5, v_6)) \to D(v_7, v_8)$ by using $\neg$ and J, letting $\mathcal{T}$ be the resulting term. Now we need to modify $\mathcal{T}$ so that the repetitions are taken into account. To introduce one repetition, first use p on $\mathcal{T}$ repeatedly to bring the involved coordinates to the right end of tuples and after that use I. Note that p suffices (and s is not needed) because in φ_{cycle} the repeated variable occurrences are cyclically adjacent to each other in the variable ordering. Thus it is easy to see that we can now form a term $\mathcal{T}'$ which is equivalent to $(L(v_1, v_2) \wedge U(v_2, v_3) \wedge R(v_3, v_4)) \rightarrow D(v_4, v_1)$, and the term $\mathcal{T}'$ can easily be modified to give a term for φ_{cycle}.

From subformulas of $\varphi_{\mathbb{T}}$, we will consider the formula $\neg(P_t(x) \wedge R(x, y) \wedge P_{t'}(y))$. Here $\psi(x, y) := R(x, y) \wedge P_{t'}(y)$ is equivalent to $\mathcal{T} := IJ(R, P_{t'})$ and thus $P_t(x) \wedge \psi(x, y)$ is equivalent to $pIpJ(P_t, \mathcal{T})$. It should be now clear how the rest of $\varphi_{\mathbb{T}}$ and the other remaining formulas can be translated into terms of the algebra $\mathrm{GRA}(p, I, \neg, J, \exists)$. □

Corollary 5.15. *The satisfiability problem of* $\mathrm{GRA}(e, p, I, \neg, J, \exists)$ *is* Π^0_1*-complete.*

6 Ordered logic

In this chapter we study the so-called ordered first-order logic (or just ordered logic), which turns out to be essentially a fragment of the fluted logic. We will show first that the satisfiability problem for the ordered logic is PSPACE-complete, and thus much easier than the satisfiability problem for the fluted logic. We will then consider a natural extension of the ordered logic which extends equality-free FO^2 and show that it has NEXPTIME-complete satisfiability problem. We conclude this chapter with the observation that adding the cyclic permutation operator p to ordered logic leads to a logic for which the satisfiability problem is undecidable.

6.1 Complexity of the ordered logic

We start by giving a precise definition for the ordered logic. The definition that we give is syntactically different from the one that Herzig gave [8], but it is nevertheless equivalent with it (on the level of sentences).

Definition 6.1. Let $\overline{v}_\omega = (v_1, v_2, ...)$ and let τ be a vocabulary. For every $k \in \mathbb{N}$ we define sets $\mathrm{OL}^k[\tau]$ as follows.

1. Let $R \in \tau$ be an ℓ-ary relational symbol and consider the prefix
$$(v_1, ..., v_\ell)$$
of $\overline{v}_\omega$ containing precisely ℓ-variables. If $k \geq \ell$, then $R(v_1, ..., v_\ell) \in \mathrm{OL}^k[\tau]$.
2. Let $\ell \leq \ell' \leq k$ and suppose that $\varphi \in \mathrm{OL}^\ell[\tau]$ and $\psi \in \mathrm{OL}^{\ell'}[\tau]$. Then $\neg\varphi, (\varphi \wedge \psi) \in \mathrm{OL}^k[\tau]$.
3. If $\varphi \in \mathrm{OL}^{k+1}[\tau]$, then $\exists v_{k+1}\varphi \in \mathrm{OL}^k[\tau]$.

Finally we define $\mathrm{OL}[\tau] := \bigcup_k \mathrm{OL}^k[\tau]$.

Note that the definition implies that if $\varphi \in \mathrm{OL}$ and ψ is a subformula of φ, then the set of free variables of ψ forms a gap-free prefix of $\overline{v}_\omega$.

Example 6.2. The sentence $\exists v_1 \exists v_2 (P(v_1) \wedge R(v_1, v_2))$ belongs to the ordered logic while $\exists v_1 \exists v_2 (P(v_2) \wedge R(v_1, v_2))$ belongs to fluted logic but not to the ordered logic.

Although the syntax of OL is rather involved, it turns out to have a very clean algebraic characterization.

Theorem 6.3. OL *and* $\mathrm{GRA}(\neg, \cap, \exists)$ *are sententially equiexpressive.*

Proof. We start by showing how to translate every sentence in $\varphi \in \mathrm{OL}$ into an equivalent one which satisfies the following property. If $(\psi \circ \chi)$, where $\circ \in \{\vee, \wedge\}$, is a subformula of φ, then $\mathrm{Free}(\psi) = \mathrm{Free}(\chi)$. This can be achieved by pushing quantifiers inwards as follows. Suppose that we have a subformula of the form $Qv_i(\psi \circ \chi)$. Now we know that if $v_i \in \mathrm{Free}(\psi)$ and $v_i \in \mathrm{Free}(\chi)$, then $\mathrm{Free}(\psi) = \mathrm{Free}(\chi)$, since v_i must be the variable among the free variables of ψ and χ with the largest index. On the other hand if we have that v_i occurs as a free variable only in one of the formulas, say ψ, then $Qv_i(\psi \circ \chi)$ is equivalent to $(Qv_i\psi \circ \chi)$. Continuing this way it is clear that we achieve an equivalent sentence with the desired property.

We then translate sentences of ordered logic to algebraic terms. Suppose that $\varphi \in \mathrm{OL}$ is a sentence which satisfies the above property. Formulas which have the form $R(v_1, ..., v_k)$ are translated to R. Suppose then that we have translated ψ to $\mathcal{T}$ and χ to $\mathcal{S}$. Then we can translate $(\psi \wedge \chi)$ to $(\mathcal{T} \cap \mathcal{S})$, $\neg\psi$ to $\neg\mathcal{T}$ and $\exists v_i \psi$ to $\exists\mathcal{T}$. In the first case we used the fact that $\mathrm{Free}(\psi) = \mathrm{Free}(\chi)$ and in the last case we used the fact that v_i must be the free variable of ψ with the largest index.

The translation from algebraic terms into ordered logic is similar to the translation from $\mathrm{GRA}(\neg, \dot{\cap}, \exists)$ to fluted logic in theorem 4.13 and thus will be omitted here. □

In the rest of this section we will show that the satisfiability problem for $\mathrm{GRA}(\neg, \cap, \exists)$ is Pspace-complete. We will first show that the satisfiability problem of $\mathrm{GRA}(\neg, \cap, \exists)$ can be reduced to that of modal logic over serial frames. Since the satisfiability problem of modal logic over serial frames is in Pspace (see, e.g. [1]), we obtain the desired upper bound. Intuitively, the need for serial frames arises from the fact that although $\forall P \cap \neg\exists P$ is not satisfiable, the corresponding formula $\Box p \wedge \neg\Diamond p$ is satisfiable over general frames. For the lower bound, we will give a reduction from the problem of deciding whether a given quantified boolean sentence QBF is valid, which is a well-known Pspace-hard problem.

Before proceeding with the upper bound, we fix some notation. We will use $\mathfrak{M} = (W, S, V)$ to denote a Kripke model, where W is the set of worlds, S is the accessibility relation and V is a valuation. Given a Kripke model $\mathfrak{M}$, $w \in W$ and $\varphi \in \mathrm{ML}$ we will use $\mathfrak{M}, w \Vdash \varphi$ to denote that φ is true in the world w.

Lemma 6.4. *The satisfiability problem of* $\mathrm{GRA}(\neg, \cap, \exists)$ *is in Pspace.*

Proof. We start by describing a translation from $\mathrm{GRA}(\neg, \cap, \exists)$ into modal logic. Let τ be a vocabulary and let Φ be a set of propositional symbols, one for each $R \in \tau$. We will use the convenient but ambiguous convention that R denotes both a relation symbol and the corresponding propositional symbol. Given a term $\mathcal{T} \in \mathrm{GRA}(\neg, \cap, \exists)[\tau]$, we define recursively its translation $t(\mathcal{T})$ as follows.

1. If $\mathcal{T} = R \in \tau$, then $t(R) = R \in \Phi$.
2. If $\mathcal{T} = \neg\mathcal{P}$, then $t(\mathcal{T}) = \neg t(\mathcal{P})$.
3. If $\mathcal{T} = (\mathcal{P} \cap \mathcal{S})$, then $t(\mathcal{T}) = (t(\mathcal{P}) \wedge t(\mathcal{S}))$.
4. If $\mathcal{T} = \exists\mathcal{P}$, then $t(\mathcal{T}) = \Diamond t(\mathcal{P})$.

We now claim that for every $\mathcal{T} \in \mathrm{GRA}(\neg, \cap, \exists)[\tau]$ we have that it is satisfiable iff $t(\mathcal{T})$ is satisfiable over the class of serial Kripke frames.

Suppose first that $\mathfrak{A}$ is a model of vocabulary τ. We then define a Kripke model $\mathfrak{M} = (W, S, V)$ by setting that

1. $W = A^{<\omega}$, i.e. the set of all finite tuples over A (including the 0-ary tuple ϵ).
2. $S = \{(\overline{a}, \overline{a}b) \mid \overline{a}b \in W\}$.
3. For every $R \in \Phi$, we define $V(R) = R^{\mathfrak{A}}$.

Clearly S is serial. We now claim that for every $\mathcal{T} \in \mathrm{GRA}(\neg, \cap, \exists)[\tau]$ and for every $\overline{a} \in A^{ar(\mathcal{T})}$, we have that

$$\overline{a} \in [\![\mathcal{T}]\!]_{\mathfrak{A}} \iff \mathfrak{M}, \overline{a} \Vdash t(\mathcal{T}).$$

In particular, this claim implies that if $\mathcal{T}$ is satisfiable, then so is $t(\mathcal{T})$.

The claim can be proved by a simple induction on $\mathcal{T}$. Here we will only consider the case of $\mathcal{T} = \exists\mathcal{P}$. Suppose first that $\overline{a} \in [\![\mathcal{T}]\!]_{\mathfrak{A}}$. Thus there exists $b \in A$ so that $\overline{a}b \in [\![\mathcal{P}]\!]_{\mathfrak{A}}$. Using the induction hypothesis, we see that $\mathfrak{M}, \overline{a}b \Vdash t(\mathcal{P})$. By definition, we have that $(\overline{a}, \overline{a}b) \in S$, which implies that $\mathfrak{M}, \overline{a} \Vdash \Diamond t(\mathcal{P})$. The other direction is similar.

For the converse direction suppose that $\mathfrak{M} = (W, S, V)$ is a Kripke model where S is serial. Define $\mathfrak{F}$ as the set

$$\{f \mid f \text{ is a function } W \to W \text{ and } f \subseteq S\}.$$

Now fix some $w_0 \in W$. We can construct a model $\mathfrak{A} = (\mathfrak{F}, (R^{\mathfrak{A}})_{R\in\tau})$ by defining that for every $R \in \tau$

$$R^{\mathfrak{A}} = \{(f_1, ..., f_n) \mid (f_n \circ ... \circ f_1)(w_0) \in V(R)\}.$$

Now we claim that for every $\mathcal{T} \in \mathrm{GRA}(\neg, \cap, \exists)[\tau]$ and for every $(f_1, ..., f_{ar(\mathcal{T})}) \in A^{ar(\mathcal{T})}$, we have that

$$(f_1, ..., f_{ar(\mathcal{T})}) \in [\![\mathcal{T}]\!]_{\mathfrak{A}} \iff \mathfrak{M}, (f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0) \Vdash t(\mathcal{T}).$$

Here we agree that $(\epsilon)(w_0) = w_0$. From this claim it would now follow in particular that if $\mathfrak{M}, w_0 \Vdash t(\mathcal{T})$, then $\epsilon \in [\![\mathcal{T}]\!]_{\mathfrak{A}}$, i.e. if $t(\mathcal{T})$ is satisfiable then so is $\mathcal{T}$.

The claim can be proved again by using a simple induction on $\mathcal{T}$. Again the only interesting case is the case of $\mathcal{T} = \exists\mathcal{P}$. Suppose first that $(f_1, ..., f_{ar(\mathcal{T})}) \in [\![\mathcal{T}]\!]_{\mathfrak{A}}$. Thus there exists $g \in A$ so that $(f_1, ..., f_{ar(\mathcal{T})}, g) \in [\![\mathcal{P}]\!]_{\mathfrak{A}}$. By induction we know that $\mathfrak{M}, (g \circ f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0) \Vdash t(\mathcal{P})$. Since $g \subseteq S$, we can conclude that $\mathfrak{M}, (f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0) \Vdash \Diamond t(\mathcal{P})$. Conversely, suppose that $\mathfrak{M}, (f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0) \Vdash \Diamond t(\mathcal{P})$. Thus there exists a $w \in W$ so that $((f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0), w) \in S$ and $\mathfrak{M}, w \Vdash t(\mathcal{P})$. Since S is serial, there clearly exists a function $g \subseteq S$ so that $g((f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0)) = w$. Thus $\mathfrak{M}, (g \circ f_{ar(\mathcal{T})} \circ ... \circ f_1)(w_0) \Vdash t(\mathcal{P})$, and hence by induction $(f_1, ..., f_{ar(\mathcal{T})}, g) \in [\![\mathcal{P}]\!]_{\mathfrak{A}}$, which implies that $(f_1, ..., f_{ar(\mathcal{T})}) \in [\![\exists\mathcal{T}]\!]_{\mathfrak{A}}$.

□

We will next prove the PSPACE-hardness via a reduction from the problem of determining whether a given QBF is valid. Our proof is heavily based on the standard proof of the PSPACE-hardness of modal logic, which can be found for instance in [1]. We note that one could also use the techniques of lemmas 6.4 and 6.5 to prove the PSPACE-hardness of $\mathrm{GRA}(\neg, \cap, \exists)$ via a reduction from the satisfiability problem of modal logic over serial frames.

Lemma 6.5. *The satisfiability problem of* $\mathrm{GRA}(\neg, \cap, \exists)$ *is* PSPACE*-hard.*

Proof. Fix a QBF sentence $\varphi := Q_1x_1...Q_nx_n\psi$, where ψ is built using $\neg$ and $\wedge$ and treat other connectives as abbreviations in the usual way. Let τ denote the vocabulary $\{R_{i,j} \mid 1 \leq j \leq i \leq n\}$, where each relation $R_{i,j}$ has arity i. Our goal is to write down a term $\mathcal{T} \in \mathrm{GRA}(\neg, \cap, \exists)[\tau]$ which is satisfiable iff φ is valid.

Before moving forward, we will fix some notation. If $Q_{i+1}y_{i+1}...Q_ny_n\chi$ is a QBF formula and $v : \{y_1, ..., y_i\} \to \{0, 1\}$ is a mapping, where $y_1, ..., y_i$ is the set of free variables of the QBF formula, then we use $v \models Q_{i+1}y_{i+1}...Q_ny_n\chi$ to denote the fact that v satisfies the formula $Q_{i+1}y_{i+1}...Q_ny_n\chi$. Furthermore, if v is such a valuation, $c \in \{0, 1\}$ and z is a variable, then we use $v[c/z]$ to denote the valuation which maps z to c and otherwise agrees with v.

Now we can start defining the term $\mathcal{T}$. The idea is that we can associate each tuple of elements $(a_1, ..., a_i)$ to a valuation $v : \{x_1, ..., x_i\} \to \{0, 1\}$ by using the relations $R_{i,j}$ to describe the values received by the variables. Intuitively the tuple $(a_1, ..., a_i)$ should belong to $R_{i,j}$ if and only if the valuation corresponding to $(a_1, ..., a_i)$ maps x_j to 1.

If v is the valuation corresponding to a tuple $\overline{a} \in A^i$ and v' is the valuation corresponding to some extension $\overline{ab}$ of $\overline{a}$, then for every $1 \leq j \leq i$ we should have that $v(x_j) = v'(x_j)$. We use the following term to enforce this.

$$\mathcal{T}_1 := \bigcap_{1 \leq i < n} \forall^i \bigcap_{1 \leq j \leq i} ((\neg R_{i,j} \cup \forall R_{i+1,j}) \cap (R_{i,j} \cup \forall \neg R_{i+1,j}))$$

Here $\forall^i$ denotes a sequence of $\forall$ of length i. A second term is needed to make sure that if x_i is universally quantified, then every tuple $\overline{a} \in A^{i-1}$ has two extensions $\overline{a}b$ and $\overline{a}b'$ so that if v is the valuation corresponding to $\overline{a}$, then $\overline{a}b$ and $\overline{a}b'$ correspond to the valuations $v[1/x_i]$ and $v[0/x_i]$ respectively.

$$\mathcal{T}_2 := \bigcap_{\substack{1 \leq i \leq n, \\ x_i \text{ is universally} \\ \text{quantified in } \varphi}} \forall^{i-1} (\exists R_{i,i} \cap \exists \neg R_{i,i})$$

Finally we have to enforce the fact that after all the variables have received an interpretation, the resulting sentence of propositional logic should evaluate to true. To do this, we define the term $\mathcal{T}(\psi)$ as the term obtained from ψ by replacing each variable x_i with the relation symbol $R_{n,i}$ and each conjunction with $\cap$. Then the following term will suffice.

$$\mathcal{T}_3 := \forall^n \mathcal{T}(\psi)$$

We then define that $\mathcal{T} := \mathcal{T}_1 \cap \mathcal{T}_2 \cap \mathcal{T}_3$. We will now show that $\mathcal{T}$ is satisfiable if and only if φ is valid. Suppose first that there exists a model $\mathfrak{A}$ so that $[\![\mathcal{T}]\!]_{\mathfrak{A}}$ is non-empty. To every $1 \leq i \leq n$ and $\overline{a} \in A^i$ we will associate a valuation $v(\overline{a}) : \{x_1, ..., x_i\} \to \{0, 1\}$, which is defined by setting that $v(\overline{a})(x_j) = 1$ iff $(a_1, ..., a_i) \in R^{\mathfrak{A}}_{i,j}$. We will prove using induction that for every $1 \leq i \leq n$ and $\overline{a} \in A^{i-1}$ we have that $v(\overline{a}) \models Q_i x_i ... Q_n x_n \psi$. For the base case we note that $v(\overline{a}) \models \psi$, for every $\overline{a} \in A^n$, since $\mathfrak{A} \models \mathcal{T}_3$. Suppose then that $v(\overline{a}) \models Q_{i+1} x_{i+1} ... Q_n x_n \psi$ for every $\overline{a} \in A^i$. Let $\overline{a} \in A^{i-1}$. We have now two cases based on whether x_i is existentially or universally quantified. If x_i is existentially quantified, then the claim follows trivially from the induction hypothesis. To establish the claim in the case that x_i is universally quantified, we use the fact that $\mathfrak{A} \models \mathcal{T}_2$, from which we can conclude that there exists $b, b' \in A$ so that $\overline{a}b \in R^{\mathfrak{A}}_{i,i}$ and $\overline{a}b' \notin R^{\mathfrak{A}}_{i,i}$. By assumption we know then that $v(\overline{a}b) \models Q_{i+1} x_{i+1} ... Q_n x_n \psi$ and $v(\overline{a}b') \models Q_{i+1} x_{i+1} ... Q_n x_n \psi$. Since $\mathfrak{A} \models \mathcal{T}_1$, we have that $v(\overline{a}b) = v(\overline{a})[1/x_i]$ and $v(\overline{a}b') = v(\overline{a})[0/x_i]$, and hence we can conclude that $v(\overline{a}) \models Q_i x_i ... Q_n x_n \psi$.

Suppose then that φ is valid. We will construct a model $\mathfrak{A}$ with domain $\{0, 1\}$ by specifying the interpretations for the relation symbols $R_{1,1}, ..., R_{n,n}$. This suffices, since for every $i < j \leq n$ the interpretation for $R_{j,i}$ is determined by the interpretation $R_{i,i}$, which follows from the fact that the model $\mathfrak{A}$ should satisfy $\mathcal{T}_1$. The interpretations will be defined inductively while maintaining the following condition: if the interpretations have been fixed for the relations $R_{1,1}, ..., R_{i,i}$, then for every $\overline{a} \in \{0, 1\}^i$ we have that $v(\overline{a}) \models Q_{i+1} x_{i+1} ... Q_n x_n \psi$.

So, suppose that the interpretations have been fixed for the relations $R_{1,1}, ..., R_{i,i}$, where $i < n$, and we want to define the interpretation for the relation $R_{i+1,i+1}$. Let $\overline{a} \in \{0, 1\}^i$. We have now two cases based on whether x_{i+1} is existentially or universally quantified. If x_{i+1} is existentially quantified, then there exists $b \in \{0, 1\}$ so that $v(\overline{a})[b/x_{i+1}] \models Q_{i+2} x_{i+2} ... Q_n x_n \psi$. We now add the tuples $(\overline{a}, 0)$ and $(\overline{a}, 1)$ to the interpretation of $R_{i+1,i+1}$ iff $b = 1$. Suppose then that x_{i+1} is universally quantified. Then we add only $(\overline{a}, 1)$ to the interpretation of $R_{i+1,i+1}$. Since $\overline{a} \in \{0, 1\}^i$ was arbitrary, this takes care of defining the interpretation for $R_{i+1,i+1}$. Clearly we have managed to maintain the above condition.

The above strategy for defining the interpretations for the relations is clearly consistent with the requirement that the resulting model should satisfy $\mathcal{T}_2$. Furthermore, we know by construction that for every $\overline{a} \in \{0,1\}^n$ we have that $v(\overline{a}) \models \psi$, which in turn implies that $\mathfrak{A}$ does satisfy the term $\mathcal{T}_3$. Thus $\mathfrak{A}$ is a model of $\mathcal{T}$. □

Corollary 6.6. *The satisfiability problem of* GRA$(\neg, \cap, \exists)$ *is* P*SPACE-complete.*

6.2 Decidable extension of the ordered logic

In this section we will consider the algebra GRA$(s, I, \neg, C, \cap, \exists)$, which is a natural extension of the ordered logic. To motivate this algebra, we note that theorem 4.7 implies that over vocabularies with at most binary relation symbols, the algebra is sententially equivalent with the equality-free fragment of FO^2. Thus one can consider GRA$(s, I, \neg, C, \cap, \exists)$ to be a generalization of equality-free FO^2 into contexts with arbitrary relational vocabularies.

The purpose of this section is to show that GRA$(s, I, \neg, \cap, \exists)$ is decidable by showing that it has the exponential model property. The proof will be a rather straightforward extension of the proof for the exponential model property of FO^2, although our situation is somewhat simpler since we do not have to deal with kings (due to the fact that our algebra is clearly contained in the equality-free fragment of FO). For more details on the original proof of the exponential model property of FO^2, see the paper [6].

We start by defining the concept of a type and stating some auxiliary facts about them.

Definition 6.7. Let τ be a vocabulary. An *atomic k-ary term* is a k-ary term from GRA(s, I). A *k-type* is a maximally consistent set of k-ary terms from GRA$(s, I, \neg)[\tau]$.

Remark. Note that according to the above definition a k-type is an infinite set. However, it is easy to see that we can associate with each k-type a equivalent k-ary term.

Definition 6.8. Let τ be a vocabulary and let $\mathfrak{A}$ be a model of vocabulary τ. Let $\overline{a} \in A^k$ be a k-tuple. The *type of $\overline{a}$* is defined as the set

$$tp_{\mathfrak{A}}(\overline{a}) := \{\alpha \mid \alpha \text{ is an atomic } k\text{-term}, \overline{a} \in [\![\alpha]\!]_{\mathfrak{A}}\} \cup \{\neg\alpha \mid \alpha \text{ is an atomic } k\text{-term}, \overline{a} \notin [\![\alpha]\!]_{\mathfrak{A}}\}$$

We also say that $\overline{a}$ *realizes* the type $tp_{\mathfrak{A}}(\overline{a})$.

Lemma 6.9. *Let τ be a vocabulary and let $\mathfrak{A}$ and $\mathfrak{B}$ be models of vocabulary τ. Let $\overline{a} \in A^k$ and $\overline{b} \in B^k$ be k-tuples so that $tp_{\mathfrak{A}}(\overline{a}) = tp_{\mathfrak{B}}(\overline{b})$. Now the following conditions hold.*

1. $tp_{\mathfrak{A}}(a_1, ..., a_k, a_{k-1}) = tp_{\mathfrak{B}}(b_1, ..., b_k, b_{k-1})$.
2. $tp_{\mathfrak{A}}(a_1, ..., a_k, a_k) = tp_{\mathfrak{B}}(b_1, ..., b_k, b_k)$.

Proof. Straightforward. □

Definition 6.10. Let τ be a vocabulary and let $\mathfrak{A}$ and $\mathfrak{B}$ be models of vocabulary τ. Let $\overline{a} \in A^k$ and $\overline{b} \in B^k$ be k-tuples. We say that $\overline{a}$ and $\overline{b}$ are *similar*, if the following conditions hold.

1. $tp_{\mathfrak{A}}(\overline{a}) = tp_{\mathfrak{B}}(\overline{b})$.
2. $tp_{\mathfrak{A}}(a_{k-1}) = tp_{\mathfrak{B}}(b_{k-1})$.
3. $tp_{\mathfrak{A}}(a_k) = tp_{\mathfrak{B}}(b_k)$.

Lemma 6.11. *Let τ be a vocabulary and let $\mathfrak{A}$ and $\mathfrak{B}$ be models of vocabulary τ. Let $\mathcal{T} \in \mathrm{GRA}(s, I, \neg, C, \cap)$ be a k-ary term. Now, for every similar k-tuples $\overline{a} \in A^k$ and $\overline{b} \in B^k$ we have that*

$$\overline{a} \in [\![\mathcal{T}]\!]_{\mathfrak{A}} \iff \overline{b} \in [\![\mathcal{T}]\!]_{\mathfrak{A}}.$$

Proof. Induction using lemma 6.9. □

Next we will introduce a Scott-type normal form for $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$. Given a term $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$ term $\mathcal{T}$, we say that it is in **normal form** if it has the following shape

$$\bigcap_{i \in I} \forall^{n_i}(\neg\alpha_i \cup \exists\beta_i) \cap \bigcap_{j \in J} \forall^{n_j}(\neg\alpha_j \cup \forall\beta_j),$$

where I, J are some sets of indices, α_i and α_j represent n_i-ary and n_j-ary terms from the algebra $\mathrm{GRA}(s, I, \neg, C, \cap)$, and β_i, β_j represent $(n_i + 1)$-ary and $(n_j + 1)$-ary terms from $\mathrm{GRA}(s, I, \neg, C, \cap)$. Here $\forall^m$ denotes a sequence of $\forall$ of length m. We will also allow n_i and n_j to be zero, in which case $\forall^{n_i}(\neg\alpha_i \cup \exists\beta_i)$ and $\forall^{n_j}(\neg\alpha_j \cup \forall\beta_j)$ represent $\exists\beta_i$ and $\forall\beta_j$.

In a rather standard way one can prove the following lemma.

Lemma 6.12. *There exists a nondeterministic polynomial time procedure which translates each 0-ary $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$ term $\mathcal{T}$ to a $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$ 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 $\mathfrak{A}'$ of $\mathfrak{A}$ so that $\mathfrak{A}' \models \mathcal{T}'$, and vice versa, if $\mathfrak{A} \models \mathcal{T}'$, then $\mathfrak{A} \models \mathcal{T}$.*

Proof. If $\mathcal{T}$ is a 0-ary term of the form $\exists\mathcal{P}$ or $\neg\exists\mathcal{P}$, where $\mathcal{P}$ is quantifier-free, then it already is in normal form. Otherwise $\mathcal{T}$ contains a proper subterm $\exists\mathcal{P}$, where $\mathcal{P}$ is a quantifier-free term. If $\mathcal{P}$ is unary, then we will guess a truth value, and replace $\exists\mathcal{P}$ with either $\bot$ or $\top$ according to this guess. If the resulting term is $\mathcal{T}'$, then $\mathcal{T}$ is satisfiable over the same domain as $\mathcal{T}' \cap \exists\mathcal{P}$ or $\mathcal{T}' \cap \neg\exists\mathcal{P}$, depending on what was the guessed truth value.

In the case where the arity of $\mathcal{P}$ is at least two, we will introduce a fresh relation symbol R of same arity as $\exists\mathcal{P}$, and replace the latter with the former in $\mathcal{T}$. If the resulting term is $\mathcal{T}'$, then it is easy to verify that $\mathcal{T}$ is satisfiable over the same domain as

$$\mathcal{T}' \cap \forall^{ar(R)}(\neg R \cup \exists\mathcal{P}) \cap \forall^{ar(R)}(R \cup \forall\neg\mathcal{P}).$$

In both cases by repeating the above process on the term $\mathcal{T}'$, we will eventually end up with an equi-satisfiable term which is in normal form. □

Let $\mathcal{T}$ be a $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$-term in normal form. Conjuncts $\forall^{n_i}(\neg\alpha_i \cup \exists\beta_i)$ of $\mathcal{T}$ are called **existential requirements** and denoted by $\mathcal{T}_i^{\exists}$. Let $\mathfrak{A}$ be a model of $\mathcal{T}$ and let $\overline{a} \in A^{n_i}$ so that $\overline{a} \in [\![\alpha_i]\!]_{\mathfrak{A}}$. Then the element $b \in A$ so that $\overline{a}b \in [\![\beta_i]\!]_{\mathfrak{A}}$ is called a **witness** for $\overline{a}$ and $\mathcal{T}_i^{\exists}$.

Theorem 6.13. *Every satisfiable term $\mathcal{T}$ of $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$ has a model of size bounded exponentially in $|\mathcal{T}|$.*

Proof. Using lemma 6.12 we can assume that $\mathcal{T}$ is in normal form. Let τ be the set of relation symbols occuring in $\mathcal{T}$ and let $\mathfrak{A}$ be a model of $\mathcal{T}$ of vocabulary τ. We will now construct a bounded model $\mathfrak{B} \models \mathcal{T}$.

As the domain of the model $\mathfrak{B}$ we will take the set

$$B := \{tp_{\mathfrak{A}}(a) \mid a \in A\} \times \{1, ..., m_{\exists}\} \times \{0, 1, 2\},$$

where $m_{\exists}$ is the number of existential requirements in $\mathcal{T}$. Now, to construct the model $\mathfrak{B}$, we have to assign a type to every tuple of elements from B so that the following conditions hold.

i) For every tuple $\overline{b}$ of elements from B, there exists a tuple $\overline{a}$ of elements from A so that $\overline{b}$ and $\overline{a}$ are similar.

ii) There exists a witness for every existential requirement $\mathcal{T}_i^{\exists}$ and for every tuple $\overline{b} \in [\![\alpha_i]\!]_{\mathfrak{A}}$

If we construct $\mathfrak{B}$ this way, then by $i)$ it satisfies all the conjuncts $\forall^{n_j}(\neg\alpha_j \cup \forall\beta_j)$ and by $ii)$ it satisfies all the existential requirements of $\mathcal{T}$, and thus would be a model of $\mathcal{T}$.

We will assign the types inductively by starting with the 1-types: for every $b = (tp_{\mathfrak{A}}(a), i, j)$ in B we define that $tp_{\mathfrak{B}}(b) := tp_{\mathfrak{A}}(a)$. Suppose then that we have defined the types for k-tuples in a manner that is consistent with the requirements $i)$ and $ii)$. We start defining the types for $(k+1)$-tuples by providing witnesses for all the relevant tuples. So, consider an existential requirement $\mathcal{T}_i^{\exists}$ and a tuple $\overline{b} \in B^k$ so that $\overline{b} \in [\![\alpha_i]\!]_{\mathfrak{B}}$. Suppose that $b_k = (tp_{\mathfrak{A}}(a), i', j)$. By $i)$ there exists a tuple $\overline{a} \in A^k$ so that $\overline{b}$ and $\overline{a}$ are similar. Thus - by lemma 6.11 - $\overline{a} \in [\![\alpha_i]\!]_{\mathfrak{A}}$. Since $\mathfrak{A} \models \mathcal{T}_i^{\exists}$, there exists an element $c \in A$ which is a witness for $\overline{a}$ and $\mathcal{T}_i^{\exists}$. If $c = a_k$, then we will use b_k as a witness for $\overline{b}$ by defining that $tp_{\mathfrak{B}}(\overline{b}b_k) := tp_{\mathfrak{A}}(\overline{a}a_k)$ (note that this is consistent with the type of $tp_{\mathfrak{B}}(\overline{b})$). On the other hand if $c \neq a_k$, then we will use the element $d = (tp_{\mathfrak{A}}(c), i, j+1 \mod 3) \in B$ as a witness for $\overline{b}$ by defining that $tp_{\mathfrak{B}}(\overline{b}d) := tp_{\mathfrak{A}}(\overline{a}c)$ and $tp_{\mathfrak{B}}((b_1, ..., b_{k-1}, d, b_k)) := tp_{\mathfrak{A}}((a_1, ..., a_{k-1}, c, a_k))$.

Before moving forward, let us argue that our method of assigning witnesses does not produce conflicts, where by conflict we mean a situation where we are assigning two different types for the same tuple. So consider a tuple $\overline{b} = (b_1, ..., b_k) \in B^k$ and $d \in B$ so that we used d as a witness for $\overline{b}$ and some existential requirement $\mathcal{T}_i^{\exists}$. We will argue that the type for the tuple $\overline{b}d$ is specified only once. If $b_k = d$, then the type of $\overline{b}d$ was determined by the type of $\overline{b}$, i.e. it is specified only once. In the case where $b_k \neq d$, we note that we have reserved 1-types for each of the existential requirements, and thus we used d as a witness for $\overline{b}$ only for the existential requirement $\mathcal{T}_i^{\exists}$. We then note that since we are assigning witnesses for tuples in a "cyclic" manner, we will not use b_k as a witness for the tuple $(b_1, ..., b_{k-1}, d)$. Since these two cases are the only possible ways that we might have specified again the type of the tuple $\overline{b}d$, we conclude that it is only specified once.

We will now assign types for the remaining $(k+1)$-tuples. So, consider a tuple $\overline{b} \in B^k$ and $d = (tp_{\mathfrak{A}}(c), i, j) \in B$ so that we have not defined the type for the tuple $\overline{b}d$. If $d = b_k$, then the type of $tp_{\mathfrak{B}}(\overline{b}b_k)$ is determined by the type of $tp_{\mathfrak{B}}(\overline{b})$. On the other hand, if $d \neq b_k$, then by $i)$ there exists a tuple $\overline{a} \in A^k$ which is similar to $\overline{b}$. Now we define that $tp_{\mathfrak{B}}(\overline{b}d) := tp_{\mathfrak{A}}(\overline{a}c)$ and that $tp_{\mathfrak{B}}((b_1, ..., b_{k-1}, d, b_k)) = tp_{\mathfrak{A}}((a_1, ..., a_{k-1}, c, a_k))$.

Having now assigned types for all the tuples, we see that $\mathfrak{B}$ satisfies the conditions $i)$ and $ii)$, and thus is a model of $\mathcal{T}$. Since

$$|\{tp_{\mathfrak{A}}(a) \mid a \in A\}| \leq 2^{|\tau|} \leq 2^{|\mathcal{T}|}$$

we have that $B \leq 3m_{\exists}2^{|\mathcal{T}|}$, which is exponential in $|\mathcal{T}|$. Thus $\mathfrak{B}$ is the desired model. □

Our next goal is to prove a matching lower bound for the satisfiability problem of GRA$(s, I, \neg, C, \cap, \exists)$, and in fact we will prove that the satisfiability problem is NexpTime-hard already for GRA$(s, \neg, \cap, \exists)$. To prove this, we will give a reduction from the satisfiability problem of two-dimensional S5, which we will denote by S5^2. The satisfiability problem for this logic is known to be NexpTime-hard [19].

The models of S5^2 are Kripke-structures $(W, \equiv_1, \equiv_2, V)$, where $W = U \times U$ for some set U, V is a valuation and for $i \in \{1, 2\}$, $\equiv_i$ is the accessibility relation

$$\{(w, v) \in W^2 \mid w_i = v_i\},$$

where w_i refers to the i:th coordinate of w. The language of $S5^2$ is the same as standard modal logic with the exception that instead of a single diamond, we now have two diamonds $\Diamond_1$ and $\Diamond_2$ corresponding to the two accessibility relations.

Lemma 6.14. *The satisfiability problem for* $\mathrm{GRA}(s, \neg, \cap, \exists)$ *is* NExpTime*-hard. This holds already over binary vocabularies.*

Proof. As mentioned above, we will give a reduction from the satisfiability problem of $S5^2$. Let $\varphi \in S5^2$ be a fixed formula. Our goal is to construct a term $\mathcal{T} \in \mathrm{GRA}(s, \neg, \cap, \exists)$ which is satisfiable if and only if φ is, and we will begin by fixing the set of relation symbols occurring in the term. First, for every propositional symbol p occurring in φ, we will add binary relation symbol P. Secondly, for each subformula φ of the form $\Diamond_i\psi$, we will add a binary relation symbol $S_{\Diamond_i\psi}$. Let τ denote the resulting vocabulary.

Now we are ready to define recursively a mapping

$$t : \mathrm{Subf}(\varphi) \to \mathrm{GRA}(s, \neg, \cap, \exists)[\tau],$$

where $\mathrm{Subf}(\varphi)$ denotes the set of subformulas of φ, as follows.

1. $t(p) = P$, where p is a propositional symbol.
2. $t(\neg\psi) = \neg t(\psi)$.
3. $t(\psi \wedge \chi) = t(\psi) \cap t(\chi)$.
4. $t(\Diamond_i\psi) = S_{\Diamond_i\psi}$.

Then we define

$$\mathcal{T} := \exists\exists t(\varphi) \cap \bigcap_{\Diamond_1\psi \in \mathrm{Subf}(\varphi)} \forall((\neg\exists S_{\Diamond_1\psi} \cup \exists t(\psi)) \cap (\neg\exists t(\psi) \cup \forall S_{\Diamond_1\psi}))$$
$$\cap \bigcap_{\Diamond_2\psi \in \mathrm{Subf}(\varphi)} \forall((\neg\exists s S_{\Diamond_2\psi} \cup \exists s t(\psi)) \cap (\neg\exists s t(\psi) \cup \forall s S_{\Diamond_2\psi})).$$

We claim that φ is satisfiable if and only if $\mathcal{T}$ is satisfiable. We will sketch a proof of the direction from right to left. So, suppose that $\mathfrak{A}$ is a model of $\mathcal{T}$. The model $\mathfrak{A}$ translates quite directly to a model $\mathfrak{M}$ of $S5^2$, by defining $W = A \times A$. Using an induction on the set of subformulas ψ of φ, one can show that for every $(a, b) \in W$ we have that $\mathfrak{M}, (a, b) \Vdash \psi$ if and only if $(a, b) \in [\![t(\psi)]\!]_{\mathfrak{A}}$. Since there exists $(a, b) \in W$ so that $(a, b) \in [\![t(\varphi)]\!]_{\mathfrak{A}}$, it follows from this that φ is satisfiable.

The only interesting case in the induction is the case of $\Diamond_i\psi$. For concreteness, let us fix $i = 1$. Now, suppose that $\mathfrak{M}, (a, b) \Vdash \Diamond_1\psi$. Thus there exists $c \in A$ so that $\mathfrak{M}, (a, c) \Vdash \psi$. Using induction hypothesis, we see that $(a, c) \in [\![t(\psi)]\!]_{\mathfrak{A}}$. Thus, for every $c' \in A$, we have that $(a, c') \in [\![S_{\Diamond_1\psi}]\!]_{\mathfrak{A}}$. In particular, we have that $(a, b) \in [\![S_{\Diamond_1\psi}]\!]_{\mathfrak{A}}$, i.e. $(a, b) \in [\![t(\Diamond_1\psi)]\!]_{\mathfrak{A}}$. For the converse direction, suppose that $(a, b) \in [\![t(\Diamond_1\psi)]\!]_{\mathfrak{A}} = [\![S_{\Diamond_1\psi}]\!]_{\mathfrak{A}}$. Thus there exists $c \in A$ so that $(a, c) \in t(\psi)$. Using induction hypothesis, we see that $\mathfrak{M}, (a, c) \Vdash \psi$, and hence $\mathfrak{M}, (a, b) \Vdash \Diamond_1\psi$. □

Theorem 6.15. *The satisfiability problem for* $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$ *is* NExpTime*-complete.*

Proof. The lower bound follows from the lemma 6.14. For the upper bound we describe a non-deterministic procedure running in exponential time. When given a term $\mathcal{T}$ of the algebra $\mathrm{GRA}(s, I, \neg, C, \cap, \exists)$, the procedure will first convert it into a term $\mathcal{T}'$ in normal form. Then the procedure will guess an exponential model $\mathfrak{B}$ for $\mathcal{T}'$ (note that also the description of $\mathfrak{B}$ is bounded exponentially in $|\mathcal{T}|$, since we are dealing with at most $|\mathcal{T}|$ many relations of arity at most $|\mathcal{T}|$) and verify that it is a model of $\mathcal{T}'$. This latter task can be simply performed in an exhaustive way. □

6.3 Undecidable extension of ordered logic

In this section we will observe that adding the cyclic permutation operator p to ordered logic leads to a undecidable logic. It turns out that this logic is quite expressive, and thus it is relatively easy to show that its satisfiability problem is undecidable. One option is to reduce the tiling problem to its satisfiability problem via a similar method as the one used in the proof of theorem 5.14. However, here we will reduce the satisfiability problem of $\mathrm{S5}^3$ to the satisfiability problem of $\mathrm{GRA}(p, \neg, \cap, \exists)$.

As the models of $\mathrm{S5}^2$ were "grids", the models of $\mathrm{S5}^3$ will be "cubes". More precisely, the models of $\mathrm{S5}^3$ are Kripke-structures $(W, \equiv_1, \equiv_2, \equiv_3, V)$, where $W = U \times U \times U$ for some set U, V is a valuation and for $i \in \{1, 2, 3\}$, $\equiv_i$ is the accessibility relation

$$\{(w, v) \in W^2 \mid w_j = v_j, \text{ for every } j \neq i.\}$$

The language of $\mathrm{S5}^3$ is similar to the language of $\mathrm{S5}^2$, with the exception that we have now three diamonds $\Diamond_1, \Diamond_2$ and $\Diamond_3$, which correspond to the three accessibility relations. It is well-known that the satisfiability problem for $\mathrm{S5}^3$ is undecidable[1].

The satisfiability problem of $\mathrm{S5}^3$ can be reduced to the satisfiability problem of the algebra $\mathrm{GRA}(p, \neg, \cap, \exists)$ in a manner that is very similar to the proof of lemma 6.14. The main difference is that the vocabulary will now consists of ternary relation symbols.

Theorem 6.16. *The satisfiability problem of* $\mathrm{GRA}(p, \neg, \cap, \exists)$ *is* Π^0_1*-complete. This holds already over ternary relations.*

It is interesting to note that there is a natural way of turning the logic $\mathrm{S5}^3$ decidable.[2] The idea is to replace the relations $\equiv_i$ with the relations $\sim_i$ which are defined as

$$\{(w, v) \in W \mid w_i = v_i\}.$$

In fact, if the accessibility relations are defined this way, then the logic $\mathrm{S5}^n$ becomes decidable, for every $n \in \omega$.

Now, let us take an another look at the "standard translation" that we gave in the proof of lemma 6.14. How would the translation look like if we were to translate $\mathrm{S5}^3$ with the semantics that we just introduced? One part of the resulting sentence would have the form

$$\bigcap_{\Diamond_1\psi \in \mathrm{Subf}(\varphi)} \forall((\neg\forall\forall S_{\Diamond_1\psi} \cup \exists\exists t(\psi)) \cap (\neg\exists\exists t(\psi) \cup \forall\forall S_{\Diamond_1\psi})),$$

and here we note that this term is *one-dimensional*, in the sense that whenever we are applying projection, we apply it enough times so that the resulting term is an unary term. A quick verification shows that even if we would translate $\mathrm{S5}^n$ to $\mathrm{GRA}(p, \neg, \cap, \exists)$, the resulting terms would still be one-dimensional. All of the terms of $\mathrm{GRA}(p, \neg, \cap, \exists)$ are also *uniform* in the sense that we can only express conjunctions of formulas if the formulas have the same set of free variables. Thus we have managed to translate $\mathrm{S5}^n$ into the so-called one-dimensional uniform fragment of FO, which was shown to be decidable in [7] and NexpTime-complete in [13].

---

[1] I was unable to find a direct reference for this fact in the literature. However, it is not hard to show that $\mathrm{S5}^3$ is undecidable, since one can easily translate sentences of FO^3 to equisatisfiable formulas of $\mathrm{S5}^3$.

[2] This was noted at least in [18], although they did provide a reference for a proof of this claim. However, as we will see, the decidability of this system follows easily from already existing results.

7 Remarks on the decidability problem

In this final chapter we collect general results and remarks on the decidability problem. We will first show that the problem of determining whether a given fragment of FO is decidable is Σ^0_3-complete problem. After this we will study the question of whether having the full expressive power of FO will make a fragment of FO undecidable. It turns out that over general structures the answer is positive, while over finite structures the answer is negative. *Throughout this chapter we understand fragments of* FO *as being recursive subsets of the set of sentences of* FO.

7.1 Undecidability of decidability

In this section we study the problem of whether there exists an effective procedure for checking whether a given fragment of FO is decidable. Since a fragment is potentially an infinite set, an effective procedure for solving our problem can't simply receive the fragment as its input. To bypass this problem, we will define the input of the problem to be a description of a Turing machine that accepts (encodings of) those sentences of FO that belong to the given fragment.

Let us now formally define our decision problem. Suppose that we have fixed some reasonable encoding of the set of sentences of FO. Given a Turing machine M, we will use $L(M)$ to denote the language that consists of exactly those strings that M accepts. We say that a language L correspondence to a fragment of FO, if it is recursive and every string in L is an encoding of a sentence of FO. We now define our problem DECIDABLE as the following set.

$$\{M \mid M \text{ halts on every input and } L(M) \text{ correspondence to a decidable fragment of FO}\}.$$

Our goal is to prove that this problem is Σ^0_3-complete, i.e. it belongs to the third level of arithmetical hierarchy. For an introduction to the arithmetical hierarchy, see Lecture 35 in [15]. We will start by proving the upper bound for this problem.

Lemma 7.1. *DECIDABLE is in* Σ^0_3.

Proof. Let M denote the Turing machine that is given as an input to the problem and let X_M denote the corresponding fragment. Since the set of valid sentences of FO is recursively enumerable, X_M is decidable if and only if the set of satisfiable sentences of X_M is recursively enumerable. Let N denote a Turing machine which enumerates the set of valid sentences of FO. We will define five recursive predicates $TM(x)$, $H(x,t)$, $FO(x)$, $R(x,y,t)$, $M(x,t)$ and $N(x,t)$ as follows.

1. $TM(x)$ holds iff x is a description of a Turing machine.

2. $H(x,t)$ holds iff on input x the machine M halts in t-steps.

3. $FO(x)$ holds iff x is a description of a sentence of FO.

4. $R(x,y,t)$ holds iff x is a description of a Turing machine, and if we run the corresponding Turing machine for t-steps, then it prints out the string y.

5. $M(x,t)$ holds iff M accepts x in t-steps.

6. $N(x,t)$ holds iff x is a description of a sentence of FO and N has not printed out the negation of x after we have run it for t-steps.

Now the following sentence, which is not yet the Σ^0_3-description of the problem, expresses the property that the set of satisfiable sentences of X_M is recursively enumerable

$$\exists x(TM(x) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \wedge \psi_5),$$

where the formulas $\psi_1, \psi_2, \psi_3, \psi_4$ and ψ_5 are defined as follows.

$$\psi_1 := \forall y \exists t' H(y, t')$$

$$\psi_2 := \forall y \forall t (M(y, t) \to FO(y))$$

$$\psi_3 := \forall y \forall t (R(x, y, t) \to \exists t' M(y, t'))$$

$$\psi_4 := \forall y \forall t (R(x, y, t) \to \forall t' N(y, t'))$$

$$\psi_5 := \forall y (\exists t M(y, t) \to (\forall t' N(y, t') \to \exists t'' R(x, y, t'')))$$

The first two formulas make sure that everything that M always halts and M accepts only sentences of FO, which guarantees that $L(M)$ correspondence to a fragment of FO. The third formula makes sure that everything that (the Turing machine corresponding to) x prints out is accepted by M. The remaining formulas make sure that x prints out exactly those sentences ψ of X_M for which N does not print out $\neg\psi$. Now it is easy to transform this sentence into a sentence of the form $\exists x \forall y \forall t \forall t' \exists t'' \exists t''' \psi$, where ψ is quantifier-free formula, which is then the desired Σ^0_3-description of the problem. □

Let RECURSIVE denote the problem of determining whether $L(M)$ is recursive for a given Turing machine M. It is known that RECURSIVE is a Σ^0_3-hard problem (see for example Lecture 36 in [15]).

Lemma 7.2. *DECIDABLE is* Σ^0_3*-hard.*

Proof. We will give a reduction from the problem RECURSIVE. Thus we need to give a recursive mapping which associates to each Turing machine M a fragment X_M of FO so that $L(M)$ – the set of strings accepted by M – is recursive if and only if X_M is decidable. First we note that without loss of generality we can assume that M uses a single one-way infinite tape, the underlying vocabulary is $\{0, 1\}$ and that it has a unique accepting state. Now, it is routine to construct an effective procedure which maps each string $w \in \{0, 1\}^*$ to a sentence φ^w_M of FO so that M accepts w if and only if φ^w_M is unsatisfiable (for an example of such a construction, see the proof of Theorem 2.1.1 in [3]). As the fragment X_M we will choose the following set

$$\{\neg\varphi^w_M \mid w \in \{0, 1\}^*\}.$$

This set is clearly recursive. Now it is easy to see that $L(M)$ is recursive if and only if X_M is decidable. □

Corollary 7.3. *DECIDABLE is* Σ^0_3*-complete.*

7.2 Decidability and expressive power

In this section we study the question of how much correlation there exists between decidability and expressive power of fragments of FO. We will start by noting that if an arbitrary fragment of FO has the same expressive power as FO itself, then its satisfiability problem will be undecidable.

Proposition 7.4. *Let X be a fragment of* FO*. Suppose that X has the same expressive power as* FO*. Then X is undecidable.*

Proof. It suffices to show that to every given sentence $\varphi \in$ FO we can effectively associate an equivalent sentence of X. Since the set of sentences of FO is recursively enumerable, an effective procedure can simply start to enumerate the set of valid sentences of FO until it reaches a sentence of the form $(\varphi \leftrightarrow \psi)$. Since X is recursive, the procedure can verify whether $\psi \in X$ and return ψ if this is the case. In the other case the procedure will continue to enumerate the set of valid sentences. This procedure will eventually halt, since by assumption there does exists a sentence of X which is equivalent with φ. □

Since the previous proposition implies that for every decidable fragment of FO there exists a sentence of FO not expressible in this fragment, we can always extend the expressive power of a decidable FO fragment. The proof of the following proposition shows that we can always do this in such a way that we preserve the decidability of the original fragment.

Proposition 7.5. *Let X be a decidable* FO*-fragment. Then there exists another decidable* FO*-fragment which is strictly more expressive than X.*

Proof. Let X be a decidable fragment of FO. Using the previous proposition we conclude that there exists a sentence $\psi \in$ FO which is not equivalent to any sentence in X. We claim that the set $Y = X \cup \{\psi\}$ is a decidable fragment of FO. As the union of two recursive sets it is recursive. Furthermore the following is a simple effective procedure for verifying whether a given formula $\varphi \in Y$ is decidable. First the procedure checks whether $\varphi = \psi$. If this is the case, then the procedure accepts if and only if ψ is satisfiable (this information can be assumed to be part of the procedure, since ψ is fixed). Otherwise the procedure runs the procedure for X on φ and returns the answer returned by this procedure. Since Y is clearly strictly more expressive than X the claim follows. □

If we restrict our attention to finite models, then it is known that the set of valid sentences of FO is no longer recursively enumerable [4]. This raises the question of whether proposition 7.4 continues to hold if our attention is restricted to finite models. As the final result of this section we show that it indeed does fail.

Proposition 7.6. *There exists a fragment of* FO *for which the finite satisfiability problem is decidable in constant time and which is equi-expressive with* FO *over finite models.*

Proof. Let θ_n denote the formula $\exists v_1 ... \exists v_n v_1 = v_1$ and consider the following set

$$X = \{\exists v_1 \neg v_1 = v_1\} \cup \{(\theta_n \wedge \varphi) \mid \varphi \in \text{FO}, \varphi \text{ has a finite model of size } n.\}$$

We claim that X is the desired fragment of FO. We start by showing that X is recursive. Given a sentence $\varphi \in$ FO, our procedure will check whether φ is of the form $\exists v_1 \neg v_1 = v_1$ or of the form $(\theta_n \wedge \varphi)$. In the first case it accepts, while in the second it will go through all models of size n of some fixed domain and check whether any of them is a model of φ. If such a model exists then the procedure accepts, and otherwise the procedure rejects.

Next we note that it is not hard to see that the finite satisfiability problem for X is decidable in constant time, since a procedure can simply check whether the given sentence is $\exists v_1 \neg v_1 = v_1$. Finally it is also not hard to see that X is equi-expressive with FO over finite models, since each satisfiable sentence φ is just equivalent with $\theta_n \wedge \varphi$, for some θ_n, while each contradiction is equivalent with the sentence $\exists v_1 \neg v_1 = v_1$. This completes the proof. □

Remark. In the above proof we used very little the fact that we were dealing with first-order logic. One could for instance replace first-order logic by second-order logic (or any higher-order logic) and the same result would still hold. Indeed, it seems that the most vital properties of the logic are that it has a recursive syntax and its model checking problem is decidable over finite models.

What is the precise content of proposition 7.6? In some sense it is saying that there is no need to search for stronger and stronger fragments of FO with a decidable finite satisfiability problem, since there is at least one decidable fragment of FO that has the same expressive power as FO over finite models. However, my own interpretation is that this proposition shows that our definition of fragment is too general, since the fragment X that was considered in the proof of proposition 7.6 is certainly not a natural one.

Although X is not a natural fragment, it is not immediately clear what is the precise reason that makes it unnatural (except the fact that it is far from being practical). Lauri Hella pointed out to me that the definition of X is in some sense *semantical*. This raises the question of whether there exists a fragment L of FO with the following properties.

1. The definition of L is syntactical.

2. L is equi-expressive with FO over finite models.

3. The finite satisfiability problem for L is decidable over finite models.

The first requirement is, of course, vague. Nevertheless there is a clear intuition behind this requirement: if the definition of a fragment L is syntactical, then whether or not a given sentence φ of FO belongs to L depends only on the *form* of φ. This definition excludes the fragment X used in the proof of proposition 7.6, since whether or not a given sentence $(\theta_n \wedge \varphi)$ of FO belongs to L depends on an external *semantical requirement*. We also point out that there is no obvious way of turning X into a fragment which satisfies the aforementioned requirements 1,2 and 3. For instance, we can't replace $(\theta_n \wedge \varphi)$ with a sentence of the form $(\Theta_n \wedge \varphi)$, where Θ_n describes (up to isomorphism) a single structure with n elements. This would lead to a decidable fragment with a purely syntactical definition, but it would violate the second requirement, since the sentence $(\Theta_n \wedge \varphi)$ is not equivalent with φ.

We conclude this section by sketching one way of making the first requirement precise using the algebraic framework that we have studied in this thesis. Namely, if $\mathcal{F}$ is a finite set of relation operators that are in some sense FO-definable, then $\mathrm{GRA}(\mathcal{F})$ correspondence to a fragment of FO with a purely syntactical definition. Here we will not attempt to make the concept of "FO-definable relation operator" precise, but if one can formally define it, then one can also formalize the following statement: there exists a finite set $\mathcal{F}$ of FO-definable relation operations so that the finite satisfiability problem is decidable for $\mathrm{GRA}(\mathcal{F})$ and $\mathrm{GRA}(\mathcal{F})$ has the same expressive power as FO over finite models. We leave the precise formulation of this statement for future work.

Bibliography

[1] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[2] Mikolaj Bojanczyk et al. "Two-variable logic on data trees and XML reasoning". In: *Journal of the ACM* 56.3 (2009), 13:1–13:48.

[3] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[4] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer Monographs in Mathematics. Springer, 1999.

[5] Erich Grädel. "On The Restraining Power of Guards". In: *Journal of Symbolic Logic* 64.4 (1999), pp. 1719–1742.

[6] Erich Grädel, Phokion Kolaitis, and Moshe Vardi. "On the decision problem for two-variable first-order logic". In: *Bulletin of Symbolic Logic* 3.1 (1997), pp. 53–69.

[7] Lauri Hella and Antti Kuusisto. "One-dimensional Fragment of First-order Logic". In: *Advances in Modal Logic 10 AiML, invited and contributed papers from the tenth conference on "Advances in Modal Logic"*. Ed. by Rajeev Goré, Barteld P. Kooi, and Agi Kurucz. College Publications, 2014, pp. 274–293.

[8] Andreas Herzig. "A new decidable fragment of first order logic". In: *Abstracts of the 3rd Logical Biennial Summer School and Conference in Honour of S. C.Kleene* (1990).

[9] Robin Hirsch and Ian Hodkinson. *Relation algebras by games*. North Holland, 2002.

[10] Reijo Jaakkola. "Ordered fragments of first-order logic". In: *arXiv Preprint* arXiv:2103.08046 (2021).

[11] Reijo Jaakkola and Antti Kuusisto. "Algebraic classifications for fragments of first-order logic and beyond". In: *arXiv Preprint* arXiv:2005.01184 (2020).

[12] Reijo Jaakkola and Antti Kuusisto. "Algebraic classifications for fragments of first-order logic and beyond". In: *arXiv Preprint* arXiv:2005.01184v2 (2021).

[13] Emanuel Kieronski and Antti Kuusisto. "Complexity and Expressivity of Uniform One-Dimensional Fragment with Equality". In: *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS, Proceedings, Part I*. Ed. by Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik. Vol. 8634. Lecture Notes in Computer Science. Springer, 2014, pp. 365–376.

[14] Emanuel Kieronski et al. "Two-Variable First-Order Logic with Equivalence Closure". In: *SIAM Journal on Computing* 43.3 (2014), pp. 1012–1063.

[15] Dexter C. Kozen. *Theory of Computation (Texts in Computer Science)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 1846282977.

[16] Antti Kuusisto. "On Games and Computation". In: *CoRR* abs/1910.14603 (2019).

[17] Dirk Leinders et al. "The Semijoin Algebra and the Guarded Fragment." In: *Journal of Logic, Language and Information* 14.3 (2005), pp. 331–343.

[18] Maarten Marx and Yde Venema. "Local Variations on a Loose Theme: Modal Logic and Decidability, in Finite Model Theory and Its Applications". In: Springer, 2007.

[19] Maarten Max. “Complexity of products of modal logics”. In: *Journal of Logic and Computation* 64 (1999), pp. 1563–1572.

[20] Martin Otto. “Two variable first-order logic over ordered domains”. In: *Journal of Symbolic Logic* 66.2 (2001), pp. 685–702.

[21] Ian Pratt-Hartmann, Wieslaw Szwast, and Lidia Tendera. “The Fluted Fragment Revisited”. In: *Journal of Symbolic Logic* 84.3 (2019), pp. 1020–1048.

[22] Willard Van Quine. “Algebraic logic and predicate functors”. In: *Logic and Art*. Bobbs-Merrill, Indianapolis, Indiana, 1972, pp. 214–238.

[23] Willard Van Quine. “On the limits of decision”. In: *Proceedings of the 14th International Congress of Philosophy*. Vol. III. University of Vienna, 1969, pp. 57–62.

[24] Willard Van Quine. “Toward a calculus of concepts”. In: *The Journal of Symbolic Logic* I (1936), pp. 2–25.

[25] Willard Van Quine. “Variables explained away”. In: *Proceedings of the American Philosophical Society*. 1960.

[26] Moshe Vardi. “What makes a modal logic so robustly decidable?, in Descriptive Complexity and Finite Models”. In: (1997).

[27] Marco Voigt. “Separateness of Variables – A Novel Perspective on Decidable First-Order Fragments”. In: *arXiv Preprint* arXiv:1911.11500 (2019).