*Fundamenta Informaticae 167 (2019) 287–321*
*DOI 10.3233/FI-2019-1819*
*IOS Press* 

# Universal (and Existential) Nulls

**Gösta Grahne***, **Ali Moallemi**

*Department of Computer Science and Software Engineering*
*Concordia University*
*Montreal, Canada*
*grahne@cs.concordia.ca, moa_ali@encs.concordia.ca*

**Abstract.** Incomplete Information research is quite mature when it comes to so called *existential nulls*, where an existential null is a value stored in the database, representing an unknown object. For some reason *universal nulls*, that is, values representing *all* possible objects, have received almost no attention. We remedy the situation in this paper, by showing that a suitable finite representation mechanism, called *Star Cylinders*, handling universal nulls can be developed based on the *Cylindric Set Algebra* of Henkin, Monk and Tarski. We provide a finitary version of the cylindric set algebra, called *Cylindric Star Algebra*, and show that our star-cylinders are closed under this algebra. Moreover, we show that any *First Order Relational Calculus* query over databases containing universal nulls can be translated into an equivalent expression in our cylindric star-algebra, and vice versa. All cylindric star-algebra expressions can be evaluated in time polynomial in the size of the database.
The representation mechanism is then extended to *Naive Star Cylinders*, which are star-cylinders allowing existential nulls in addition to universal nulls. For positive queries (with universal quantification), the well known naive evaluation technique can still be applied on the existential nulls, thereby allowing polynomial time evaluation of certain answers on databases containing both universal and existential nulls. If precise answers are required, certain answer evaluation with universal and existential nulls remains in coNP. Note that the problem is coNP-hard, already for positive existential queries and databases with only existential nulls. If inequalities $\neg(x_i \approx x_j)$ are allowed, reasoning over existential databases is known to be $\Pi_2^p$-complete, and it remains in $\Pi_2^p$ when universal nulls and full first order queries are allowed.

**Keywords:** Relational Algebra, Cylindric Set Algebra, Incomplete Information, Query Languages

*Address for correspondence: Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.

Received April 2018; revised May 2019.

# 1. Introduction

In this paper we revisit the foundations of the relational model and unearth *universal nulls*, showing that they can be treated on par with the usual *existential nulls* [1, 2, 3]. Recall that an existential null in a tuple in a relation $R$ represents an existentially quantified variable in an atomic sentence $R(\ldots)$. This corresponds to the intuition "value exists, but is unknown." A universal null, on the other hand, does not represent anything unknown, but stands for *all* values of the domain. In other words, a universal null represents a universally quantified variable. Universal nulls have an obvious application in databases, as the following example shows. The symbol "$*$" denotes a universal null.

**Example 1.1.** Consider binary relations $F$*(ollows)* and $H$*(obbies)*, where $F(x, y)$ means that user $x$ follows user $y$ on a social media site, and $H(x, z)$ means that $z$ is a hobby of user $x$. Let the database be the following.

| $F$ | |
|---|---|
| Alice | Chris |
| $*$ | Alice |
| Bob | $*$ |
| Chris | Bob |
| David | Bob |

| $H$ | |
|---|---|
| Alice | Movies |
| Alice | Music |
| Bob | Basketball |

This is to be interpreted as expressing the facts that Alice follows Chris and Chris and David follow Bob. Alice is a journalist who would like to give access to everyone to articles she shares on the social media site. Therefore, everyone can follow Alice. Bob is the site administrator, and is granted the access to all files anyone shares on the site. Consequently, Bob follows everyone. "Everyone" in this context means all current and possible future users. The query below, in domain relational calculus, asks for the interests of people who are followed by everyone:

$$x_4 \;.\; \exists x_2 \exists x_3 \forall x_1 \Big( F(x_1, x_2) \wedge H(x_3, x_4) \wedge (x_2 \approx x_3) \Big). \tag{1}$$

The answer to our example query is $\{(\text{Movies}), (\text{Music})\}$. Note that "*"-nulls also can be part of an answer. For instance, the query $x_1, x_2 \,.\, F(x_1, x_2)$ would return all the tuples in $F$. ◂

Another area of applications of "*"-nulls relates to intuitionistic, or constructive database logic. In the constructive four-valued approach of [4] and the three-valued approach of [3, 5] the proposition $A \vee \neg A$ is not a tautology. In order for $A \vee \neg A$ to be true, we need either a constructive proof of $A$ or a constructive proof of $\neg A$. Therefore both [4] and [5] assume that the database $I$ has a theory of the negative information, i.e. that $I = (I^+, I^-)$, where $I^+$ contains the positive information and $I^-$ the negative information. The papers [4] and [5] then show how to transform an FO-query $\varphi(\bar{x})$ to a pair of queries $(\varphi^+(\bar{x}), \varphi^-(\bar{x}))$ such that $\varphi^+(\bar{x})$ returns the tuples $\bar{a}$ for which $\varphi(\bar{a})$ is true in $I^+$, and $\varphi^-(\bar{x})$ returns the tuples $\bar{a}$ for which $\varphi(\bar{a})$ is true in $I^-$ (i.e. $\varphi(\bar{a})$ is false in $I$). It turns out that databases containing "*"-nulls are suitable for storing $I^-$.

**Example 1.2.** Suppose that the instance in Example 1.1 represents $I^+$, and that all negative information we have deduced about the $H(obbies)$ relation, is that we know Alice doesn't play Volleyball, that Bob only has Basketball as hobby, and that Chris has no hobby at all. This negative information about the relation $H$ is represented by the table $H^-$ below. Note that $H^-$ is part of $I^-$.

| $H^-$ | |
|---|---|
| Alice | Volleyball |
| Bob | $*$ (except Basketball) |
| Chris | $*$ |

Suppose the query $\varphi$ asks for people who have a hobby, that is $\varphi = x_1 \,.\, \exists x_2\, H(x_1, x_2)$. Then $\varphi^+ = \varphi$, and $\varphi^- = x_1 \,.\, \forall x_2\, H(x_1, x_2)$. Evaluating $\varphi^+$ on $I^+$ returns $\{$(Alice), (Bob)$\}$, and evaluating $\varphi^-$ on $I^-$ returns $\{$(Chris)$\}$. Note that there is no closed-world assumption as the negative facts are explicit. Thus it is unknown whether David has a hobby or not.

Universal nulls were first studied in the early days of database theory by Biskup in [6]. This was a follow-up on his earlier paper on existential nulls [7]. The problem with Biskup's approach, as noted by himself, was that the semantics for his algebra worked only for individual operators, not for compound expressions (i.e. queries). This was remedied in the foundational paper [1] by Imielinski and Lipski, as far as existential nulls were concerned. Universal nulls next came up in [8], where Imielinski and Lipski showed that Codd's Relational Algebra could be embedded in CA, the *Cylindric Set Algebra* of Henkin, Monk, and Tarski [9, 10]. As a side remark, Imielinski and Lipski suggested that the semantics of their "$*$" symbol could be seen as modeling the universal null of Biskup. In this paper we follow their suggestion[1], and fully develop a finitary representation mechanism for databases with universal nulls, as well as an accompanying finitary algebra. We show that any FO (First Order / Domain Relational Calculus) query can be translated into an equivalent expression in a finitary version of CA, and that such algebraic expressions can be evaluated "naively" by the rules "$* = *$" and "$* = a$" for any constant "$a$." Our finitary version is called *Cylindric Star Algebra* (SCA) and operates on finite relations containing constants and universal nulls "$*$." These relations are called *Star Cylinders* and they are finite representations of a subclass of the infinite cylinders of Henkin, Monk, and Tarski. Interestingly, the class of star-cylinders is closed under first order querying, meaning that the infinite result of an FO query on an infinite instance represented by a finite sequence of finite star-cylinders can be represented by a finite star-cylinder.[2] This is achieved by showing that the class of star-cylinders are closed under our cylindric star-algebra, and that SCA as a query language is equivalent in expressive power with FO.

The Cylindric Set Algebra [9, 10] —as an algebraization of first order logic— is an algebra on sets of valuations of variables in an FO-formula. A *valuation* $\nu$ of variables $\{x_1, x_2, \ldots\}$ can be represented as a tuple $\nu$, where $\nu(i) = \nu(x_i)$. The set of all valuations can then be represented by a relation $C$ of such tuples. In particular, if the FO-formula only involves a finite number $n$ of variables, then the

[1]We note that Sundarmurthy et. al. [11] very recently have proposed a construct related to our universal nulls, and studied ways on placing constraints on them.

[2]Consequently there is no need to require calculus queries to be "domain independent."

representing relation $C$ has arity $n$. Note however that $C$ has an infinite number of tuples, since the domain $\mathbb{D}$ of the variables (such as the users of a social media site), should be assumed unbounded. One of the basic connections [9, 10] between FO and Cylindric Set Algebra is that, given any interpretation $I$ and FO-formula $\varphi$, the set of valuations $\nu$ under which $\varphi$ is true in $I$ can be represented as such a relation $C$. Moreover, each logical connective and quantifier corresponds to an operator in the Cylindric Set Algebra. Naturally disjunction corresponds to union, conjunction to intersection, and negation to complement. More interestingly, existential quantification on variable $x_i$ corresponds to *cylindrification* $\mathsf{c}_i$ on column $i$, where

$$\mathsf{c}_i(C) \;=\; \{\nu : \nu(i/a) \in C, \text{ for some } a \in \mathbb{D}\},$$

and $\nu(i/a)$ denotes the valuation (tuple) $\nu'$, where $\nu'(i) = a$ and $\nu'(j) = \nu(j)$ for $i \neq j$. The algebraic counterpart of universal quantification can be derived from cylindrification and complement, or be defined directly as *inner cylindrification*

$$\mathsf{ɔ}_i(C) \;=\; \{\nu : \nu(i/a) \in C, \text{ for all } a \in \mathbb{D}\}.$$

In addition, in order to represent equality, the Cylindric Set Algebra also contains constant relations $d_{ij}$ representing the equality $x_i \approx x_j$. That is, $d_{ij}$ is the set of all valuations $\nu$, such that $\nu(i) = \nu(j)$.

The objects $C$ and $d_{ij}$ of [9, 10] are of course infinitary. In this paper we therefore develop a finitary representation mechanism, namely relations containing universal nulls "$*$" and certain equality literals. We denote these finitary constructs $\dot{C}$ and $\dot{d}_{ij}$, respectively. These objects are called *Star Tables* when they represent the records stored in the database. When used as run-time constructs in algebraic query evaluation, they will be called *Star Cylinders*. Example 1.1 showed star-tables in a database. The run-time variable binding pattern of the query (1), as well as its algebraic evaluation is shown in the star-cylinders in Example 1.3 below.

**Example 1.3.** Continuing Example 1.1, in that database the atoms $F(x_1, x_2)$ and $H(x_3, x_4)$ of query (1) are represented by star-tables $\dot{C}_F$ and $\dot{C}_H$, and the equality atom is represented by the star-diagonal $\dot{d}_{23}$. Note that these are positional relations, the "attributes" $x_1, x_2, x_3, x_4$ are added for illustrative purposes only.

$\dot{C}_F$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| Alice | Chris | $*$ | $*$ |
| $*$ | Alice | $*$ | $*$ |
| Bob | $*$ | $*$ | $*$ |
| Chris | Bob | $*$ | $*$ |

$\dot{C}_H$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| $*$ | $*$ | Alice | Movies |
| $*$ | $*$ | Alice | Music |
| $*$ | $*$ | Bob | Basketball |

$\dot{d}_{23}$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | |
|---|---|---|---|---|
| $*$ | $*$ | $*$ | $*$ | 2=3 |

The algebraic translation of query (1) is the expression

$$\dot{\mathsf{c}}_2(\dot{\mathsf{c}}_3(\dot{\text{ɔ}}_1((\dot{C}_F \dot{\cap} \dot{C}_H) \dot{\cap} \dot{d}_{23}))). \tag{2}$$

The intersection of $\dot{C}_F$ and $\dot{C}_H$ is carried out as star-intersection $\dot{\cap}$, where for instance $\{(a,*,*)\} \dot{\cap} \{(*,b,*)\} = \{(a,b,*)\}$. The result will contain 12 tuples, and when these are star-intersected with $\dot{d}_{23}$, the star-diagonal $\dot{d}_{23}$ will act as a selection by columns 2 and 3 being equal. The result is the star-cylinder $\dot{C}' = (\dot{C}_F \dot{\cap} \dot{C}_H) \dot{\cap} \dot{d}_{23}$ below.

$\dot{C}'$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| $*$ | Alice | Alice | Movies |
| $*$ | Alice | Alice | Music |
| Bob | Alice | Alice | Movies |
| Bob | Alice | Alice | Music |
| Bob | Bob | Bob | Basketball |
| Chris | Bob | Bob | Basketball |

The inner star-cylindrification on column 1 then yields $\dot{C}'' = \dot{\text{ɔ}}_1((\dot{C}_F \dot{\cap} \dot{C}_H) \dot{\cap} \dot{C}_{23})$.

$\dot{C}''$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| $*$ | Alice | Alice | Movies |
| $*$ | Alice | Alice | Music |

Finally, applying outer star-cylindrifications on columns 2 and 3 of star-cylinder $C''$ yields the final result $C''' = \dot{\mathsf{c}}_2(\dot{\mathsf{c}}_3(\dot{\text{ɔ}}_1((C_F \dot{\cap} C_H) \dot{\cap} C_{23})))$.

$\dot{C}'''$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| $*$ | $*$ | $*$ | Movies |
| $*$ | $*$ | $*$ | Music |

The system can now return the answer, i.e. the values of column 4 in cylinder $\dot{C}'''$. Note that columns where all rows are "$*$" do not actually have to be materialized at any stage. Negation requires some additional details that will be introduced in Section 3.2.

**Paper outline.** The aim of this paper is to develop a clean and sound modelling of universal nulls, and furthermore show that the model can be seamlessly extended to incorporate the existential nulls of Imielinski and Lipski [1]. We show that FO and our SCA are equivalent in expressive power when it comes to querying databases containing universal nulls, and that SCA queries can be evaluated (semi) naively. This will be done in three steps: In Section 2 we show the equivalence between FO and Cylindric Set Algebra over infinitary databases. This was of course only the starting point of [9, 10], and we recast the result here in terms of database theory.[3] In Section 3 we introduce our finitary Cylindric Star Algebra. Section 3.1 develops the machinery for the positive case, where there is no negation in the query or database. This is then extended to include negation in Section 3.2. By these two sections we show that certain infinitary cylinders can be finitely represented as star-cylinders, and that our finitary Cylindric Star Algebra on finite star-cylinders mirrors the Cylindric Set Algebra on the infinite cylinders they represent. In Section 4 we tie these two results together, delivering the promised SCA evaluation of FO queries on databases containing universal nulls. In Section 5 we seamlessly extend our framework to also handle existential nulls, and show that naive evaluation can still be used for positive queries (allowing universal quantification, but not negation) on databases containing both universal and existential nulls. Section 6 then shows that all SCA expressions can be evaluated in time polynomial in the size of the database when only universal nulls are present. We also show that when both universal and existential nulls are present, the certain answer to any negation-free (allowing inner cylindrification, i.e. universal quantification) SCA-query can be evaluated naively in polynomial time. When negation is present it has long been known that the problem is coNP-complete for databases containing existential nulls. We show that the problem remains coNP-complete when universal nulls are allowed in addition to the existential ones. For databases containing existential nulls it has been known that database containment and view containment are coNP-complete and $\Pi_2^p$-complete, respectively. We also show that the addition of universal nulls does not increase these complexities.

## 2. Relational calculus and cylindric set algebra

Throughout this paper we assume a fixed schema $\mathscr{R} = \{R_1, \ldots, R_m, \approx\}$, where each $R_p$, $p \in \{1, \ldots, m\}$, is a relational symbol with an associated positive integer $ar(R_p)$, called the *arity* of $R_p$. The symbol $\approx$ represents equality.

**Logic.** Our calculus is the standard domain relational calculus. Let $\{x_1, x_2, \ldots\}$ be a countably infinite set of *variables*. We define the set of *FO-formulas* $\varphi$ (over $\mathscr{R}$) in the usual way: $R_p(x_{i_1}, \ldots, x_{i_{ar(R_p)}})$ and $x_i \approx x_j$ are atomic formulas, and these are closed under $\wedge, \vee, \neg, \exists x_i$, and $\forall x_i$, in a well-formed manner possibly using parentheses for disambiguation.

Let $\varphi$ be an FO-formula. We denote by $vars(\varphi)$ the set of variables in $\varphi$, by $free(\varphi)$ the set of free variables in $\varphi$, and by $sub(\varphi)$ the set of subformulas of $\varphi$ (for formal definitions, see [13]). If $\varphi$ has $n$ variables we say that $\varphi$ is an *FO$_n$-formula*. We assume without loss of generality that each variable occurs only once in the formula, except in equality literals, and that a formula with $n$ variables uses variables $x_1, \ldots, x_n$.

**Instances.** Let $\mathbb{D} = \{a_1, a_2, \ldots\}$ be a countably infinite *domain*. An *instance* $I$ (over $\mathscr{R}$) is a mapping that assigns a possibly infinite subset $R_p^I$ of $\mathbb{D}^{ar(R_p)}$ to each relation symbol $R_p$, and $\approx^I = \{(a, a) : a \in \mathbb{D}\}$. Note that our instances are infinite model-theoretic ones. The set of tuples actually recorded in the database will be called the *stored database* (to be defined in Section 4).

[3] Van Den Bussche [12] has recently referred to [9, 10] in similar terms.

In order to define the (standard) notion of truth of an $\mathrm{FO}_n$-formula $\varphi$ in an instance $I$ we first define a *valuation* to be a mapping $\nu : \{x_1, \dots, x_n\} \to \mathbb{D}$. If $\nu$ is a valuation, $x_i$ a variable and $a \in \mathbb{D}$, then $\nu_{(i/a)}$ denotes the valuation which is the same as $\nu$, except $\nu_{(i/a)}(x_i) = a$. Then we use the usual recursive definition of $I \vDash_\nu \varphi$, meaning *instance $I$ satisfying $\varphi$ under valuation $\nu$*, i.e. $I \vDash_\nu (x_i \approx x_j)$ if $(\nu(x_i), \nu(x_j)) \in \approx^I$, $I \vDash_\nu R_p(x_{i_1}, \dots, x_{i_{ar(R_p)}})$ if $(\nu(x_{i_1}), \dots, \nu(x_{i_{ar(R_p)}})) \in R_p^I$, and $I \vDash_\nu \exists x_i \varphi$ if $I \vDash_{\nu_{(i/a)}} \varphi$ for some $a \in \mathbb{D}$, and so on. Our stored databases will be finite representations of infinite instances, so the semantics of answers to FO-queries will be defined in terms of the infinite instances:

**Definition 2.1.** Let $I$ be an instance, and $\varphi$ an $\mathrm{FO}_n$-formula with $free(\varphi) = \{x_{i_1}, \dots, x_{i_k}\}$, $k \leq n$. Then the *answer to $\varphi$ on $I$* is defined as

$$\varphi^I = \{(\nu(x_{i_1}), \dots, \nu(x_{i_k})) \ : \ I \vDash_\nu \varphi\}.$$

**Algebra.** As noted in [8] the relational algebra is really a disguised version of the Cylindric Set Algebra of Henkin, Monk, and Tarski [9, 10]. We shall therefore work directly with the Cylindric Set Algebra instead of Codd's Relational Algebra. Apart from the conceptual clarity, the Cylindric Set Algebra will also allow us to smoothly introduce the promised universal nulls.

Let $n$ be a fixed positive integer. The basic building block of the Cylindric Set Algebra is an *$n$-dimensional cylinder* $C \subseteq \mathbb{D}^n$. Note that a cylinder is essentially an infinite $n$-ary relation. They will however be called cylinders, in order to distinguish them from instances. The rows in a cylinder will represent run-time variable valuations, whereas tuples in instances represent facts about the real world. We also have special cylinders called *diagonals*, of the form

$$d_{ij} = \{t \in \mathbb{D}^n \ : \ t(i) = t(j)\},$$

representing the equality $x_i \approx x_j$. We can now define the Cylindric Set Algebra.

**Definition 2.2.** Let $C$ and $C'$ be infinite $n$-dimensional cylinders. The *Cylindric Set Algebra* consists of the following operators.

1. *Union:* $C \cup C'$. Set theoretic union.
2. *Complement:* $\overline{C} = \mathbb{D}^n \setminus C$.
3. *Outer cylindrification:* $\mathsf{c}_i(C) = \{t \in \mathbb{D}^n \ : \ t(i/a) \in C, \text{ for some } a \in \mathbb{D}\}$.

The operation $\mathsf{c}_i$ is called outer cylindrification on the $i$:th dimension, and will correspond to existential quantification of variable $x_i$. For the geometric intuition behind the name cylindrification, see [9, 8]. Intersection is considered a derived operator, and we also introduce the following derived operators:

4. *Inner cylindrification:* $\mathsf{ɔ}_i(C) = \overline{\mathsf{c}_i(\overline{C})}$, corresponding to universal quantification. Note that

$$\mathsf{ɔ}_i(C) = \{t \in \mathbb{D}^n \ : \ t(i/a) \in C, \text{ for all } a \in \mathbb{D}\}.$$

5. *Substitution:* $\mathsf{s}_i^j(C) = \mathsf{c}_i(\mathsf{d}_{ij} \cap C)$ if $i \neq j$, and $\mathsf{s}_i^i(C) = C$.
6. *Swapping:* If $i, j \neq k$, and $C$ is a $k$-full cylinder[4], then $\mathsf{z}_j^i(C) = \mathsf{s}_i^k(\mathsf{s}_j^i(\mathsf{s}_k^j(C)))$. In other words, $\mathsf{z}_j^i(C)$ interchanges the values of dimensions $i$ and $j$. We also define $\mathsf{z}_{j_1,j_2}^{i_1,i_2}(C) = \mathsf{z}_{j_1}^{i_1}(\mathsf{z}_{j_2}^{i_2}(C))$, and recursively $\mathsf{z}_{j_1,\ldots,j_k}^{i_1,\ldots,i_k}(C) = \mathsf{z}_{j_1}^{i_1}(\mathsf{z}_{j_2,\ldots,j_k}^{i_2,\ldots,i_k}(C))$.

We also need the notion of cylindric set algebra expressions.

**Definition 2.3.** Let $\mathbf{C} = (C_1, \ldots, C_m, d_{ij})_{i,j \in \{1,\ldots,n\}}$ be a sequence of infinite $n$-dimensional cylinders and diagonals. The set of *CA$_n$-expressions* (over $\mathbf{C}$) is obtained by closing the atomic expressions $\mathsf{C}_p$ and $\mathsf{d}_{ij}$ under union, intersection, complement, and inner and outer cylindrifications. Then $E(\mathbf{C})$, the *value of expression $E$ on sequence* $\mathbf{C}$ is defined in the usual way, e.g. $\mathsf{C}_p(\mathbf{C}) = C_p$, $\mathsf{d}_{ij}(\mathbf{C}) = d_{ij}$, $\mathsf{c}_i(E)(\mathbf{C}) = \mathsf{c}_i(E(\mathbf{C}))$ etc.

**Equivalence of FO and CA.** In the next two theorems we will restate, in the context of the relational model, the correspondence between domain relational calculus and cylindric set algebra as query languages on instances [9, 10]. When translating an FO$_n$-formula to a CA$_n$-expression we first need to extend all $k$-ary relations in $I$ to $n$-ary by filling the $n - k$ last columns in all possible ways. Formally, this is expressed as follows:

**Definition 2.4.** The *horizontal $n$-expansion* of an infinite $k$-ary relation $R$ is

$$\mathsf{h}^n(R) = R \times \mathbb{D}^{n-k}.$$

The equality relation $\approx^I = \{(a, a) : a \in \mathbb{D}\}$ is expanded into diagonals $d_{ij}$ for $i, j \in \{1 \ldots, n\}$, where

$$d_{ij} = \bigcup_{(a,a) \in \approx^I} \mathbb{D}^{i-1} \times \{a\} \times \mathbb{D}^{j-i-1} \times \{a\} \times \mathbb{D}^{n-j},$$

and for an instance $I = (R_1^I, \ldots, R_m^I, \approx^I)$, we have

$$\mathsf{h}^n(I) = (\mathsf{h}^n(R_1^I), \ldots, \mathsf{h}^n(R_m^I), d_{ij})_{i,j}.$$

Once an instance is expanded it becomes a sequence $\mathbf{C} = (C_1, \ldots, C_m, d_{ij})_{i,j}$ of $n$-dimensional cylinders and diagonals, on which Cylindric Set Algebra Expressions can be applied. ◂

The main technical difficulty in the translation from FO$_n$ to CA$_n$ is the correlation of the variables in the FO$_n$-sentence $\varphi$ with the columns in the expanded relations in the instance. This can be achieved using the swapping operator $\mathsf{z}_j^i$. Every atom $R_p$ in $\varphi$ will correspond to a CA$_n$-expression

[4]Cylinder $C$ is $k$-full if $\mathsf{c}_k(C) = C$. A cylinder with this property is called *dimension complemented* in [9, 10]. In a $k$-full cylinder $C$ the dimension $k$ can be used to temporarily store the content of another dimension. This allows the definition of the swapping operators in terms of the substitution operators, which in turn are defined through intersection, diagonal, and outer cylindrification. Following [9, 10] we therefore do not need to define swapping or substitution as primitives, which would require corresponding renaming operators in the language for FO.

$C_p = \mathsf{h}^n(R_p^I)$. However, for every occurrence of an atom $R_p(x_{i_1}, \ldots, x_{i_k})$ in $\varphi$ we need to interchange the columns $1, \ldots, k$ with columns $i_1, \ldots, i_k$. This is achieved by the expression $\mathsf{z}^{1,\ldots,k}_{i_1,\ldots,i_k}(\mathsf{C}_p)$. The entire FO$_n$-formula $\varphi$ with $free(\varphi) = \{x_{i_1}, \ldots, x_{i_k}\}$ will then correspond to the CA$_n$-expression $E_\varphi = \mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}(F_\varphi)$, where $F_\varphi$ is defined recursively as follows:

- If $\varphi = R_p(x_{i_1}, \ldots, x_{i_k})$ where $k = ar(R_p)$, then $F_\varphi = \mathsf{z}^{1,\ldots,k}_{i_1,\ldots,i_k}(\mathsf{C}_p)$.
- If $\varphi = x_i \approx x_j$, then $F_\varphi = \mathsf{d}_{ij}$.
- If $\varphi = \psi \vee \chi$, then $F_\varphi = F_\psi \bigcup F_\chi$, if $\varphi = \psi \wedge \chi$, then $F_\varphi = F_\psi \bigcap F_\chi$, and if $\varphi = \neg\,\psi$, then $F_\varphi = \overline{F_\psi}$.
- If $\varphi = \exists x_i \psi$, then $F_\varphi = \mathsf{c}_i(F_\psi)$.
- If $\varphi = \forall x_i \psi$, then $F_\varphi = \mathsf{ɔ}_i(F_\psi)$.

For an example, let us reformulate the $FO_4$-query $\varphi$ from (1) as

$$x_4 \,.\, \exists x_2 \exists x_3 \forall x_1 \Big(R_1(x_1, x_2) \wedge R_2(x_3, x_4) \wedge (x_2 \approx x_3)\Big). \tag{3}$$

When translating $\varphi$ the relation $R_1^I$ is first expanded to $C_1 = R_1^I \times \mathbb{D} \times \mathbb{D}$, and $R_2^I$ is expanded to $C_2 = R_2^I \times \mathbb{D} \times \mathbb{D}$. In order to correlate the variables in $\varphi$ with the columns in the expanded databases, we do the shifts $\mathsf{z}^{1,2}_{1,2}(C_1)$ and $\mathsf{z}^{1,2}_{3,4}(C_2)$. The equality $(x_2 \approx x_3)$ was expanded to the diagonal $d_{23} = \{t \in \mathbb{D}^n : t(2) = t(3)\}$ so here the variables are already correlated. After this the conjunctions are replaced with intersections and the quantifiers with cylindrifications. Finally, the column corresponding to the free variable $x_4$ in $\varphi$ (whose bindings will constitute the answer) is shifted to column 1. The final CA$_n$-expression will then be evaluated against $I$ as $E_\varphi(\mathsf{h}^4(I)) =$

$$\mathsf{z}^4_1(\mathsf{c}_2(\mathsf{c}_3(\mathsf{ɔ}_1\Big(\mathsf{z}^{1,2}_{1,2}(R_1^I \times \mathbb{D}^2) \bigcap \mathsf{z}^{1,2}_{3,4}(R_2^I \times \mathbb{D}^2) \bigcap d_{23}))\Big)). \tag{4}$$

We now have $E_\varphi(\mathsf{h}^4(I)) = \mathsf{h}^4(\varphi^I)$. The following fundamental result follows from [9, 10]. A proof is included in the Appendix for the benefit of the readers who don't want to consult [9, 10].

**Theorem 2.5.** For all FO$_n$-formulas $\varphi$, there is a CA$_n$-expression $E_\varphi$, such that

$$E_\varphi(\mathsf{h}^n(I)) \;=\; \mathsf{h}^n(\varphi^I),$$

for all instances $I$. ◂

On the other hand, CA$_n$-expressions $E$ are translated into FO$_n$-formulas $\varphi_E$ recursively as follows:

- If $E = \mathsf{C}_p$, then

$$\varphi_E \;=\; R_p(x_1, \ldots, x_{ar(R_p)}) \;\wedge \bigwedge_{k \in \{ar(R_p)+1,\ldots,n\}} (x_k \approx x_k).$$

- If $E = \mathsf{d}_{ij}$, then

$$\varphi_E \;=\; (x_i \approx x_j) \;\wedge \bigwedge_{k\,\in\,\{1,\ldots,n\}\smallsetminus\{i,j\}} (x_k \approx x_k).$$

- If $E = F \bigcup G$, then $\varphi_E = \varphi_F \vee \varphi_G$, if $E = F \bigcap G$, then $\varphi_E = \varphi_F \wedge \varphi_G$, and if $E = \overline{F}$, then $\varphi_E = \neg\varphi_F$.
- If $E = \mathsf{c}_i(F)$, then $\varphi_E = (\exists x_i \varphi_F) \wedge (x_i \approx x_i)$.
- If $E = \mathsf{ɔ}_i(F)$, then $\varphi_E = (\forall x_i \varphi_F) \wedge (x_i \approx x_i)$.

The following result can also be extracted from [9, 10]. A proof is given in the Appendix.

**Theorem 2.6.** For every CA$_n$-expression $E$ the FO$_n$-formula $\varphi_E$ above is such that

$$\varphi_E^I \;=\; E(\mathsf{h}^n(I)),$$

for all instances $I$. ◂

# 3. Cylindric set algebra and cylindric star algebra

Since cylinders can be infinite, we want a finite mechanism to represent (at least some) infinite cylinders, and the mechanism to be closed under queries. Our representation mechanism comes in two variations, depending on whether negation is allowed or not. We first consider the positive (no negation) case.

## 3.1. Positive framework

**Star Cylinders.** We define an *$n$-dimensional (positive) star-cylinder* $\dot{C}$ to be a finite set of *$n$-ary star-tuples*, the latter being elements of $(\mathbb{D} \cup \{*\})^n \times \wp(\Theta_n)$, where $\Theta_n$ denotes the set of all *equalities* of the form $(i = j)$, with $i, j \in \{1, \ldots, n\}$, as well as the logical constant $\mathsf{false}$. Also, $\wp(.)$ denotes the powerset operation. Star-tuples will be denoted $\dot{t}, \dot{u}, \ldots$, where a star-tuple such as $\dot{t} = (a, *, c, *, *, \{(4 = 5)\})$ is meant to represent the set of all "ordinary" tuples $(a, x, c, y, y)$ where $x, y \in \mathbb{D}$. It will be convenient to assume that all our star-cylinders are in the following normal form.

**Definition 3.1.** An $n$-dimensional star-cylinder $\dot{C}$ is said to be in *normal form* if $\dot{t}(n+1) \not\vDash \mathsf{false}$, and $\dot{t}(n+1) \vDash (i = j)$ if and only if $(i = j) \in \dot{t}(n+1)$ and $\dot{t}(i) = \dot{t}(j)$, for all star-tuples $\dot{t} \in \dot{C}$ and $i, j \in \{1, \ldots, n\}$.

The symbol $\vDash$ above stands for standard logical implication. It is easily seen that maintaining star-cylinders in normal form can be done efficiently in polynomial time. We shall therefore assume without loss of generality that all star-cylinders and star-tuples are in normal form. We next define the notion of *dominance*, where a dominating star-tuple represents a superset of the ordinary tuples represented by the dominated star-tuple. First we define a relation $\leq \;\subseteq\; (\mathbb{D} \cup \{*\})^2$ by $a \leq a$, $* \leq *$, and $a \leq *$, for all $a \in \mathbb{D}$.

**Definition 3.2.** Let $\dot{t}$ and $\dot{u}$ be $n$-dimensional star-tuples. We say that $\dot{u}$ *dominates* $\dot{t}$, denoted $\dot{t} \leq \dot{u}$, if $\dot{t}(i) \leq \dot{u}(i)$ for all $i \in \{1, \ldots, n\}$, and $\dot{u}(n+1) \subseteq \dot{t}(n+1)$.

**Definition 3.3.** We extend the order $\leq$ to include "ordinary" $n$-ary tuples $t \in \mathbb{D}^n$ by identifying $(a_1, \ldots, a_n)$ with star-tuple $(a_1, \ldots, a_n, \theta)$, where $\theta$ contains $(i = j)$ iff $a_i = a_j$. Let $\dot{C}$ be an $n$-dimensional star-cylinder. We can now define the meaning of $\dot{C}$ to be the set $[[\dot{C}\,]]$ of all ordinary tuples it represents, where

$$[[\dot{C}\,]] = \{t \in \mathbb{D}^n : \; t \leq \dot{u} \text{ for some } \dot{u} \in \dot{C}\}.$$

We lift the order to $n$-dimensional star-cylinders $\dot{C}$ and $\dot{D}$, by stipulating that $\dot{C} \leq \dot{D}$, if for all star-tuples $\dot{t} \in \dot{C}$ there is a star-tuple $\dot{u} \in \dot{D}$, such that $\dot{t} \leq \dot{u}$.

**Lemma 3.4.** Let $\dot{C}$ and $\dot{D}$ be $n$-dimensional (positive) star-cylinders. Then $[[\dot{C}]] \subseteq [[\dot{D}]]$ iff $\dot{C} \leq \dot{D}$.

**Proof:**
We first show that $[[\{\dot{t}\}]] \subseteq [[\dot{D}]]$ if and only if there is a star-tuple $\dot{u} \in \dot{D}$, such that $\dot{t} \leq \dot{u}$.[5] For a proof, we note that if $\dot{t} \leq \dot{u}$ for some $\dot{u} \in \dot{D}$, then $[[\{\dot{t}\,\}]] \subseteq [[\dot{D}]]$. For the other direction, assume that $[[\{\dot{t}\,\}]] \subseteq [[\dot{D}]]$. Let $A \subseteq \mathbb{D}$ be the finite set of constants appearing in $\dot{t}$ or $\dot{D}$. Construct the tuple $t \in (A \cup \{*\})^n$, where $t(i) = \dot{t}(i)$ if $\dot{t}(i) \in A$, and $t(i) = a_i$ if $\dot{t}(i) = *$. Here $a_i$ is a unique value in the set $\mathbb{D} \smallsetminus A$. If $\dot{t}(n+1)$ contains an equality $(i = j)$ we choose $a_i = a_j$. Then $t \in [[\{\dot{t}\,\}]] \subseteq [[\dot{D}]]$, so there must be a tuple $\dot{u} \in \dot{D}$, such that $t \leq \dot{u}$. It remains to show that $\dot{t} \leq \dot{u}$. If $t(i) = a$ for some $a \in A$, then $\dot{t}(i) = a$, and since $t \leq \dot{u}$ it follows that $\dot{t}(i) \leq \dot{u}(i)$. If $t(i) = a_i \notin A$ then $\dot{t}(i) = *$, and therefore $t(i/b) \in [[\{\dot{t}\}]] \subseteq [[\dot{D}]]$, for any $b$ in the infinite set $\mathbb{D} \smallsetminus A$. Consequently it must be that $\dot{u}(i) = *$, and thus $\dot{t}(i) \leq \dot{u}(i)$. This is true for all $i \in \{1, \ldots, n\}$. Finally, if $(i = j) \in \dot{u}(n+1)$, we have two cases: If $t(i) \in A$ then $\dot{t}(i) = \dot{t}(j)$, and if $t(i) \notin A$ then $(i = j) \in \dot{t}(n+1)$. In summary, we have shown that $\dot{t} \leq \dot{u}$.

We now return to the proof of the claim of the lemma. The if-direction follows directly from definitions. For the only-if direction, assume that $[[\dot{C}]] \subseteq [[\dot{D}]]$. To see that $\dot{C} \leq \dot{D}$ let $\dot{t} \in \dot{C}$. Then $[[\{\dot{t}\,\}]] \subseteq [[\dot{C}]] \subseteq [[\dot{D}]]$. We have just shown above that this implies that there is a $\dot{u} \in \dot{D}$ such that $\dot{t} \leq \dot{u}$, meaning that $\dot{C} \leq \dot{D}$. □

### Positive Cylindric Star Algebra

Next we redefine the positive cylindric set algebra operators so that $[[\dot{C} \;\dot{\circ}\; \dot{D}]] = [[\dot{C}]] \circ [[\dot{D}]]$ or $[[\dot{\circ}(\dot{D})]] = \circ([[\dot{D}]])$, for each positive cylindric set algebra operator $\circ$, its redefinition $\dot{\circ}$, and star-cylinders $\dot{C}$ and $\dot{D}$. We first define the *meet* $\dot{t} \wedge \dot{u}$ of star-tuples $\dot{t}$ and $\dot{u}$ :

**Definition 3.5.** Let $\dot{t}$ and $\dot{u}$ be $n$-ary star-tuples. The $n$-ary star-tuple $\dot{t} \wedge \dot{u}$ is defined as follows: If $\dot{t}(j), \dot{u}(j) \in \mathbb{D}$ for some $j$ and $\dot{t}(j) \neq \dot{u}(j)$, then $\dot{t} \wedge \dot{u}\,(i) = *$ for $i \in \{1, \ldots, n\}$, and $\dot{t} \wedge \dot{u}\,(n+1) =$

[5] Note here the normal form requirement $\dot{t}(n+1) \nvDash \mathsf{false}$, since $\dot{t}(n+1) \vDash \mathsf{false}$ means that $[[\{\dot{t}\}]] = \emptyset$, while there is no star-tuple $\dot{u}$ , such that $\dot{t} \leq \dot{u}$ and $\dot{u}(n+1) \nvDash \mathsf{false}$.

$\{\mathsf{false}\}$.[6] Otherwise, for $i \in \{1, \ldots, n\}$

$$\dot{t} \wedge \dot{u}\,(i) = \begin{cases} \dot{t}(i) & \text{if } \dot{t}(i) \in \mathbb{D} \\ \dot{u}(i) & \text{if } \dot{u}(i) \in \mathbb{D} \\ * & \text{if } \dot{t}(i) = \dot{u}(i) = * \end{cases}$$

and

$$\dot{t} \wedge \dot{u}\,(n+1) \quad = \quad \dot{t}(n+1) \cup \dot{u}(n+1).$$

For an example, let $\dot{t} = (a, *, *, *, *, \{(3 = 4)\})$ and $\dot{u} = (*, b, *, *, *, \{(4 = 5)\})$. Then we have $\dot{t} \wedge \dot{u} = (a, b, *, *, *, \{(3 = 4), (4 = 5), (3 = 5)\})$. Note that[7] $\dot{t} \wedge \dot{u} \preceq \dot{t}$, and $\dot{t} \wedge \dot{u} \preceq \dot{u}$, and if for a star-tuple $\dot{v}$, we have $\dot{v} \preceq \dot{t}$ and $\dot{v} \preceq \dot{u}$, then $\dot{v} \preceq \dot{t} \wedge \dot{u}$.

**Definition 3.6.** The $n$-dimensional positive cylindric star-algebra consists of the following operators.

1. *Star-diagonal:* $\dot{d}_{ij} = \{(\overbrace{*, \ldots, *}^{n}, \{(i = j)\})\}$
2. *Star-union:* $\dot{C} \mathbin{\dot{\cup}} \dot{D} = \{\dot{t} \,:\, \dot{t} \in \dot{C} \text{ or } \dot{t} \in \dot{D}\}$
3. *Star-intersection:* $\dot{C} \mathbin{\dot{\cap}} \dot{D} = \{\dot{t} \wedge \dot{u} \,:\, \dot{t} \in \dot{C} \text{ and } \dot{u} \in \dot{D}\}$
4. *Outer cylindrification:* Let $i \in \{1, \ldots, n\}$, let $\dot{C}$ be an $n$-dimensional star-cylinder, and $\dot{t} \in \dot{C}$. Then

   $$\dot{\mathsf{c}}_i(\dot{t})(j) \quad = \quad \begin{cases} \dot{t}(j) & \textit{if } j \neq i \\ * & \textit{if } j = i \end{cases}$$

   for $j \in \{1, \ldots, n\}$, and

   $$\dot{\mathsf{c}}_i(\dot{t})(n+1) = \{(j = k) \in \dot{t}(n+1) \,:\, j, k \neq i\}.$$

   We then let $\dot{\mathsf{c}}_i(\dot{C}) = \{\dot{\mathsf{c}}_i(\dot{t}) \,:\, \dot{t} \in \dot{C}\}$.
5. *Inner cylindrification:* Let $\dot{C}$ be an $n$-dimensional cylinder and $i \in \{1, \ldots, n\}$. Then

   $$\dot{\mathsf{Ↄ}}_i(\dot{C}) = \{\dot{t} \in \dot{C} \,:\, \dot{t}(i) = *, \text{ and } (i = j) \notin \dot{t}(n+1) \text{ for any } j\}.$$

We illustrate the positive cylindric star-algebra with the following small example.

[6] Here $*$ can be replaced by any arbitrary constant $a$ in $\mathbb{D}$, but for consistency we prefer to use $*$.
[7] Assuming the normal form requirement $\dot{t} \wedge \dot{u}\ (n+1) \not\models \mathsf{false}$.

**Example 3.7.** Let $\dot{C}_1 = \{(a,*,*,*,*,\{(3=4)\})\}$, $\dot{C}_2 = \{(*,b,*,*,*,\{(4=5)\})\}$, and $\dot{C}_3 = \{(a,b,*,*,*,\{(4=5)\})\}$. Consider the expression $\dot{\mathsf{o}}_3((\dot{\mathsf{c}}_{1,4}(\dot{C}_1 \Cap \dot{C}_2)) \Cup \dot{C}_3)$.[8] Then we have the following.

$$\begin{array}{rcl} \dot{C}_1 \Cap \dot{C}_2 & = & \{(a,b,*,*,*,\{(3=4),(4=5),(3=5)\})\} \\ \dot{\mathsf{c}}_{1,4}(\dot{C}_1 \Cap \dot{C}_2) & = & \{(*,b,*,*,*,\{(3=5)\})\} \\ (\dot{\mathsf{c}}_{1,4}(\dot{C}_1 \Cap \dot{C}_2)) \Cup C_3 & = & \{(*,b,*,*,*,\{(3=5)\}),\ (a,b,*,*,*,\{(4=5)\})\} \\ \dot{\mathsf{o}}_3((\mathsf{c}_{1,4}(\dot{C}_1 \Cap \dot{C}_2)) \Cup C_3) & = & \{(a,b,*,*,*,\{(4=5)\})\} \end{array}$$

Next we show that the cylindric star-algebra has the promised property.

**Theorem 3.8.** Let $\dot{C}$ and $\dot{D}$ be $n$-dimensional star-cylinders and $\dot{d}_{ij}$ an $n$-dimensional star-diagonal. Then the following statements hold.

1. $[[\dot{d}_{ij}]] = d_{ij}$.
2. $[[\dot{C} \Cup \dot{D}]] = [[\dot{C}]] \cup [[\dot{D}]]$.
3. $[[\dot{C} \Cap \dot{D}]] = [[\dot{C}]] \cap [[\dot{D}]]$.
4. $[[\dot{\mathsf{c}}_i(\dot{C})]] = \mathsf{c}_i([[\dot{C}]])$,
5. $[[\dot{\mathsf{o}}_i(\dot{C})]] = \mathsf{o}_i([[\dot{C}]])$,

**Proof:**

1. $t \in [[\dot{d}_{ij}]]$ iff $t \preceq (*,\ldots,*,(i=j))$ iff $t \in \{t \in \mathbb{D}^n : t(i) = t(j)\}$ iff $t \in d_{ij}$.
2. $t \in [[\dot{C} \Cup \dot{D}]]$ iff $\exists \dot{u} \in \dot{C} : t \preceq \dot{u}$ or $\exists \dot{v} \in \dot{D} : t \preceq \dot{v}$ iff $t \in [[\dot{C}]]$ or $t \in [[\dot{D}]]$ iff $t \in [[\dot{C}]] \cup [[\dot{D}]]$.
3. Let $t \in [[\dot{C} \Cap \dot{D}]]$. Then there is a star-tuple $\dot{t} \in \dot{C} \Cap \dot{D}$ such that $t \preceq \dot{t}$, which again means that there are star-tuples $\dot{u} \in \dot{C}$ and $\dot{v} \in \dot{D}$, such that $\dot{t} = \dot{u} \wedge \dot{v}$. As a consequence $t \preceq \dot{u}$ and $t \preceq \dot{v}$, which implies $t \in [[\dot{C}]]$ and $t \in [[\dot{D}]]$, that is, $t \in [[\dot{C}]] \cap [[\dot{D}]]$. The proof for the other direction is similar.
4. Let $t \in [[\dot{\mathsf{c}}_i(\dot{C})]]$. Then there is a star-tuple $\dot{t} \in \dot{\mathsf{c}}_i(\dot{C})$ such that $t \preceq \dot{t}$. This in turn means that there is a star-tuple $\dot{u} \in \dot{C}$ such that either $\dot{u} = \dot{t}(i/a)$ for some $a \in \mathbb{D}$, or $\dot{u}(i) = *$ and $\dot{u} = \dot{t}$, except possibly $\dot{u}(n+1) \vDash \theta$ where $\theta$ is a set of equalities involving column $i$, and $\dot{t}(n+1)$ does not have any conditions on $i$.

   *Case 1.* $\dot{u} = \dot{t}(i/a)$ for some $a \in \mathbb{D}$. Then $\dot{t}(i/a) \in \dot{C}$ which means that there is a tuple $u \in [[\dot{C}]]$ such that $u \preceq \dot{t}(i/a)$. Since $[[\dot{C}]] \subseteq \mathsf{c}_i([[\dot{C}]])$, it follows that $u \in \mathsf{c}_i([[\dot{C}]])$. Suppose $u \neq t$. Then $u(j) \neq t(j)$ for some $j \in \{1\ldots,n\}$.

   If $j = i$, then $t = u(j/t(j)) \in \mathsf{c}_j([[\dot{C}]]) = \mathsf{c}_i([[\dot{C}]])$.

[8] $\dot{\mathsf{c}}_{i,j}(\dot{C})$ is a abbreviation of $\dot{\mathsf{c}}_i(\dot{\mathsf{c}}_j(\dot{C}))$.

If $j \neq i$ and $\dot{t}(j) = *$ it means that $\dot{u}(j) = *$, and thus $t = u(j/t(j)) \in [\![\dot{C}]\!]$, which in turn implies that $t \in \mathsf{c}_i([\![\dot{C}]\!])$. Otherwise, if $\dot{t}(j) \neq *$, then $\dot{t}(j) \in \mathbb{D}$, which means that $\dot{u}(j) \in \mathbb{D}$, and $u(j) = t(j)$ after all.

*Case 2.* $\dot{u}(i) = *$ and (possibly) $\dot{u}(n+1)$ contains a set of equalities say $\theta$, involving column $i$, and $\dot{t}(n+1)$ does not have any conditions on $i$.

Suppose first that $t \vDash \theta$. Then $t \leq \dot{u}$, and consequently $t \in [\![\dot{C}]\!] \subseteq \mathsf{c}_i([\![\dot{C}]\!])$.

Suppose then that $t \nvDash \theta$. If $t$ violates an equality $(i = j) \in \theta$ it must be that $\dot{t}(j) = \dot{u}(j) = *$, and $\dot{t}$ and $\dot{u}$ have the same conditions on column $j$. Let $u$ be a tuple such that $u \leq \dot{u}$. Then $t(i/u(i)) \in [\![\dot{C}]\!]$, and hence $t \in \mathsf{c}_i([\![\dot{C}]\!])$.

For the other direction, let $t \in \mathsf{c}_i([\![\dot{C}]\!])$. Then there is a tuple $u \in [\![\dot{C}]\!]$, such that $t(i/u(i)) = u$. Hence there is a star-tuple $\dot{u} \in \dot{C}$, such that $u \leq \dot{u}$ and $t(i/u(i)) \leq \dot{u}$. If $t \nleq \dot{u}$ it is because $t(i)$ violates some condition in $\dot{u}(n+1)$. Since all conditions involving column $i$ are deleted in $\dot{\mathsf{c}}_i(\dot{C})$, it follows that $\dot{\mathsf{c}}_i(\dot{C})$ must contain a star-tuple $\dot{v}$ obtained by outer cylindrification of $\dot{u}$. Then clearly $t \leq \dot{v}$ and $t \vDash \dot{v}(n+1)$. Consequently $t \in [\![\dot{\mathsf{c}}_i(\dot{C})]\!]$.

5. Let $t \in [\![\dot{\mathsf{ɔ}}_i(\dot{C})]\!]$. Then there is a star-tuple $\dot{t} \in \dot{\mathsf{ɔ}}_i(\dot{C})$, such that $t \leq \dot{t}$. Clearly, $\dot{t} \in \dot{\mathsf{ɔ}}_i(\dot{C})$ means that $\dot{t} \in \dot{C}$ where by definition $\dot{t}(i) = *$, and $(i = j) \notin \dot{t}(n+1)$ for any $j$. As a consequence $t(i/a) \leq \dot{t}$ for all $a \in \mathbb{D}$. This implies that $t(i/a) \in [\![\dot{C}]\!]$ for all $a \in \mathbb{D}$, and thus $t \in \mathsf{ɔ}_i([\![\dot{C}]\!])$.

   For the other direction, let $t \in \mathsf{ɔ}_i([\![\dot{C}]\!]) \subseteq [\![\dot{C}]\!]$. This means that $t(i/a) \in [\![\dot{C}]\!]$ for all $a \in \mathbb{D}$. That is, there exists a star-tuple $\dot{t} \in \dot{C}$, such that $t \leq \dot{t}$. Also, $t(i/a) \leq \dot{t}$ for all $a \in \mathbb{D}$, since there otherwise has to be an infinite number of star-tuples in in the finite star-cylinder $\dot{C}$. Thus it must be that $\dot{t}(i) = *$, and $(i = j) \notin \dot{t}(n+1)$ for any $j$. Consequently, $\dot{t} \in \dot{\mathsf{ɔ}}_i(\dot{C})$ and $t \in [\![\dot{\mathsf{ɔ}}_i(\dot{C})]\!]$.

□

In order to show the equivalence of positive cylindric star-algebra and positive cylindric set algebra we need the concept of algebra expressions.

**Definition 3.9.** Let $\dot{\mathbf{C}} = (\dot{C}_1, \ldots, \dot{C}_m, \dot{d}_{ij})_{i,j}$ be a sequence of $n$-dimensional star-cylinders and star-diagonals. We define the set of *positive cylindric star algebra expressions* $\mathrm{SCA}_n^+$ and values of expressions as in Definition 2.3, noting that $\dot{\mathsf{C}}_p(\dot{\mathbf{C}}) = \dot{C}_p$, and $\dot{\mathsf{d}}_{ij}(\dot{\mathbf{C}}) = \dot{d}_{ij}$.

In the following, $\mathrm{CA}_n^+$ denotes the set of all $n$-dimensional positive cylindric algebra expressions. We now have from Theorem 3.8

**Corollary 3.10.** For every $\mathrm{SCA}_n^+$-expression $\dot{E}$ and the corresponding $\mathrm{CA}_n^+$ expression $E$, it holds that

$$[\![\dot{E}(\dot{\mathbf{C}})]\!] = E([\![\dot{\mathbf{C}}]\!])$$

for every sequence of $n$-dimensional star-cylinders and star-diagonals $\dot{\mathbf{C}}$.

### 3.2. Adding negation

From here on we also allow conditions of the form $(i \neq j)$, $(i \neq a)$, for $a \in \mathbb{D}$ in star-cylinders, which then will be called *extended star-cylinders*. Conditions of the form $(i = j)$, $(i \neq j)$ or $(i \neq a)$ will be called *literals*[9], usually denoted $\ell$. In other words, in an extended $n$-dimensional star-cylinder each (extended) star-tuple $\dot{t}$ has a (finite) set of literals in position $n+1$.

**Example 3.11.** In Example 1.2 we were interested in the negative information as well as positive information. The instance from Example 1.2 can be formally represented as the extended star-cylinder below.

| $H^-$ | | |
|---|---|---|
| Alice | Volleyball | $\{\}$ |
| Bob | $*$ | $\{(2 \neq \text{Basketball})\}$ |
| Chris | $*$ | $\{\}$ |

We next extend Definitions 3.1, 3.2, 3.3, 3.5, and 3.6 to apply to extended star-cylinders. Lemma 3.4 will be replaced by Lemma 3.15 below.

**Definition 3.12.** *(Replaces Definition 3.1).* An extended $n$-dimensional star-cylinder $\dot{C}$ is said to be in *normal form*, if $\dot{t}(n+1) \not\models \mathsf{false}$, and $\dot{t}(n+1) \models \ell$ if and only if $\ell \in \dot{t}(n+1)$, and

1. $(i = j) \in \dot{t}(n+1)$ if and only if $t(i) = t(j)$,
2. $(i \neq j) \in \dot{t}(n+1)$ if and only if $t(i) \neq t(j)$, or $(i \neq j) \in \dot{t}(n+1)$ and $t(i) = t(j) = *$,
3. $(i \neq a) \in \dot{t}(n+1)$ implies $t(i) \neq a$,

for all star-tuples $\dot{t} \in \dot{C}$ and $i, j \in \{1, \ldots, n\}$.

In the proof of Theorem 6.2 in Section 6 we show that maintaining extended star-cylinders in normal for can be done in polynomial time. We therefore assume in the sequel that all extended star-cylinders and -tuples are in normal form. Keeping the extended notion of normal form in mind, it is easy to see that Definition 3.2 of dominance $\dot{t} \leq \dot{u}$ suits extended star-tuples $\dot{t}$ and $\dot{u}$ as well. Definition 3.3 remains unchanged, provided we identify an "ordinary" tuple $(a_1, \ldots, a_n)$ with the extended star-tuple $(a_1, \ldots, a_n, \theta)$, where $(i = j) \in \theta$ iff $a_i = a_j$ and $(i \neq j) \in \theta$ iff $a_i \neq a_j$. Definition 3.5 also applies as such to extended star-tuples. For the outer cylindrification in Definition 3.6 we now stipulate that $\dot{\mathsf{c}}_i(\dot{t})(n+1)$ contains all and only those literals from $\dot{t}(n+1)$ that do not involve dimension $i$. It is an easy exercise to verify that the proofs of parts 1 – 4 of Theorem 3.8 remain valid in the presence of literals. Finally, inner cylindrification will be redefined below, along with the definition of the complement operator. Before that we introduce the notion of a *sieve*-cylinder.

[9] $\mathsf{false}$ is also a literal

**Definition 3.13.** Let $\dot{\mathbf{C}}$ be a sequence of $n$-dimensional extended star-cylinders and $A$ be the set of constants appearing therein. For $t \in (A \cup \{*\})^n$, define $S_t = \{i : t(i) = *\}$ and $SS_t = \{(i,j) : t(i) = t(j) = *\}$. For each tuple $t \in (A \cup \{*\})^n$ and each subset $SS_t^+$ of $SS_t$, form the star-tuple $\dot{t}$ with $\dot{t}(i) = t(i)$ for $i \in \{1, \dots, n\}$, and $\dot{t}(n+1) =$

$$\bigcup_{i \in S_t} \{(i \neq a) : a \in A\} \bigcup_{(i,j) \in SS_t^+} \{(i = j)\} \bigcup_{(i,j) \in SS_t \setminus SS_t^+} \{(i \neq j)\}.$$

$\dot{A}$ is the extended star-cylinder of all such star-tuples $\dot{t}$, and it is called the *sieve* of $\dot{\mathbf{C}}$.

The sieve $\dot{A}$ has some useful properties stated in the next two lemmas. These properties allow us to test containmnent $[[\dot{C}]] \subseteq [[\dot{D}]]$ and to define negation and inner cylindrification on a tuple-by-tuple basis using the partial order $\preceq$.

**Lemma 3.14.** Let $\dot{C}$ be an $n$-dimensional star-cylinder and $\dot{A} = \{\dot{t}_1, \dots, \dot{t}_m\}$ its sieve. Then

1. $[[\dot{A}]] = \mathbb{D}^n$ and $\{[[\{\dot{t}_1\}]], \dots, [[\{\dot{t}_m\}]]\}$ is a partition of $[[\dot{A}]]$.
2. If $\dot{t} \wedge \dot{u} \in \dot{C} \Cap \dot{A}$ and $\dot{t} \wedge \dot{u} \neq \dot{t}_\varnothing$, then $\dot{t} \wedge \dot{u} = \dot{u}$.

**Proof:**
To see that $[[\dot{A}]] = \mathbb{D}^n$, let $t$ be an arbitrary tuple in $\mathbb{D}^n$. By construction, there are star-tuples $\dot{t} \in \dot{A}$ such that $\dot{t}(i) = t(i)$ if $t(i) \in A$, and $\dot{t}(i) = *$ if $t(i) \in \mathbb{D} \setminus A$. Since there is the subset $SS_t^+ = \{(i,j) : t(i) = t(j), \text{ and } t(i) \in \mathbb{D} \setminus A\}$ we see that for one of these $\dot{t}$-tuples it holds that $t \preceq \dot{t}$. The fact that $[[\{\dot{t}_i\}]] \cap [[\{\dot{t}_j\}]] = \varnothing$ whenever $i \neq j$ follows from the fact that if there were a tuple $t$ in the intersection, it would have to agree with $\dot{t}_i$ and $\dot{t}_j$ on all columns with values in $A$. But the $SS_t^+$ set used for $\dot{t}_i$ would be different than the one used for $\dot{t}_j$, which means that we cannot have both $t \preceq \dot{t}_i$ and $t \preceq \dot{t}_j$.

For part 2, let $\dot{t} \wedge \dot{u} \in \dot{C} \Cap \dot{A}$ and $\dot{t} \wedge \dot{u} \neq \dot{t}_\varnothing$. We claim that $\dot{u} \preceq \dot{t}$, which would imply $\dot{t} \wedge \dot{u} = \dot{u}$. Since $\dot{t} \wedge \dot{u} \neq \dot{t}_\varnothing$ there is a tuple $t \in [[\{\dot{t} \wedge \dot{u}\}]]$. For each $i \in \{1, \dots, n\}$, consider $\dot{u}(i)$. If $\dot{u}(i) = a \in A$, then $t(i) = a$, which means that $\dot{t}(i) = a$ or $\dot{t}(i) = *$. Consequently $\dot{u}(i) \preceq \dot{t}(i)$. If $\dot{u}(i) = *$ then $t(i) \in \mathbb{D} \setminus A$, since $(i \neq a) \in \dot{u}(n+1)$ for all $a \in \mathbb{D} \setminus A$. Since $t(i) \preceq \dot{t}(i)$, and $\dot{t}(i) \in A \cup \{*\}$, it follows that $\dot{t}(i) = *$. Then let $(i = j) \in \dot{t}(n+1)$. Since $\dot{t} \wedge \dot{u}\,(n+1)$ is satisfiable, and $\dot{u}(n+1)$ contains either $(i = j)$ or $(i \neq j)$, it follows that $(i = j) \in \dot{u}(n+1)$. We have now shown that $\dot{u} \preceq \dot{t}$. □

**Lemma 3.15.** Let $\dot{C}$ and $\dot{D}$ be $n$-dimensional extended star-cylinders and $\dot{A}$ their (common) sieve. Then

$$[[\dot{C}]] \subseteq [[\dot{D}]] \text{ iff } \dot{C} \Cap \dot{A} \preceq \dot{D} \Cap \dot{A}.$$

**Proof:**
For the *if*-direction, let $t \in [[\dot{C}]] = [[\dot{C} \Cap \dot{A}]]$. Then there is a star-tuple $\dot{t} \in \dot{C} \Cap \dot{A}$, such that $t \preceq \dot{t}$. Since $\dot{C} \Cap \dot{A} \preceq \dot{D} \Cap \dot{A}$ there is a star tuple $\dot{u} \in \dot{D} \Cap \dot{A}$ such that $\dot{t} \preceq \dot{u}$. Thus $t \in [[\dot{D} \Cap \dot{A}]] = [[\dot{D}]]$.

For the only-if direction, let $\dot{t}_1 \wedge \dot{u}_1 \in \dot{C} \Cap \dot{A}$, and $t \preceq \dot{t}_1$ and $t \preceq \dot{u}_1$. Then $t \in [[\dot{C}]] \subseteq [[\dot{D}]] = [[\dot{D} \Cap \dot{A}]]$, so there are star-tuples $\dot{t}_2 \in \dot{D}$ and $\dot{u}_2 \in \dot{A}$ such that $t \preceq \dot{t}_2$ and $t \preceq \dot{u}_2$. From Lemma 3.14 it follows that $\dot{u}_1 = \dot{u}_2$, and thus $\dot{t}_1 \wedge \dot{u}_1 = \dot{u}_1 = \dot{u}_2 = \dot{t}_2 \wedge \dot{u}_2$. Consequently $\dot{t}_1 \wedge \dot{u}_1 \preceq \dot{t}_2 \wedge \dot{u}_2$. □

We can now define the desired operations.

**Definition 3.16.** Let $\dot{A}$ be the sieve of $\dot{\mathbf{C}}$ and $\dot{C}$ be an extended star-cylinder in $\dot{\mathbf{C}}$. Then

1. $\dot{\neg}\dot{C} = \{\dot{t} \in \dot{A} : \{\dot{t}\} \dot{\cap} \dot{C} = \{\dot{t}_\varnothing\}\}$. and
2. $\hat{\mathfrak{I}}_i(\dot{C}) = \{\dot{t} \in \dot{C} \dot{\cap} \dot{A} : (\dot{\mathsf{c}}_i(\{\dot{t}\}) \dot{\cap} \dot{A}) \leq (\dot{C} \dot{\cap} \dot{A})\}$.

**Example 3.17.** Let $\dot{C} = \{(a, *, \{\})\}$. Then $\dot{A}$ is shown in the extended star-cylinder below, and $\dot{\neg}\dot{C}$ consists of the first, second, and fourth tuples of $\dot{A}$.

| $\dot{A}$ | | |
|---|---|---|
| $*$ | $*$ | $\{(1 \neq a), (2 \neq a), (1 = 2)\}$ |
| $*$ | $*$ | $\{(1 \neq a), (2 \neq a), (1 \neq 2)\}$ |
| $a$ | $*$ | $\{(2 \neq a)\}$ |
| $*$ | $a$ | $\{(1 \neq a)\}$ |
| $a$ | $a$ | $\{\}$ |

Now, let $\dot{C} = \{(a, *, \{(2 \neq a)\}), (a, a, \{\})\}$. Then $\dot{A}$ is as above, and $\hat{\mathfrak{I}}_2(\dot{C}) = \dot{C}$ as the reader easily can verify.

We can now verify that the new operators work as expected.

**Theorem 3.18.** Let $\dot{C}$ be an extended star-cylinder. Then

1. $[\![\dot{\neg}\dot{C}]\!] = \overline{[\![\dot{C}]\!]}$
2. $[\![\hat{\mathfrak{I}}_i(\dot{C})]\!] = \mathfrak{I}_i([\![\dot{C}]\!])$.

**Proof:**
For part 1, it is easy to see that $[\![\dot{\neg}\dot{C}]\!] \cap [\![\dot{C}]\!] = \varnothing$ which implies $[\![\dot{\neg}\dot{C}]\!] \subseteq \overline{[\![\dot{C}]\!]}$. For a proof of the other direction of part 1, for each tuple $t \in \overline{[\![\dot{C}]\!]}$, we construct the star-tuple $\dot{t}$, where $\dot{t}(i) = t(i)$ if $t(i) \in A$, and $\dot{t}(i) = *$ if $t(i) \notin A$. We then choose a subset $SS_t^+$ of $SS_t$ where $(i,j) \in SS^+$ if and only if $t(i) = t(j)$. We insert in $\dot{t}(n+1)$ the condition $(i = j)$ for each $(i,j) \in SS_t^+$, and $(i \neq j)$ for each $(i,j) \in SS_t \setminus SS_t^+$, Then clearly $t \in [\![\{\dot{t}\}]\!]$ and $\dot{t} \in \dot{A}$. It remains to show that $\dot{t} \in \dot{\neg}\dot{C}$. Towards a contradiction, suppose that there is a star-tuple $\dot{u} \in \dot{C}$ such that $\dot{t} \wedge \dot{u} \neq \dot{t}_\varnothing$. In other words, $\dot{t}(n+1) \cup \dot{u}(n+1)$ is satisfiable. Thus, whenever $\dot{t}(i) \in \mathbb{D}$, we must have $\dot{u}(i) = \dot{t}(i) = t(i) \in A$. Furthermore, for each $(i,j) \in SS_t$ there is a literal involving $i$ and $j$ in $\dot{t}(n+1)$. Therefore $\dot{u}(n+1)$ can consist of only a subset of these literals. It follows that $t \leq \dot{t} \leq \dot{u} \in \dot{C}$, meaning that $t \in [\![\dot{C}]\!]$, contradicting our initial assumption.

For a proof of part 2 of the theorem, let $t \in [\![\hat{\mathfrak{I}}_i(\dot{C})]\!]$. Then $t \in [\![\{\dot{t} \in \dot{A} : (\dot{\mathsf{c}}_i(\{\dot{t}\}) \dot{\cap} \dot{A}) \leq (\dot{C} \dot{\cap} \dot{A})\}]\!]$. Therefore there is a star tuple $\dot{t} \in \dot{A}$ such that, $t \leq \dot{t}$ and $(\dot{\mathsf{c}}_i(\{\dot{t}\}) \dot{\cap} \dot{A}) \leq (\dot{C} \dot{\cap} \dot{A})$. Lemma 3.15 then gives us $[\![\dot{\mathsf{c}}_i(\{\dot{t}\})]\!] \subseteq [\![\dot{C}]\!]$, and Theorem 3.8, part 4 (which still holds for extended star-cylinders) tell us that $[\![\dot{\mathsf{c}}_i(\{\dot{t}\})]\!] = \mathsf{c}_i([\![\{\dot{t}\}]\!])$ which implies $[\![\dot{\mathsf{c}}_i(\{\dot{t}\})]\!] \subseteq [\![\dot{C}]\!]$. By the definition of inner cylindrification in CA, the last containment implies that $[\![\{\dot{t}\}]\!] \subseteq \mathsf{\reflectbox{c}}_i([\![\dot{C}]\!])$. Consequently $t \in \mathsf{\reflectbox{c}}_i([\![\dot{C}]\!])$.

For the other direction, let $t \in \mathsf{\reflectbox{c}}_i([\![\dot{C}]\!])$, which implies $\mathsf{c}_i(\{t\}) \subseteq [\![\dot{C}]\!]$. Then there is a star-tuple $\dot{t} \in \dot{C} \Cap \dot{A}$, such that $\mathsf{c}_i(\{t\}) \subseteq \mathsf{c}_i([\![\{\dot{t}\}]\!]) \subseteq [\![\dot{C}]\!]$. Consequently, $[\![\dot{\mathsf{c}}_i(\dot{t})]\!] \subseteq [\![\dot{C}]\!]$, which by Lemma 3.15 proves that $(\dot{\mathsf{c}}_i(\{\dot{t}\}) \Cap \dot{A}) \preceq (\dot{C} \Cap \dot{A})$. Moreover, the first part of Lemma 3.15 implies that $t \in [\![\{\dot{t} \in \dot{A} : (\dot{\mathsf{c}}_i(\{\dot{t}\}) \Cap \dot{A}) \preceq (\dot{C} \Cap \dot{A})\}]\!]$. □

We can thus conclude

**Corollary 3.19.** For every SCA$_n$-expression $\dot{E}$ and the corresponding CA$_n$-expression $E$, it holds that

$$[\![\dot{E}(\dot{\mathbf{C}})]\!] = E([\![\dot{\mathbf{C}}]\!])$$

for every sequence of $n$-dimensional extended star-cylinders and star-diagonals $\dot{\mathbf{C}}$.

## 4. Stored databases with universal nulls (u-databases)

We now show how to use the cylindric star-algebra to evaluate FO-queries on stored databases containing universal nulls. Let $k$ be a positive integer. Then a $k$-ary star-relation $\dot{R}$ is a finite set of star-tuples of arity $k$. In other words, a $k$-ary star-relation is a star-cylinder of dimension $k$. A sequence $\dot{\mathbf{R}}$ of star-relations (over schema $\mathbf{R}$) is called a *stored database*. Examples 1.1 and 3.11 show stored databases. Everything that is defined for star-cylinders applies to $k$-ary star-relations. The exception is that no operators from the cylindric star-algebra are applied to star-relations. To do that, we first need to expand the stored database $\dot{\mathbf{R}}$.

**Definition 4.1.** Let $\dot{t}$ be a $k$-ary star-tuple, and $n \geq k$. Then $\dot{\mathsf{h}}^n(\dot{t})$, the *$n$-expansion* of $\dot{t}$, is the $n$-ary star-tuple $\dot{u}$, where

$$\dot{u}(i) = \begin{cases} \dot{t}(i) & \textit{if } i \in \{1, \dots, k\} \\ * & \textit{if } i \in \{k+1, \dots, n\} \\ \dot{t}(k+1) & \textit{if } i = n+1, \end{cases}$$

For a stored relation $\dot{R}$ and stored database $\dot{\mathbf{R}}$ we have

$$\begin{aligned} \dot{\mathsf{h}}^n(\dot{R}) &= \{\dot{\mathsf{h}}^n(\dot{t}) : \dot{t} \in \dot{R}\} \\ \dot{\mathsf{h}}^n(\dot{\mathbf{R}}) &= (\dot{\mathsf{h}}^n(\dot{R}_1), \dots, \dot{\mathsf{h}}^n(\dot{R}_m), \dot{\mathsf{d}}_{ij})_{i,j}. \end{aligned}$$

In other words, $\dot{\mathsf{h}}^n(\dot{\mathbf{R}})$ is the sequence of star-cylinders obtained by moving the conditions in column $k+1$ to column $n+1$, and filling columns $k+1, \dots, n$ with "*"'s in each $k$-ary relation. Examples 1.1 and 1.3 illustrate the expansion of star-relations.

On the other hand, a $k$-ary star-relation $\dot{R}$ can also be viewed as a finite representative of the infinite relation $[\![\dot{R}]\!] = \{t \in \mathbb{D}^k : t \preceq \dot{t}$ for some $\dot{t} \in \dot{R}\}$, and the stored database $\dot{\mathbf{R}}$ a finite representative of the infinite instance $I(\dot{\mathbf{R}})$, as in the following definition.

**Definition 4.2.** Let $\dot{\mathbf{R}} = (\dot{R}_1, \ldots, \dot{R}_m)$ be a stored database. Then the instance defined by $\dot{\mathbf{R}}$ is

$$I(\dot{\mathbf{R}}) = ([\![\dot{R}_1]\!], \ldots, [\![\dot{R}_m]\!], \{(a,a) : a \in \mathbb{D}\}).$$

The instance and expansion of $\dot{\mathbf{R}}$ are related as follows.

**Lemma 4.3.** $[\![\dot{\mathsf{h}}^n(\dot{\mathbf{R}})]\!] \ = \ \mathsf{h}^n(I(\dot{\mathbf{R}}))$.

**Proof:**
Follows directly from the definitions of $\mathsf{h}^n$, $\dot{\mathsf{h}}^n$ and $[\![\,]\!]$. □

We are now ready for our main result.

**Theorem 4.4.** For every FO$_n$-formula $\varphi$ there is an (extended) SCA$_n$ expression $\dot{E}_\varphi$, such that for every stored database $\dot{\mathbf{R}}$

$$\mathsf{h}^n(\varphi^{I(\dot{\mathbf{R}})}) \ = \ [\![\dot{E}_\varphi(\dot{\mathsf{h}}^n(\dot{\mathbf{R}}))]\!].$$

**Proof:**
$\mathsf{h}^n(\varphi^{I(\dot{\mathbf{R}})}) \ = \ E_\varphi(\mathsf{h}^n(I(\dot{\mathbf{R}})) \ = \ E_\varphi([\![\dot{\mathsf{h}}^n(\dot{\mathbf{R}})]\!]) \ = \ [\![\dot{E}_\varphi(\dot{\mathsf{h}}^n(\dot{\mathbf{R}}))]\!]$. The first equality follows from Theorem 2.5, the second from Lemma 4.3, and the third from Corollaries 3.10 and 3.19. □

# 5. Adding existential nulls

Let $\mathbb{N} = \{\bot_1, \bot_2, \ldots\}$ be a countable infinite set of *existential nulls*. An instance $I$ where the relations are over $\mathbb{D} \cup \mathbb{N}$, is in the literature variably called a naive table [1, 3] or a generalized instance [2]. In either case, such an instance is taken to represent an *incomplete instance*, i.e. a (possibly) infinite set of instances. In this paper we follow the model-theoretic approach of [2]. The elements in $\mathbb{D}$ represent known objects, whereas elements in $\mathbb{N}$ represent generic objects. Each generic object could turn out to be equal to one of the known objects, to another generic object, or represent an object different from all other objects. We extend our notation to include $univ(I)$, the *universe* of instance $I$. So far we have assumed that $univ(I) = \mathbb{D}$, but in this section we allow instances whose universe is any set between $\mathbb{D}$ and $\mathbb{D} \cup \mathbb{N}$. We are lead to the following definitions.

**Definition 5.1.** Let $h$ be a mapping on $\mathbb{D} \cup \mathbb{N}$ that is identity on $\mathbb{D}$, and let $I$ and $J$ be instances (over $\mathscr{R}$), such that $h(univ(I)) = univ(J)$. We say that $h$ is a *possible world homomorphism* from $I$ to $J$, if $h(R_p^I) \subseteq R_p^J$ for all $p$, and $h(\approx^I) = \approx^J$. This is denoted $I \to_h J$.

**Definition 5.2.** Let $I$ be an instance with $\mathbb{D} \subseteq univ(I) \subseteq \mathbb{D} \cup \mathbb{N}$. Then the set of instances represented by $I$ is

$$Rep(I) \ = \ \{J \ : \ \exists\, h \ s.t.\, I \to_h J\}.$$

We can now formulate the (standard) notion of a *certain answer* to a query.[10] By FO$^+$ below we mean the set of all FO-formulas not using negation.

[10] Here we of course assume that valuations have range $univ(J)$, and that other details are adjusted accordingly.

**Definition 5.3.** Let $I$ be an incomplete instance and $\varphi$ an FO$^+$-formula. The certain answer to $\varphi$ on $I$ is

$$Cert(\varphi, I) \;=\; \bigcap_{J \in Rep(I)} \varphi^J.$$

**Back to cylinders.** We now extend positive $n$-dimensional cylinders to be subsets of $(\mathbb{D} \cup \mathbb{N})^n$, and use the notation $univ(\mathbf{C})$ and $univ(C)$ with the obvious meanings. This also applies to the notation $\mathbf{C} \rightarrow_h \mathbf{D}$, and $Rep(\mathbf{C})$. Cylinders $C$ with $\mathbb{D} \subseteq univ(C) \subseteq \mathbb{D} \cup \mathbb{N}$ will be called *naive cylinders*. The operators of the positive cylindric set algebra CA$^+$ remain the same, except $\mathbb{D}$ is substituted with $univ(\mathbf{C})$ or $univ(C)$, i.e. we use naive evaluation. For instance, the outer cylindrification now becomes

$$\mathsf{c}_i(C) = \{t \in univ(C)^n \;:\; t(i/x) \in C, \text{ for some } x \in univ(C)\}.$$

The crucial property of the positive cylindric set algebra is the following.

**Theorem 5.4.** Let $E$ be an expression in CA$_n^+$, and $\mathbf{C}$ and $\mathbf{D}$ sequences of $n$-dimensional naive cylinders and diagonals. If $\mathbf{C} \rightarrow_h \mathbf{D}$ for some possible world homomorphism $h$, then $E(\mathbf{C}) \rightarrow_h E(\mathbf{D})$.

**Proof:**
Suppose $\mathbf{C} \rightarrow_h \mathbf{D}$. We show by induction on the structure of $E$ that $E(\mathbf{C}) \rightarrow_h E(\mathbf{D})$.

- For $E = \mathsf{C}_i$ and $E = \mathsf{d}_{ij}$ the claim follows directly from the definition of a possible world homomorphism.

- Let $t \in h(F \bigcup G\,(\mathbf{C})) \;=\; h(F(\mathbf{C}) \cup G(\mathbf{C})) \;=\; h(F(\mathbf{C})) \cup h(G(\mathbf{C}))$. Then there is a tuple $s$ in $F(\mathbf{C})$ or in $G(\mathbf{C})$ such that $t = h(s)$. If $s$ is in, say, $F(\mathbf{C})$, then, since $F(\mathbf{C}) \rightarrow_h F(\mathbf{D})$ and $F(\mathbf{D}) \subseteq F \bigcup G\,(\mathbf{D})$, it follows that $t = h(s) \in F \bigcup G\,(\mathbf{D})$.

- Let $t \in h(F \bigcap G\;(\mathbf{C})) \;=\; h(F(\mathbf{C}) \cap G(\mathbf{C}))$. Then there is a tuple $s$ in $F(\mathbf{C})$ and a tuple $s'$ in $G(\mathbf{C})$ such that $t = h(s) = h(s')$. Thus $h(s) \in h(F(\mathbf{C})) \subseteq F(\mathbf{D})$, and $h(s') \in h(G(\mathbf{C})) \subseteq G(\mathbf{D})$. Consequently $t = h(s) = h(s') \in F(\mathbf{D}) \cap G(\mathbf{D}) \;=\; F \bigcap G\,(\mathbf{D})$.

- Let $t \in h(\mathsf{c}_i(F(\mathbf{C})))$. Then there is an $s \in \mathsf{c}_i(F(\mathbf{C}))$, such that $t = h(s)$. Furthermore, $s(i/x) \in F(\mathbf{C})$ for some $x \in univ(\mathbf{C})$. Then $h(s(i/x)) = h(s)(i/h(x)) \in h(F(\mathbf{C}))$, for $h(x) \in h(univ(\mathbf{C})) = univ(\mathbf{D})$, This means that $t = h(s) \in \mathsf{c}_i(F(\mathbf{D}))$.

- Let $t \in h(\mathsf{ɔ}_i(F(\mathbf{C})))$. Then there is an $s \in \mathsf{ɔ}_i(F(\mathbf{C}))$, such that $t = h(s)$. Furthermore, $s(i/x) \in F(\mathbf{C})$ for all $x \in univ(\mathbf{C})$. Then $h(s(i/x)) = h(s)(i/h(x)) = t(i/h(x)) \in h(F(\mathbf{C}))$ for all $x \in univ(\mathbf{C})$. In other words, $t(i/y) \in h(F(\mathbf{C})) \;\subseteq\; F(\mathbf{D})$ for all $y \in h(univ(\mathbf{C})) = univ(\mathbf{D})$. Thus $t \in \mathsf{ɔ}_i(F(\mathbf{D}))$

□

Also, for an $n$-dimensional naive cylinder $C$, we denote the subset $C \cap \mathbb{D}^n$ by $C_\downarrow$. We now have our main "naive evaluation" theorem.

**Theorem 5.5.** Let $\mathbf{C}$ be a sequence of $n$-dimensional naive cylinders and diagonals, and let $E$ be an expression in CA$_n^+$. Then

$$E(\mathbf{C})_\downarrow \;=\; \bigcap_{\mathbf{D} \in Rep(\mathbf{C})} E(\mathbf{D}).$$

**Proof:**
Let $t \in E(\mathbf{C})_{\downarrow} \subseteq E(\mathbf{C})$, and $\mathbf{D} \in Rep(\mathbf{C})$. Since $\mathbf{C} \rightarrow_h \mathbf{D}$ for some possible world homomorphism $h$, by Theorem 5.4, $h(t) \in E(\mathbf{D})$. Since $t$ is all constants, $h(t) = t$ for all $h$. In other words, $t \in E(\mathbf{D})$, for all $\mathbf{D} \in Rep(\mathbf{C})$.

For the $\supseteq$-direction, let $t \in \bigcap_{\mathbf{D} \in Rep(\mathbf{C})} E(\mathbf{D})$. Then $t \in \mathbb{D}^n$, and for all possible world homomorphisms $h$ it holds that $t \in E(h(\mathbf{C}))$. Since identity is a valid $h$, is follows that $t \in E(\mathbf{C})$, and since $t$ is all constants we have $t \in E(\mathbf{C})_{\downarrow}$. □

**Mixing existential and universal nulls**

We want to achieve a representation mechanism able to handle both universal nulls and naive existential nulls. To this end we need the following definition.

**Definition 5.6.** A naive $n$-dimensional (positive) star-cylinder is a finite subset $\ddot{C}$ of $(\mathbb{D} \cup \mathbb{N} \cup \{*\})^n \times \wp(\Theta_n)$. A naive diagonal is defined as $\ddot{d}_{ij} = \{(x,x) : x \in univ(\ddot{\mathbf{C}})\}$. A sequence of $n$-dimensional star-cylinders and diagonals is denoted $\ddot{\mathbf{C}}$.

After this we extend Definitions 3.1, 3.2, and 3.5 in Section 3 from star-cylinders to naive star-cylinders, by replacing $\mathbb{D}$ with $univ(\ddot{\mathbf{C}})$ or $univ(\ddot{C})$ everywhere. Theorem 3.1 will still hold, but Corollary 3.10 only holds in the weakened form as Corollary 5.10 below. First we need two lemmas and a definition.

**Lemma 5.7.** Suppose all possible world homomorphisms $h$ are extended by letting $h(*) = *$. Let $\ddot{C}$ be an $n$-dimensional naive star-cylinder. Then

$$h([\![\ddot{C}]\!]) = [\![h(\ddot{C})]\!],$$

for all possible world homomorphisms $h$.

**Proof:**
Let $t \in h([\![\ddot{C}]\!])$. Then there exists a tuple $u \in [\![\ddot{C}]\!]$, such that $t = h(u)$. Also there exists a naive star-tuple $\ddot{u} \in [\![\ddot{C}]\!]$, such that $u \le \ddot{u}$. Now it is sufficient to show that $t \le h(\ddot{u})$, for all $i \in \{1, 2, \dots, n\}$.

If $\ddot{u}(i) \in \mathbb{D}$, then $u(i) = \ddot{u}(i)$. Also, homomorphisms are identity on constants and therefore $h(u(i)) = u(i)$, which implies $t(i) = u(i)$.

If $\ddot{u}(i) = *$, then $u(i) \in univ(\ddot{C})$. As a result $h(u(i)) \in univ(h(\ddot{C}))$, which implies $t(i) \le * = h(\ddot{u}(i))$, since homomorphisms map stars to themselves.

If $\ddot{u}(i) \in \mathbb{N}$, then $u(i) = \ddot{u}(i)$, which implies $t(i) = h(u(i)) = h(\ddot{u}(i))$.

For the other direction, let $t \in [\![h(\ddot{C})]\!]$. Then there exists a tuple $\ddot{t} \in h(\ddot{C})$ and a tuple $\ddot{u} \in \ddot{C}$, such that $t \le \ddot{t}$ and $\ddot{t} = h(\ddot{u})$. Consequently, $t \le h(\ddot{u})$. We show that we can find a tuple $u \in [\![\ddot{u}]\!]$ such that $h(u) = t$.

If $\ddot{u}(i) \in \mathbb{D}$, then $u(i) = \ddot{u}(i)$. Since $h$ is the identity on constants $h(\ddot{u}(i)) = \ddot{u}(i)$, which implies $t(i) = u(i)$.

If $\ddot{u}(i) = *$, then $h(\ddot{u}(i)) = *$. As $h$ is onto $univ(h(\ddot{C}))$, it follows that there is a value $\ddot{u}(i) \in univ(\ddot{C})$, such that $h(u(i)) = t(i)$.

If $\ddot{u}(i) \in \mathbb{N}$, then $u(i) = \ddot{u}(i)$, which implies $t(i) = h(\ddot{u}(i)) = h(u(i))$. □

**Definition 5.8.** Let $\mathcal{I}$ and $\mathcal{J}$ be sets of instances. We say that $\mathcal{I}$ and $\mathcal{J}$ are *co-initial*, denoted $\mathcal{I} \sim \mathcal{J}$, if for each instance $J \in \mathcal{J}$ there is an instance $I \in \mathcal{I}$, and a possible world homomorphism $h$, such that $I \to_h J$, and vice-versa.

We extend Definition 5.2 from infinite instances and sequences of cylinders to sequences of naive star-cylinders as follows.

**Definition 5.9.** Let $\ddot{\mathbf{C}}$ be a sequence of $n$-dimensional naive star-cylinders and diagonals with $univ(\ddot{\mathbf{C}}) = \mathbb{D} \cup \mathbb{N}$. Then the (infinite) set of (infinite) $n$-dimensional cylinders represented by $\ddot{\mathbf{C}}$ is

$$Rep([\![\ddot{\mathbf{C}}]\!]) = \{\mathbf{D} : [\![\ddot{\mathbf{C}}]\!] \to_h \mathbf{D}\}.$$

In the context of naive star-cylinders Corollary 3.10 will be weakened as follows.

**Corollary 5.10.** For every SCA$_n^+$-expression $\dot{E}$ and the corresponding CA$_n$-expression $E$, it holds that

$$Rep([\![\dot{E}(\ddot{\mathbf{C}})]\!]) \sim E(Rep([\![\ddot{\mathbf{C}}]\!])),$$

for every sequence of $n$-dimensional naive star-cylinders and star-diagonals $\ddot{\mathbf{C}}$.

**Proof:**
We have $Rep([\![\dot{E}(\ddot{\mathbf{C}})]\!]) \sim Rep(E([\![\ddot{\mathbf{C}}]\!]))$ from Corollary 3.10. It remains to show that $Rep(E([\![\ddot{\mathbf{C}}]\!]) \sim E(Rep([\![\ddot{\mathbf{C}}]\!]))$. Let's denote $[\![\ddot{\mathbf{C}}]\!]$ by $\mathbf{C}$. We'll show that $Rep(E(\mathbf{C})) \sim E(Rep(\mathbf{C}))$.

Let $D \in E(Rep([\![\mathbf{C}]\!]))$, meaning that $D = E(\mathbf{C}')$ for some $\mathbf{C}' \in Rep([\![\mathbf{C}]\!])$. Then there is a possible world homomorphism $h$ such that $\mathbf{C} \to_h \mathbf{C}'$. Theorem 5.4 then yields $E(\mathbf{C}) \to_h E(\mathbf{C}')$, and since $E(\mathbf{C}) \in Rep(E(\mathbf{C}))$ one direction of Definition 5.8 is satisfied.

Then let $D \in Rep(E(\mathbf{C}))$. Then there is a possible world homeomorphism $h$, such that $E(\mathbf{C}) \to_h D$. Since $E(\mathbf{C}) \in E(Rep(\mathbf{C}))$, it means that the vice-versa direction is also satisfied. □

**Naive evaluation of existential nulls**

For a naive star-cylinder $\ddot{\mathbf{C}}$ we let $\ddot{\mathbf{C}}_\downarrow = \ddot{\mathbf{C}} \cap (\mathbb{D} \cup \{*\})^n$. We note that obviously $[\![\ddot{\mathbf{C}}_\downarrow]\!] = ([\![\ddot{\mathbf{C}}]\!])_\downarrow$, and that if $Rep(\ddot{\mathbf{C}}) \sim Rep(\ddot{\mathbf{D}})$ then $\ddot{\mathbf{C}}_\downarrow = \ddot{\mathbf{D}}_\downarrow$. We now have the main result of this section.

**Theorem 5.11.** For every SCA$^+$-expression $\dot{E}$ and the corresponding CA$^+$-expression $E$, it holds that

$$[\![\dot{E}(\ddot{\mathbf{C}})_\downarrow]\!] = \bigcap_{\mathbf{D} \in Rep([\![\ddot{\mathbf{C}}]\!])} E(\mathbf{D}).$$

for every sequence $\ddot{\mathbf{C}}$ of naive star-cylinders and diagonals.

**Proof:**
$[\![\dot{E}(\ddot{\mathbf{C}})_\downarrow]\!] = [\![\dot{E}(\ddot{\mathbf{C}})]\!]_\downarrow = (E([\![\ddot{\mathbf{C}}]\!]))_\downarrow = \bigcap_{\mathbf{C} \in Rep([\![\ddot{\mathbf{C}}]\!])} E(\mathbf{C})$. The second equality follows from Corollary 5.10, the third from Theorem 5.5. □

**Stored databases with universal and existential nulls (ue-databases)**

We extend the Definitions 4.1 and 4.2 of Section 4 from stored databases to naive stored databases (ue-databases) by substituting $\mathbb{D}$ with $\mathbb{D} \cup \mathbb{N}$ everywhere. Lemma 4.3 then becomes

**Lemma 5.12.** Let $\ddot{\mathbf{C}}$ be a stored ue-database with universe $\mathbb{D} \cup \mathbb{N}$. Then $[[\dot{\mathsf{h}}^n(\ddot{\mathbf{R}})]] = \mathsf{h}^n(I(\ddot{\mathbf{R}}))$.

We first note that Theorem 4.4 in the ue-setting becomes

**Theorem 5.13.** For every $\mathrm{FO}_n^+$-formula $\varphi$ there is an $\mathrm{SCA}_n^+$ expression $\dot{E}_\varphi$, such that

$$[[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))]] = \mathsf{h}^n(\varphi^{I(\ddot{\mathbf{R}})})$$

for every stored ue-database $\ddot{\mathbf{R}}$

We also have

**Theorem 5.14.** For every $\mathrm{FO}_n^+$-formula $\varphi$ there is a $\mathrm{CA}_n^+$ expression $\dot{E}_\varphi$, such that

$$Rep([[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))]]) \sim \{\mathsf{h}^n(\varphi^J) : J \in Rep([[\ddot{\mathbf{R}}]])\}$$

for every stored ue-database $\ddot{\mathbf{R}}$

We have now arrived our main theorem for ue-databases.

**Theorem 5.15.** For every $\mathrm{FO}_n^+$-formula $\varphi$ there is an $\mathrm{SCA}_n^+$ expression $\dot{E}_\varphi$, such that

$$[[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))_\downarrow]] = \bigcap_{J \in Rep([[\ddot{\mathbf{R}}]])} \mathsf{h}^n(\varphi^J)$$

for every stored ue-database $\ddot{\mathbf{R}}$

**Proof:**
We have $\{\mathsf{h}^n(\varphi^J) : J \in Rep([[\ddot{\mathbf{R}}]])\} \sim Rep([[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))]])$ by Theorem 5.14. Hence $\bigcap_{J \in Rep([[\ddot{\mathbf{R}}]])} \mathsf{h}^n(\varphi^J) = \bigcap Rep([[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))]]) = ([[\dot{E}_\varphi(\ddot{\mathsf{h}}^n(\ddot{\mathbf{R}}))]])_\downarrow = [[\dot{E}_\varphi(\dot{\mathsf{h}}^n(\ddot{\mathbf{R}}))_\downarrow]]$. □

# 6. Complexity

In this section we provide complexity results for Cylindric Star Algebra and Star Cylinders. We start by defining the size of extended star-cylinders. Let $\dot{\mathbf{C}}$ be a sequence of $n$-dimensional extended star-cylinders and diagonals. By $|\dot{\mathbf{C}}|$ we denote the larger of the number of star-tuples in $\dot{\mathbf{C}}$ and the number of literals in the star-tuple with the largest condition column $n + 1$. The same notation also applies to sequences of naive star-cylinders $\ddot{\mathbf{C}}$. First, we show that star-cylinders can be transformed into normal form in polynomial time.

**Theorem 6.1.** Let $\dot{C}$ be an $n$-dimensional extended star-cylinder. Then $\dot{C}$ can in polynomial time be transformed into a normal form star-cylinder $\dot{C}'$ such that $[[\dot{C}']] = [[\dot{C}]]$.

**Proof:**
The first requirement is that each extended star-tuple has a consistent and logically closed set of literals. To achieve this, we associate with each $\dot{t} \in \dot{C}$ a graph with vertices $\{1, \ldots, n\}$, with a blue edge $\{i, j\}$ if $(i = j) \in \dot{t}(n+1)$, and a red edge $\{i, j\}$ if $(i \neq j) \in \dot{t}(n+1)$. Next, we compute the transitive closure of the graph wrt the blue edges. To account for the inequality conditions, we further extend the graph by repeatedly checking if there is a red edge $\{i, j\}$ and a blue edge $\{j, k\}$, in which case we add, unless already there, a red edge $\{i, k\}$. Then each connected component of blue edges represents an equivalence class of dimensions with equal values in $\dot{t}$, unless there is a pair $\{i, j\}$ that has both a blue and a red edge, in which case $\dot{t}(n+1) \vDash \mathsf{false}$, $[[\{\dot{t}\}]] = \varnothing$, and $\dot{t}$ can be removed. We need to consider conditions of the form $(i \neq a)$ in star-tuples $\dot{t}$ as well. They will be handled similarly to the inequality conditions. More precisely, for each $(i \neq a)$ we add a black self-loop labelled $a$ to vertex $i$. If there is a blue edge $\{i, j\}$, we recursively add an $a$-labelled self-loop to vertex $j$. In the end, if there is a vertex $i$ having an $a$-labelled self-loop while $\dot{t}(i) = b \neq a$, we again have $\dot{t}(n+1) \vDash \mathsf{false}$, $[[\{\dot{t}\}]] = \varnothing$, and therefore remove star-tuple $\dot{t}$. All the above graph-manipulation can clearly be performed in time polynomial in $n$.

We still need to verify that $\dot{t}$ satisfies conditions (1) – (3) of Definition 3.12. If $\dot{t}$ violates a condition, it is easy to see that $[[\{\dot{t}\}]] = \varnothing$, so $\dot{t}$ can be removed from $\dot{C}$. The only exception is for condition (1), when $\dot{t}(n+1) \vDash (i = j)$, $t(i) = *$, and $t(j) = a \in \mathbb{D}$. In this case $\dot{t}$ is retained, but with $t(i)$ replaced by $a$. If $t(j) = *$ and $t(i) = a$ then $t(j)$ is replaced with $a$.

The remaining star-tuples form the normalized star-cylinder $\dot{C}'$, and $[[\dot{C}']] = [[\dot{C}]]$ by construction. □

Next, we investigate the complexity of evaluating SCA-expressions over naive star-cylinders and then we characterize various membership and containment problems. It turns out $\dot{E}(\ddot{\mathbf{C}})$ can be computed efficiently for SCA$_n$-expressions $\dot{E}$, even though universal quantification and negation are allowed. First we need the following general result.

**Theorem 6.2.** Let $\dot{E}$ be a fixed SCA$_n$-expression, and $\dot{\mathbf{C}}$ a sequence of $n$-dimensional extended star-cylinders and diagonals. Then there is a polynomial $\pi$, such that $|\dot{E}(\dot{\mathbf{C}})| = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$. Moreover, $\dot{E}(\dot{\mathbf{C}})$ can be computed in time $\mathcal{O}(\pi(|\dot{\mathbf{C}}|))$, and if negation is not used in $\dot{E}$ this applies to naive star-cylinders $\ddot{\mathbf{C}}$ as well.

**Proof:**
Since $\dot{E}$ is fixed it is sufficient to prove the first claim for each operator separately. Note that since $\dot{E}$ is fixed, it follows that $n$ is also fixed.

1. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathsf{C}}_p(\dot{\mathbf{C}})$, then $|\dot{E}(\dot{\mathbf{C}})| = |\dot{C}_p| \leq |\dot{\mathbf{C}}| = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$.
2. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathsf{d}}_{ij}(\dot{\mathbf{C}})$, then $|\dot{E}(\dot{\mathbf{C}})| = \mathcal{O}(|\dot{\mathbf{C}}|) \times \mathcal{O}(1) = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$.
3. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathsf{C}}_p(\dot{\mathbf{C}}) \cup \dot{\mathsf{C}}_q(\dot{\mathbf{C}})$, then $|\dot{E}(\dot{\mathbf{C}})| \leq |\dot{\mathbf{C}}| = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$.

4. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathsf{C}}_p(\dot{\mathbf{C}}) \dot{\cap} \dot{\mathsf{C}}_q(\dot{\mathbf{C}})$, then the number of tuples in $\dot{E}(\dot{\mathbf{C}})$ is at most $|\dot{\mathbf{C}}|^2$, and each tuple in the output can have a condition of length at most $2 \cdot |\dot{\mathbf{C}}|$. As a result, $|\dot{E}(\dot{\mathbf{C}})| \leq 2 \cdot |\dot{\mathbf{C}}|^3 = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$.

5. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathsf{c}}_i(\mathsf{C}_p(\dot{\mathbf{C}}))$, then $|\dot{E}(\dot{\mathbf{C}})| \leq |\dot{C}_p| \leq |\dot{\mathbf{C}}| = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$.

6. For the case $\dot{E}(\dot{\mathbf{C}}) = \dot{\mathfrak{I}}_i(\mathsf{C}_p(\dot{\mathbf{C}}))$ we note that $\dot{\mathfrak{I}}_i(\mathsf{C}_p(\dot{\mathbf{C}})) \subseteq (\mathsf{C}_p(\dot{\mathbf{C}}) \dot{\cap} \dot{A}) \subseteq \dot{A}$. We can construct the star-tuples in $\dot{A}$ by iterating over the star-tuples in $\dot{C}_p$ and using the constants in $A$. This means that $|\dot{A}| = (n \times (|A|)) + (2^n + |A|) \leq \mathcal{O}(1) \times \mathcal{O}(|\dot{\mathbf{C}}|) + \mathcal{O}(1) \times \mathcal{O}(|\dot{\mathbf{C}}|) = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$. Note that $n$ is the dimensionality of $\dot{\mathbf{C}}$ and is a constant.

7. If $\dot{E}(\dot{\mathbf{C}}) = \dot{\neg}(\mathsf{C}_p(\dot{\mathbf{C}}))$, then similar to the inner cylindrification we have $\dot{\neg}\,\dot{C}_p \subseteq \dot{A}$ which implies $|\dot{E}(\dot{\mathbf{C}})| = \mathcal{O}(\pi(|\dot{\mathbf{C}}|))$. □

**Membership.** In the membership problems, we ask if an ordinary tuple $t$ belongs to the set specified by a (naive) star-cylinder, of by a fixed expression $\dot{E}$ and a (naive) star-cylinder. In other words, all results refer to data complexity.

**Theorem 6.3.** Let $t \in \mathbb{D}^n$ and $\ddot{\mathbf{C}}$ a sequence of $n$-dimensional naive star-cylinders and diagonals. The membership problems and their respective data complexities are as follows.

1. $t \overset{?}{\in} \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$ is in polytime for positive $E$.

2. $t \overset{?}{\in} \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$ is coNP-complete for $E$ where negation is allowed in equality atoms only.

**Proof:**

1. By Theorem 5.11, we have $\bigcap E(Rep([\![\ddot{\mathbf{C}}]\!])) = [\![\dot{E}(\ddot{\mathbf{C}})_{\downarrow}]\!]$, so to test if $t \in \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$, we compute $\dot{E}(\ddot{\mathbf{C}})_{\downarrow}$, and see if there is a star-tuple $\dot{t} \in \dot{E}(\ddot{\mathbf{C}})_{\downarrow}$, such that $t \leq \dot{t}$. By Theorem 6.2, $\dot{E}(\ddot{\mathbf{C}})_{\downarrow}$ can be computed in polytime.

2. To check if $t \notin \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$, it is sufficient to find a homomorphism $h$ such that $t \notin h([\![\ddot{\mathbf{C}}]\!])$. We guess the homomorphism $h$, and check in polytime if $t \notin h([\![\ddot{\mathbf{C}}]\!])$. Thus $t \notin \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$ is in NP, and $t \in \bigcap E(Rep([\![\ddot{\mathbf{C}}]\!]))$ is in coNP. The lower bound follows from Theorem 5.2.2 in [14]. □

**Containment.** The containment problems ask for containment of star-cylinders (naive star-cylinders), or views over star-cylinders (naive star-cylinders). We have the following.

**Theorem 6.4.** Let $\dot{\mathbf{C}}$ and $\dot{\mathbf{D}}$ (resp. $\ddot{\mathbf{C}}$ and $\ddot{\mathbf{D}}$) be sequences of $n$-dimensional (naive) star-cylinders and diagonals. Then

1. $E_1([\![\dot{\mathbf{C}}]\!]) \overset{?}{\subseteq} E_2([\![\dot{\mathbf{D}}]\!])$ is in polytime for $\mathrm{CA}_n$ expression $E_1$ and $E_2$.

2. $Rep([\![\ddot{\mathbf{C}}]\!]) \overset{?}{\subseteq} Rep([\![\ddot{\mathbf{D}}]\!])$ is NP-complete.

3. $E_1(Rep([[\ddot{\mathbf{C}}]])) \overset{?}{\subseteq} E_2(Rep([[\ddot{\mathbf{D}}]]))$ is $\Pi_2^p$-complete for positive $E_1$ and $E_2$.

**Proof:**

1. By Lemma 3.15, we have $[[\dot{E}_1(\dot{\mathbf{C}})]] \subseteq [[\dot{E}_2(\dot{D})]]$ if and only if $\dot{E}_1(\dot{\mathbf{C}}) \Cap \dot{A} \preceq \dot{E}_2(\dot{D}) \Cap \dot{A}$. The latter dominance is true if and only if for each star-tuple $\dot{t} \in \dot{E}_1(\dot{\mathbf{C}}) \Cap \dot{A}$ there is a star-tuple $\dot{u} \in \dot{E}_2(\dot{\mathbf{D}}) \Cap \dot{A}$, such that $\dot{t} \leq \dot{u}$. From Theorem 6.2 we know that $\dot{E}_1(\dot{\mathbf{C}}) \Cap \dot{A}$ and $\dot{E}_2(\dot{\mathbf{D}}) \Cap \dot{A}$ can be computed in polytime.

2. We first extend the domain of possible world homomorphisms by stipulating that they are the identity on $*$. Then it is easy to see that $Rep([[\ddot{\mathbf{C}}]]) \subseteq Rep([[\ddot{\mathbf{D}}]])$ if and only if there exists a possible world homomorphism $h$ such that $\ddot{\mathbf{D}} \to_h \ddot{\mathbf{C}}$. This makes the problem NP-complete.

3. The lower bound follows from Theorem 4.2.2 in [14], For the upper bound we observe that $E_1(Rep([[\ddot{\mathbf{C}}]])) \subseteq E_2(Rep([[\ddot{\mathbf{D}}]])$ iff for every $\mathbf{C} \in Rep([[\ddot{\mathbf{C}}]])$ there exists a $\mathbf{D} \in Rep([[\ddot{\mathbf{D}}]])$ such that $E_1(\mathbf{C}) = E_2(\mathbf{D})$ iff for every possible world homomorphism $h$ on $\ddot{\mathbf{C}}$ there exists a possible world homomorphism $g$ on $\ddot{\mathbf{D}}$ such that $E_1(h([[\ddot{\mathbf{C}}]])) = E_2(g([[\ddot{\mathbf{D}}]]))$. By Corollary 3.10, this equality holds iff $[[\dot{E}_1(h(\ddot{\mathbf{C}}))]] = [[\dot{E}_1(g(\ddot{\mathbf{D}}))]]$. By Lemma 3.15, the last equality holds iff $E_1(h([[\ddot{\mathbf{C}}]])) \Cap \dot{A} \preceq E_2(g([[\ddot{\mathbf{D}}]])) \Cap \dot{A}$, and vice-versa. By Theorem 6.2, the star-cylinders in the two dominances $\preceq$ can be computed in polynomial time. □

# 7. Related and future work

Cylindric Set Algebra gave rise to a whole subfield of Algebra, called Cylindric Algebra. For a fairly recent overview, the reader is referred to [15]. Within database theory, a simplified version of the star-cylinders and a corresponding Codd-style positive relational algebra with evaluation rules "$* = *$" and "$* = a$" was proposed by Imielinski and Lipski in [8]. Such cylinders correspond to the structures in *diagonal-free* Cylindric Set Algebras [9, 10]. The exact FO-expressive power of these diagonal-free star-cylinders is an open question. Nevertheless, using the techniques of this paper, it can be shown that naive existential nulls can be seamlessly incorporated in diagonal-free star-cylinders.

In addition to the above and the work described in Section 1, Imielinski and Lipski also showed in [8] that the fact that Codd's Relational Algebra does not have a finite axiomatization, and the fact that equivalence of expressions in it is undecidable, follow from known results in Cylindric Algebra. This is of course true for a host of general results in Mathematical Logic.

Yannakakis and Papadimitriou [16] formulated an algebraic version of dependency theory using Codd's Relational Algebra. Around the same time Cosmadakis [17] proposed an interpretation of dependency theory in terms of equations over certain types of expressions in Cylindric Set Algebra, and described a complete finite axiomatization of his system. It was however later shown by Düntsch, Hodges, and Mikulas [18, 19], again using known results from Cylindric Algebra, that Cosmadakis's axiomatization was incomplete, and that no finite complete axiomatization exists.

Interestingly, it turns out that one of the models for constraint databases in [20] by Kanellakis, Kuper, and Revesz — the one where the constraints are equalities over an infinite domain — is equivalent with our star-tables. Even though [20] develops a bottom-up (recursive) evaluation mechanism for

FO-queries, the mechanism is goal-oriented and contrary to our star-cylinders, there is no algebra operating on the constraint databases. We note however that the construction of the sieve $\dot{A}$ in Section 3 is inspired by the constraint solving techniques of [20]. It therefore seems that our star-cylinders and algebra can be made to handle inequality constraints on dense linear orders as well as polynomial constraints over real-numbers, as is done in [20]. We also note that our work is related to the orbit finite sets, treated in a general computational framework in [21].

As noted in Section 1, the existential nulls have long been well understood. According to [22] the fact that positive queries (no negation, but allowing universal quantification) are preserved under onto-homomorphisms are folklore in the database community. Using this monotonicity property, Libkin [3] has recently shown that positive queries can be evaluated naively on finite existential databases $I$ under a so called *weak closed world assumption*, where $Rep(I)$ consists of all complete instances $J$, such that $h(I) \subseteq J$ and $J$ only involves constants that occur in $I$, and furthermore $h$ is onto from the finite universe of $I$ to the finite universe of $J$. Our Theorem 5.5 generalizes Libkin's result to infinite databases. In this context it is worth noting that Lyndon's Positivity Theorem [23] tells us that a first order formula is preserved under onto-homomorphisms on all structures if and only if it is equivalent to a positive formula. It has subsequently been shown that the only-if direction fails for finite structures [24, 25]. Since our star-cylinders represent neither finite nor unrestricted infinite structures, it would be interesting to know whether the only-if direction holds for infinite structures represented by star-cylinders. If it does, it would mean that our Theorem 5.5 would be optimal, meaning that if $\varphi$ is not equivalent to a positive formula, then naive evaluation does not work for $\varphi$ on databases represented by naive star-cylinders.

Finally we note that Sundarmurthy et al. [11] have generalized the conditional tables of [1, 26] by replacing the labelled nulls with a single null $\mathbf{m}$ that initially represents all possible domain values. They then add constraints on the occurrences of these $\mathbf{m}$-values, allowing them to represent a finite or infinite subset of the domain, and to equate distinct occurrences of $\mathbf{m}$. Sundarmurthy et al. then show that their $\mathbf{m}$-tables are closed under positive (but not allowing universal quantification) queries by developing a difference-free Codd-style relational algebra that $\mathbf{m}$-tables are closed under. Merging our approach with theirs could open up interesting possibilities.

# References


[1] Imielinski T, Lipski W. Incomplete Information in Relational Databases. *J. ACM*, 1984. **31**(4):761–791.

[2] Fagin R, Kolaitis PG, Popa L, Tan WC. Reverse data exchange: coping with nulls. In: PODS. 2009 pp. 23–32.

[3] Gheerbrant A, Libkin L, Sirangelo C. Naïve Evaluation of Queries over Incomplete Databases. *ACM Trans. Database Syst.*, 2014. **39**(4):31:1–31:42. doi:10.1145/2691190.2691194. URL `http://doi.acm.org/10.1145/2691190.2691194`.

[4] Grahne G, Moallemi A, Onet A. Intuitionistic Data Exchange. In: Proc. of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management, Lima, Peru, May 6 - 8, 2015. 2015 .

[5] Libkin L. Negative Knowledge for Certain Query Answers. In: Ortiz M, Schlobach S (eds.), Web Reasoning and Rule Systems - 10th International Conference, RR 2016, Aberdeen, UK, September 9-11,

2016, Proceedings, volume 9898 of *Lecture Notes in Computer Science*. Springer. ISBN 978-3-319-45275-3, 2016 pp. 111–127. doi:10.1007/978-3-319-45276-0_9. URL https://doi.org/10.1007/978-3-319-45276-0_9.

[6] Biskup J. Extending the Relational Algebra for Relations with Maybe Tuples and Existential and Universal Null Values. *Fundam. Inform.*, 1984. **7**(1):129–150.

[7] Biskup J. A Foundation of Codd's Relational Maybe-Operations. *ACM Trans. Database Syst.*, 1983. **8**(4):608–636. doi:10.1145/319996.320014. URL http://doi.acm.org/10.1145/319996.320014.

[8] Imielinski T, Lipski W. The Relational Model of Data and Cylindric Algebras. *J. Comput. Syst. Sci.*, 1984. **28**(1):80–102. doi:10.1016/0022-0000(84)90077-1. URL https://doi.org/10.1016/0022-0000(84)90077-1.

[9] Henkin L, Monk JD, Tarski A. Cylindric Algebras–Part I,volume 64 of Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1971.

[10] Henkin L, Monk JD, Tarski A. Cylindric Algebras–Part II, volume 115 of Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1985.

[11] Sundarmurthy B, Koutris P, Lang W, Naughton JF, Tannen V. m-tables: Representing Missing Data. In: 20th International Conference on Database Theory, ICDT 2017, March 21-24, 2017, Venice, Italy. 2017 pp. 21:1–21:20. doi:10.4230/LIPIcs.ICDT.2017.21. URL https://doi.org/10.4230/LIPIcs.ICDT.2017.21.

[12] den Bussche JV. Applications of Alfred Tarski's Ideas in Database Theory. In: Fribourg L (ed.), Computer Science Logic, 15th International Workshop, CSL 2001. 10th Annual Conference of the EACSL, Paris, France, September 10-13, 2001, Proceedings, volume 2142 of *Lecture Notes in Computer Science*. Springer. ISBN 3-540-42554-3, 2001 pp. 20–37. doi:10.1007/3-540-44802-0_2. URL https://doi.org/10.1007/3-540-44802-0_2.

[13] Abiteboul S, Hull R, Vianu V. Foundations of Databases. Addison-Wesley, 1995. ISBN 0-201-53771-0.

[14] Abiteboul S, Kanellakis PC, Grahne G. On the Representation and Querying of Sets of Possible Worlds. *Theor. Comput. Sci.*, 1991. **78**(1):158–187. doi:10.1016/0304-3975(51)90007-2. URL https://doi.org/10.1016/0304-3975(51)90007-2.

[15] Andréka H, Ferenczi M, Németi I. Cylindric-like Algebras and Algebraic Logic, volume 22. Springer Science & Business Media, 2014.

[16] Yannakakis M, Papadimitriou CH. Algebraic Dependencies. *J. Comput. Syst. Sci.*, 1982. **25**(1):2–41. doi:10.1016/0022-0000(82)90008-3. URL https://doi.org/10.1016/0022-0000(82)90008-3.

[17] Cosmadakis SS. Database Theory and Cylindric Lattices (Extended Abstract). In: 28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987. 1987 pp. 411–420. doi:10.1109/SFCS.1987.17. URL https://doi.org/10.1109/SFCS.1987.17.

[18] Hodkinson I, Mikulás S. Axiomatizability of reducts of algebras of relations. *Algebra Universalis*, 2000. **43**(2):127–156.

[19] Düntsch I, Mikulás S. Cylindric structures and dependencies in relational databases. *Theor. Comput. Sci.*, 2001. **269**(1-2):451–468. doi:10.1016/S0304-3975(01)00016-0. URL https://doi.org/10.1016/S0304-3975(01)00016-0.

[20] Kanellakis P, Kuper G, Revesz P. Constraint Query Languages. *Journal of Computer and System Sciences*, 1995. **51**(1):26 – 52.

[21] Bojańczyk M. Orbit-Finite Sets and Their Algorithms (Invited Talk). In: 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland. 2017 pp. 1:1–1:14. doi:10.4230/LIPIcs.ICALP.2017.1. URL https://doi.org/10.4230/LIPIcs.ICALP.2017.1.

[22] Deutsch A, Nash A, Remmel JB. The chase revisited. In: Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada. 2008 pp. 149–158. doi:10.1145/1376916.1376938. URL http://doi.acm.org/10.1145/1376916.1376938.

[23] Lyndon RC. Properties preserved under homomorphism. *Pacific J. Math.*, 1959. **9**(1):143–154. URL https://projecteuclid.org:443/euclid.pjm/1103039459.

[24] Ajtai M, Gurevich Y. Monotone Versus Positive. *J. ACM*, 1987. **34**(4):1004–1015. doi:10.1145/31846.31852. URL http://doi.acm.org/10.1145/31846.31852.

[25] Stolboushkin AP. Finitely Monotone Properties. In: Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science, LICS '95. IEEE Computer Society, Washington, DC, USA. ISBN 0-8186-7050-6, 1995 pp. 324–. URL http://dl.acm.org/citation.cfm?id=788017.788758.

[26] Grahne G. The Problem of Incomplete Information in Relational Databases, volume 554 of *Lecture Notes in Computer Science*. Springer, 1991. ISBN 3-540-54919-6. doi:10.1007/3-540-54919-6. URL https://doi.org/10.1007/3-540-54919-6.

# Appendix

This Appendix contains the proofs of Theorems 2.5 and 2.6. Before we get to the proofs, we need some technical prerequisites.

**Proposition 7.1.** [9]. Let $C$ be an $n$-dimensional cylinder, and $i, j \in \{1, \ldots, n\}$. Then

1. $\mathsf{z}^i_j(C) = \mathsf{z}^j_i(C)$.
2. $\mathsf{z}^i_j(\mathsf{z}^j_i(C)) = C$.
3. $\mathsf{c}_i(\mathsf{z}^i_j(C)) = \mathsf{z}^i_j(\mathsf{c}_j(C))$.
4. If $i \neq j$ then $\mathsf{z}^i_j(C \smallsetminus C') = \mathsf{z}^i_j(C) \smallsetminus \mathsf{z}^i_j(C')$.
5. If $\mathsf{c}_i(C) = C$ and $\mathsf{c}_j(C) = C$ then $\mathsf{z}^i_j(C) = C$.

**Proposition 7.2.** Let $i, j, k$ be pairwise distinct natural numbers, such that $\{i, j, k\} \cap \{1, 2, 3\} = \varnothing$, and let $C$ be an $n$-dimensional cylinder that is 2-full Then

$$\mathsf{z}^{i,k}_{1,2}(\mathsf{z}^{3,2,1}_{k,j,i}(C)) = \mathsf{z}^{1,2,3}_{1,j,2}(C).$$

**Proof:**

$$\mathsf{z}^{i,k}_{1,2}(\mathsf{z}^{3,2,1}_{k,j,i}(C)) = \mathsf{z}^{i,k,3,2,1}_{1,2,k,j,i}(C) = \mathsf{z}^{i,3,2,2,1}_{1,2,k,j,i}(C) = \mathsf{z}^{i,3,2,1}_{1,2,j,i}(C) = \mathsf{z}^{i,1,3,2}_{1,i,2,j}(C) = \mathsf{z}^{i,1,3,2}_{i,1,2,j}(C) = \mathsf{z}^{1,3,2}_{1,2,j}(C).$$

The second equality follows from Theorem 1.5.18 in [9], the third equality holds since $\mathsf{c}_2(C) = C$ and $\mathsf{c}_k(C) = C$, the fourth since $\{1, i\} \cap \{2, 3, j\} = \varnothing$. The last two equalities follow from Theorem 1.5.17 and 1.5.13 in [9], respectively. □

We are now ready for the main proofs.

**Theorem 2.5.** For all FO$_n$-formulas $\varphi$, there is a CA$_n$ expression $E_\varphi$, such that

$$E_\varphi(\mathsf{h}^n(I)) = \mathsf{h}^n(\varphi^I),$$

for all instances $I$.

**Proof:**
We prove the stronger claim: For all FO$_n$-formulas $\varphi$, for all $\psi \in sub(\varphi)$, with $free(\psi) = \{x_{i_1}, \ldots, x_{i_k}\}$, there is an CA$_n$ expression $E_\psi$, such that

$$\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\psi(\mathsf{h}^n(I))\Big) = \mathsf{h}^n(\psi^I),$$

for all instances $I$. The main claim the follows since $\varphi \in sub(\varphi)$, and the outermost sequence of swappings can be considered part of the final expression $E_\varphi$. In all cases below we assume wlog[11] that $k < n$ so that the $k + 1$:st column can be used in the necessary swappings.

[11]If $k = n$ we can introduce an additional variable $x_{n+1}$ and the conjunct $\exists x_{n+1}(x_{n+1} \approx x_{n+1})$ which would assure that the $n + 1$:st dimension is full. Alternatively, we could introduce swapping as a primitive in the algebra. This however would require a corresponding renaming operator in the FO-formulas, see [9].

- $\psi = R_p(x_{i_1}, \ldots, x_{i_k})$, where $k = ar(R_p)$. We let $E_\psi = \mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathsf{C}_p)$. We have

$$\begin{aligned}
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\psi(\mathsf{h}^n(I))\Big) &=\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathsf{C}_p(\mathsf{h}^n(I))\Big) &=\\
& &\textit{By Proposition 7.1 (2)}\\
&\mathsf{C}_p(\mathsf{h}^n(I)) &=\\
&\mathsf{h}^n(R_p^I) &=\\
&\mathsf{h}^n(\psi^I).
\end{aligned}$$

- $\psi = x_i \approx x_j$. We assume wlog that $n > 2$ so that swaps can be performed. We let $E_\psi = \mathsf{d}_{ij}$. We then have

$$\begin{aligned}
& & &\{t \in \mathbb{D}^n \,:\, t(1) = t(2)\} &=\\
&\mathsf{z}^{i,j}_{1,2}\Big(E_\psi(\mathsf{h}^n(I))\Big) &= \quad &\{(a,a) \,:\, a \in \mathbb{D}\} \times \mathbb{D}^{n-2} &=\\
&\mathsf{z}^{i,j}_{1,2}\Big(\mathsf{d}_{ij}\Big) &= \quad &\mathsf{h}^n(\{(a,a) \,:\, a \in \mathbb{D}\}) &=\\
&\mathsf{z}^{i,j}_{1,2}\Big(\{t \in \mathbb{D}^n \,:\, t(i) = t(j)\}\Big) &= \quad &\mathsf{h}^n((x_i \approx x_j)^I) &=\\
& & &\mathsf{h}^n(\psi^I).
\end{aligned}$$

- $\psi = \neg\xi$, with $free(\xi) = \{x_{i_1}, \ldots, x_{i_k}\}$. We assume wlog that $k < n$. Then $E_\psi = \overline{E_\xi}$, and the inductive hypothesis is

$$\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\xi(\mathsf{h}^n(I))\Big) \;=\; \mathsf{h}^n(\xi^I)$$

We have

$$\begin{aligned}
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\psi(\mathsf{h}^n(I))\Big) &=\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\overline{E_\xi(\mathsf{h}^n(I))}\Big) &=\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathbb{D}^n \smallsetminus E_\xi(\mathsf{h}^n(I))\Big) &=\\
& &\textit{By Proposition 7.1 (2)}\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathbb{D}^n \smallsetminus (\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}(E_\xi(\mathsf{h}^n(I)))))\Big) &=\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathbb{D}^n \smallsetminus (\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathsf{h}^n(\xi^I)))\Big) &=\\
& &\textit{By Proposition 7.1 (5)}\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathbb{D}^n) \smallsetminus (\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathsf{z}^n(\xi^I)))\Big) &=\\
& &\textit{By Proposition 7.1 (4)}\\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}(\mathbb{D}^n \smallsetminus \mathsf{h}^n(\xi^I)))\Big) &=\\
& &\textit{By Proposition 7.1 (2)}\\
&D^n \smallsetminus \mathsf{h}^n(\xi^I) &=\\
&\mathsf{h}^n((\neg\xi)^I) &=\\
&\mathsf{h}^n(\psi^I).
\end{aligned}$$

- $\psi = \xi \wedge \chi$, with $free(\psi) = \{x_{i_1}, \ldots, x_{i_k}\}$, $free(\xi) = \{x_{r_1}, \ldots, x_{r_p}\}$, $free(\chi) = \{x_{s_1}, \ldots, x_{s_q}\}$, $free(\psi) = free(\xi) \cup free(\chi)$, and[12] $free(\xi) \cap free(\chi) = \varnothing$. Now $E_\psi = E_\xi \bigcap E_\chi$. The inductive hypothesis is

$$\mathsf{z}^{r_1,\ldots,r_p}_{1,\ldots,p}\Big(E_\xi(\mathsf{h}^n(I))\Big) = \mathsf{h}^n(\xi^I).$$

$$\mathsf{z}^{s_1,\ldots,s_q}_{1,\ldots,q}\Big(E_\chi(\mathsf{h}^n(I))\Big) = \mathsf{h}^n(\chi^I).$$

We have

$$\begin{aligned}
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\psi(\mathsf{h}^n(I))\Big) && = \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\xi \bigcap E_\chi\,(\mathsf{h}^n(I))\Big) && = \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(E_\xi(\mathsf{h}^n(I)) \bigcap E_\chi(\mathsf{h}^n(I))\Big) && = \\
& && \textit{By Proposition 7.1 (2)} \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{p,\ldots,1}_{r_p,\ldots,r_1}\Big(\mathsf{z}^{r_1,\ldots,r_p}_{1,\ldots,p}(E_\xi(\mathsf{h}^n(I)))\Big) \bigcap \\
&\qquad \mathsf{z}^{q,\ldots,1}_{s_q,\ldots,s_1}\Big(\mathsf{z}^{s_1,\ldots,s_q}_{1,\ldots,q}(E_\chi(\mathsf{h}^n(I)))\Big)\Big) && = \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{p,\ldots,1}_{r_p,\ldots,r_1}\Big(\mathsf{h}^n(\xi^I)\Big) \bigcap \mathsf{z}^{q,\ldots,1}_{s_q,\ldots,s_1}\Big(\mathsf{h}^n(\chi^I)\Big)\Big) && = \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big( \\
&\qquad \mathsf{z}^{p,\ldots,1}_{r_p,\ldots,r_1}\Big(\mathsf{h}^n(\{\nu(x_{r_1}),\ldots,\nu(x_{r_p}) : I \vDash_\nu \xi\})\Big) \bigcap \\
&\qquad \mathsf{z}^{q,\ldots,1}_{s_q,\ldots,s_1}\Big(\mathsf{h}^n(\{\nu(x_{s_1}),\ldots,\nu(x_{s_q}) : I \vDash_\nu \chi\})\Big) \\
&\qquad \Big) && = \dagger \\
& && \textit{By Proposition 7.1 (5)} \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big( \\
&\qquad \mathsf{z}^{p+q,\ldots,p+1,p,\ldots,1}_{s_q,\ldots,s_1,r_p,\ldots,r_1}\Big(\mathsf{h}^n(\{\nu(x_{r_p}),\ldots,\nu(x_{r_1}) : I \vDash_\nu \xi\})\Big) \bigcap \\
&\qquad \mathsf{z}^{q+p,\ldots,q+1,q,\ldots,1}_{r_p,\ldots,r_1,s_q,\ldots,s_1}\Big(\mathsf{h}^n(\{\nu(x_{s_1}),\ldots,\nu(x_{s_q}) : I \vDash_\nu \chi\})\Big) \\
&\qquad \Big) && = \dagger \\
&\mathsf{z}^{i_1,\ldots,i_k}_{1,\ldots,k}\Big(\mathsf{z}^{k,\ldots,1}_{i_k,\ldots,i_1}\Big(\mathsf{h}^n(\{\nu(x_{i_1}),\ldots,\nu(x_{i_k}) : I \vDash_\nu \xi \wedge \chi\})\Big)\Big) && = \\
& && \textit{By Proposition 7.1 (2)} \\
&\mathsf{h}^n(\{\nu(x_{i_1}),\ldots,\nu(x_{i_k}) : I \vDash_\nu \xi \wedge \chi\} && = \\
&\mathsf{h}^n((\xi \wedge \chi)^I) && = \\
&\mathsf{h}^n(\psi^I).
\end{aligned}$$

[12]The last assumption is needed in steps †

- $\psi = \exists x_{i_j} \xi$, with $free(\xi) = \{x_{i_1}, \dots, x_{i_j}, \dots, x_{i_k}\}$. Let

$$
\begin{array}{rcl}
\{i'_1, \dots, i'_{k-1}\} & = & \{i_1, \dots, i_j, \dots, i_k\} \setminus \{i_j\} \\
\{r_1, \dots, r_{n-k}\} & = & \{1, \dots, n\} \setminus \{i_1, \dots, i_j, \dots, i_k\} \\
\{r'_1, \dots, r'_{n-k+1}\} & = & \{r_1, \dots, r_{n-k}\} \cup \{i_j\}
\end{array}
$$

We assume wlog that $k < n$. Let $E_\psi = \mathsf{c}_{i_j}(E_\xi)$. The inductive hypothesis is

$$\mathsf{z}^{i_1,\dots,i_k}_{1,\dots,k}\Big(E_\xi(\mathsf{h}^n(I))\Big) = \mathsf{h}^n(\xi^I).$$

We have

$$
\begin{array}{ll}
\mathsf{z}^{i'_1,\dots,i'_{k-1}}_{1,\dots,k-1}\Big(E_\psi(\mathsf{h}^n(I))\Big) & = \\
\mathsf{z}^{i'_1,\dots,i'_{k-1}}_{1,\dots,k-1}\Big(\mathsf{c}_{i_j}(E_\xi(\mathsf{h}^n(I)))\Big) & = \\
 & \textit{By Prop. 7.1 (3)} \\
\mathsf{z}^{i'_1,\dots,i'_{k-1}}_{1,\dots,k-1}\Big(\mathsf{c}_{i_j}(\mathsf{z}^{k,\dots,1}_{i_k,\dots,i_1}(\mathsf{z}^{i_1,\dots,i_k}_{1,\dots,k}(E_\xi(\mathsf{h}^n(I)))))\Big) & = \\
\mathsf{z}^{i'_1,\dots,i'_{k-1}}_{1,\dots,k-1}\Big(\mathsf{c}_{i_j}(\mathsf{z}^{k,\dots,1}_{i_k,\dots,i_1}(\mathsf{h}^n(\xi^I)))\Big) & = \\
 & \textit{By Prop. 7.1 (3)} \\
\mathsf{z}^{i'_1,\dots,i'_{k-1}}_{1,\dots,k-1}\Big(\mathsf{z}^{k,\dots,1}_{i_k,\dots,i_1}(\mathsf{c}_j(\mathsf{h}^n(\xi^I)))\Big) & = \\
\mathsf{z}^{i_1,\dots,i_{j-1},i_{j+1},\dots,i_k}_{1,\dots,j-1,j,\dots,k-1}\Big(\mathsf{z}^{k,\dots,j,j-1,\dots,1}_{i_k,\dots,i_j,i_{j-1},\dots,i_1}(\mathsf{c}_j(\mathsf{h}^n(\xi^I)))\Big) & = \\
\mathsf{z}^{i_1,\dots,i_{j-1}}_{1,\dots,j-1} \circ \mathsf{z}^{i_{j+1},\dots,i_k}_{j,\dots,k-1}\Big(\mathsf{z}^{k,\dots,j+1,j}_{i_k,\dots,i_{j+1},i_j} \circ (\mathsf{z}^{j-1,\dots,1}_{i_{j-1},\dots,i_1}\mathsf{c}_j(\mathsf{h}^n(\xi^I)))\Big) & = \\
 & \textit{By Prop. 7.2} \\
\mathsf{z}^{1,\dots,j-1,i_j,j,\dots,k-1}_{1,\dots,j-1,j,j+1,\dots,k}(\mathsf{c}_j(\mathsf{h}^n(\xi^I))) & = \\
 & \textit{By Prop. 7.1 (3)} \\
\mathsf{c}_{i_j}(\mathsf{z}^{1,\dots,j-1,i_j,j,\dots,k-1}_{1,\dots,j-1,j,j+1,\dots,k}(\mathsf{h}^n(\xi^I))) & = \\
\mathsf{c}_{i_j}(\mathsf{z}^{1,\dots,j-1,i_j,j,\dots,k-1}_{1,\dots,j-1,j,j+1,\dots,k}(\mathsf{h}^n(\{(\nu(x_{i_1}), \dots, \nu(x_{i_j}), \dots, \nu(x_{i_k})) : I \vDash_\nu \xi\}))) & = \\
\mathsf{c}_{i_j}(\mathsf{z}^{1,\dots,j-1,i_j,j,\dots,k-1}_{1,\dots,j-1,j,j+1,\dots,k}( & \\
\quad \{(\nu(x_{i_1}), \dots, \nu(x_{i_j}), \dots, \nu(x_{i_k}), \nu(x_{r_1}), \dots, \nu(x_{r_{n-k}})) : I \vDash_\nu \xi\})) & = \\
\mathsf{c}_{i_j}(\{(\nu(x_{i'_1}), \dots, \nu(x_{i'_{k-1}}), \nu(x_{r'_1}), \dots, \nu(x_{i_j}), \dots, \dots, \nu(x_{r'_{n-k+1}})) : I \vDash_\nu \xi\}) & = \\
\bigcup_{a \in D}\{(\nu(x_{i'_1}), \dots, \nu(x_{i'_k}), \nu(x_{r'_1}), \dots, \nu(x_{i_j}), \dots \nu(x_{r'_{n-k+1}}),) : I \vDash_{\nu_{(i_j/a)}} \xi\} & = \\
\{(\nu(x_{i'_1}), \dots, \nu(x_{i'_k}), \nu(x_{r'_1}), \dots, \nu(x_{i_j}), \dots, \nu(x_{r'_{n-k+1}})) : I \vDash_\nu \exists x_{i_j}\xi\} & = \\
\mathsf{h}^n(\{(\nu(x_{i'_1}), \dots, \nu(x_{i'_{k-1}})) : I \vDash_\nu \exists x_{i_j}\xi\}) & = \\
\mathsf{h}^n(\xi^I). &
\end{array}
$$

□

**Theorem 2.6.** For every CA$_n$ expression $E$ there is an FO$_n$ formula $\varphi_E$, such that

$$\varphi_E^I = E(\mathsf{h}^n(I)),$$

for all instances $I$. ◂

**Proof:**

We do a structural induction

- $E = \mathsf{C}_p$. Then $\varphi_E = R_p(x_1, \ldots, x_k) \wedge \bigwedge_{r \in \{k+1,\ldots,n\}} (x_r \approx x_r)$, where $k = ar(R_p)$. Clearly

$$\begin{array}{ll} \varphi_E^I & = \\ \{(\nu(x_1), \ldots, \nu(x_k), \nu(x_{k+1}), \ldots, \nu(x_n)) : I \vDash_\nu R_p(x_1, \ldots, x_k)\} & = \\ R_p^I \times \mathbb{D}^{n-k} & = \\ \mathsf{C}_p(\mathsf{h}_n(I)) & = \\ E(\mathsf{h}^n(I)). & \end{array}$$

- $E = \mathsf{d}_{ij}$. Then $\varphi_E = (x_i \approx x_j) \wedge \bigwedge_{r \in \{1,\ldots,n\} \setminus \{i,j\}} (x_r \approx x_r)$. We have

$$\begin{array}{ll} \varphi_E^I & = \\ \{(\nu(x_1), \ldots, \nu(x_i), \ldots, \nu(x_j), \ldots, \nu(x_n)) : I \vDash_\nu (x_i \approx x_j)\} & = \\ \{t \in \mathbb{D}^n : t(i) = t(j)\} & = \\ \mathsf{d}_{ij} & = \\ E(\mathsf{h}^n(I)). & \end{array}$$

- $E = F_1 \bigcap F_2$. Then $\varphi_E = \varphi_{F_1} \wedge \varphi_{F_2}$, and the inductive hypothesis is

$$\begin{array}{lll} \varphi_{F_1}^I & = & F_1(\mathsf{h}^n(I)) \\ \varphi_{F_2}^I & = & F_2(\mathsf{h}^n(I)) \end{array}$$

Then,

$$\begin{array}{ll} \varphi_E^I & = \\ (\varphi_{F_1} \wedge \psi_{F_2})^I & = \\ \{(\nu(x_1), \ldots, \nu(x_n)) : I \vDash_\nu \varphi_{F_1} \wedge \psi_{F_2}\} & = \\ \{(\nu(x_1), \ldots, \nu(x_n)) : I \vDash_\nu \varphi_{F_1}\} \cap & \\ \{(\nu(x_1), \ldots, \nu(x_n)) : I \vDash_\nu \xi_{F_2}\} & = \\ \varphi_{F_1}^I \cap \xi_{F_2}^I & = \\ F_1(\mathsf{h}^n(I)) \bigcap F_2(\mathsf{h}^n(I)) & = \\ F_1 \bigcap F_2\ (\mathsf{h}^n(I)) & = \\ E(\mathsf{h}^n(I)). & \end{array}$$

- $E = \overline{F}$, where Then $\varphi_E = \neg\varphi_F$, and the inductive hypothesis is $\varphi_F^I = F(\mathsf{h}^n(I))$. We have

$$\begin{aligned}
&\varphi_E^I &=\\
&\neg\varphi_F^I &=\\
&\overline{\varphi_F^I} &=\\
&\overline{F(\mathsf{h}^n(I))} &=\\
&E(\mathsf{h}^n(I)).
\end{aligned}$$

- $E = \mathsf{c}_i(F)$, Then $\varphi_E = (\exists x_i\, \varphi_F) \wedge (x_i \approx x_i)$. The inductive hypothesis is $\varphi_F^I = F(\mathsf{h}^n(I))$. We have

$$\begin{aligned}
&\varphi_E^I &=\\
&\{(\nu(x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu (\exists x_i\, \varphi_F) \wedge (x_i \approx x_i)\} &=\\
&\{(\nu(x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu (\exists x_i\, \varphi_F)\} \;\cap\\
&\{(\nu(x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu (x_i \approx x_i)\} &=\\
&\{(\nu(x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu (\exists x_i\, \varphi_F)\} \;\cap\; \mathbb{D}^n &=\\
&\{(\nu(x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu (\exists x_i\, \varphi_F)\} &=\\
&\textstyle\bigcup_{a\in\mathbb{D}} \{(\nu((x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_{\nu(i/a)} \varphi_F\} &=\\
&\mathsf{c}_i(\{(\nu((x_1),\ldots,\nu(x_i),\ldots,\nu(x_n)) \,:\, I \models_\nu \varphi_F\}) &=\\
&\mathsf{c}_i(\varphi_F^I) &=\\
&\mathsf{c}_i(F(\mathsf{h}^n(I))) &=\\
&E\,(\mathsf{h}^n(I)).
\end{aligned}$$

□