Toward Model Theory with Data Values

Mikołaj Bojańczyk and Thomas Place*

University of Warsaw

Abstract. We define a variant of first-order logic that deals with data words, data trees, data graphs etc. The definition of the logic is based on Fraenkel-Mostowski sets (FM sets, also known as nominal sets). The key idea is that we allow infinite disjunction (and conjunction), as long as the set of disjuncts (conjunct) is finite modulo renaming of data values. We study model theory for this logic; in particular we prove that the infinite disjunction can be eliminated from formulas.

1 Introduction

This paper uses Fraenkel-Mostowski sets (FM sets, also known as nominal sets) to study logics that describe properties of objects such as data words, data trees, or data graphs.

Suppose that $\mathbb{D}$ is an infinite set of *data values*, also called *atoms* or *ur-elements*, whose elements can only be compared for equality. A *data word* is a word (trees and graphs can also be considered, of course) whose positions are labelled by an alphabet that is not necessarily finite, but which can refer to data values in a finite way, such as in the following examples of alphabets.

$$\mathbb{D} \qquad \{0,1\} \times \mathbb{D} \qquad (\mathbb{D}^2 \cup \mathbb{D}) \qquad \{0,1\} \times \{\{c,d\} : c,d \in \mathbb{D}\}.$$

The statement "data values can only be compared for equality" is formalized by saying that properties of data words should be invariant under the action of the group of bijections of $\mathbb{D}$. The statement "refer to data values in a finite way" is formalized by saying that the alphabet contains finitely many elements, modulo bijections of data values. For instance, modulo bijections, the set $\mathbb{D}^2 \cup \mathbb{D}$ has three elements, which look like this: $(d,e), (d,d)$ and d.

Properties that are invariant under the action of the group of bijections include:

1. Data words over the alphabet $\mathbb{D}$ where all positions have different labels.
2. Data words over the alphabet $\mathbb{D}$ with at least six distinct letters.
3. Graphs with edges labelled by $\mathbb{D}^2$ where for each vertex, all outgoing edges have the same data value on the first coordinate.

There is a more relaxed notion of invariance: a property is called *finitely supported* if there is a finite set of data values $C \subseteq \mathbb{D}$, such that the property is invariant under the action of permutations that preserve C. The set C is called the *support*. For instance, if we choose some two elements $c, d \in \mathbb{D}$, then

* Both authors supported by ERC Starting Grant "Sosna". A full version of this paper can be found at `www.mimuw.edu.pl/~bojan`.

4. Data words over alphabet $\mathbb{D}$ which begin with c and end with d.

is a finitely supported property, namely supported by $C = \{c, d\}$.
To give examples of properties that are not finitely supported, one needs additional assumptions on $\mathbb{D}$. For instance, if we assume that $\mathbb{D}$ is the natural numbers, then "words in $\mathbb{D}^*$ which contain only even numbers" is not a finitely supported property of data words.
The notion of finitely supported sets is the cornerstone of "permutation models" of set theory, which were studied by logicians such as Fraenkel and Mostowski starting in the 1920's. Permutation models were rediscovered, under the name "nominal sets", by Gabbay and Pitts in [3], see also [4], as an elegant approach to deal with binding and fresh names in the syntax of programming languages and logical formulas. When dealing with syntax, one thinks of data values as being variable names. Finally, these sets were rediscovered by the automata community, as an approach to describing languages of data words [2].[1]

Logic on data words and data trees. The direct predecessor of this paper is [2], which uses FM sets (under the name nominal sets) to talk about automata on data words. In the present paper, we use FM sets to talk about logics on data words (and more general structures). We define:

- A notion of *FM relational structure*. This notion generalizes data words, data trees, data graphs, etc. One can apply a permutation of data values to an FM relational structure, and get another FM relational structure.
- A notion of *FM first-order logic*. The formulas are evaluated in FM relational structures. The formulas form an FM set. The previously stated examples of properties of data words and data graphs are definable in the logic, including example 4.

Logics for data words have been extensively studied in the special case of data words and data trees with alphabets of the form $A \times \mathbb{D}$, where A is a finite set. In this special case, the approach of [6] is to use: a binary predicate $x \sim y$ which says that two positions carry the same data value; as well as a unary predicate $a(x)$ for each $a \in A$. The satisfiability problem for the logic is undecidable for most variants, see [6]. In the special case of alphabets $A \times \mathbb{D}$, our abstract definition of FM first-order logic coincides with the existing definition.
Even for words, the choice of logic is not obvious for some alphabets. Consider the alphabet "sets of data values of size at most 3". A natural predicate would be $x \subseteq y$, saying that the set in the label of x is a subset of the set in the label

[1] There are two names for the sets that can be used: "FM sets" as in mathematical logic, or "nominal sets" as in the study of name binding. In this paper, we decided to use the name "FM sets". The main reason is that our application of Frankel-Mostowski set theory is not principally concerned with the use of names and their binding. An additional reason is that, like in Fraenkel-Mostwoski set theory, but unlike in the study of name binding, we are often interested in data values with additional structure, such as a linear order.

of y. Another kind of predicate, not definable in terms of $x \subseteq y$, could be

$$|x_1 \cup \cdots \cup x_n| = k \qquad \text{for } n, k \in \mathbb{N}.$$

Which predicates should be allowed in the logic? Our definition implies that they are all allowed. We do not address the question of a minimal choice of predicates, i.e. which predicates can be defined in terms of others.

Parse trees. On a definitional level, the principal idea in this paper is to allow parse trees of formulas where the branching degree is not finite, but finite modulo bijections of data values (we call this orbit-finite branching). In normal sets, the parse tree of an expression (a formula of first-order logic, a regular expression, an arithmetic expression, etc.) is a finite tree. In FM sets, one can have a more relaxed parse tree: for each node, the set of child subtrees is only required to be finite modulo bijections of data values[2]. For instance, if for each data value $d \in \mathbb{D}$ we have a formula φ_d, and the function $d \mapsto \varphi_d$ is finitely-supported, then it makes sense to consider the infinite disjunction $\bigvee_{d \in \mathbb{D}} \varphi_d$. On a technical level, the main contribution of this paper is Theorem 5.2 which says that the infinite disjunction can be eliminated from formulas.

Related work. A logic for nominal sets, called nominal logic, was studied by Pitts in [5]. Nominal logic and the logic from this paper have different goals: nominal logic is designed to axiomatise nominal sets, while the formulas in this paper are used to define languages of data words and similar objects. Also, the logics are defined differently: the formulas and models for nominal logic are defined in normal set theory; while the formulas and models in this paper are defined inside FM set theory[3]. Finally, the principal technical result of this paper is elimination of infinite disjunction, this result cannot be even stated in the language of [5].

Acknowledgement. We would like to thank Nathanaël Fijalkow, Bartek Klin, and the anonymous referees for their comments and suggestions.

2 Preliminaries

Data symmetry. The notion of FM sets is parametrized by a set of *data values* $\mathbb{D}$, and a group G of bijections on $\mathbb{D}$. The group G need not contain all bijections of $\mathbb{D}$. The idea is that $\mathbb{D}$ has some structure, and G contains the structure-preserving bijections. The pair $(\mathbb{D}, G)$ is called a *data symmetry*. In this paper, we use the following data symmetries:

[2] This appears already explicitly in [1], where terms of λ-calculus have orbit-finitely branching parse-trees. Implicitly, the idea goes back the work of Gabbay and Pitts, where the whole point of nominal sets was to model the use of binding.

[3] One could say that our logic is an internal logic for FM sets, while the logic of [5] is external.

– The set $\mathbb{D}$ is empty, and G has only the identity element. We call this the *classical symmetry*. FM sets in the classical symmetry are normal sets.
– The set $\mathbb{D}$ is a countable set, say the natural numbers. The group G consists of all bijections on $\mathbb{D}$. We call this the *equality symmetry*. FM sets in the equality symmetry are the same thing as nominal sets [3, 4].
– The set $\mathbb{D}$ is the vertices of the undirected countable homogeneous graph (also called the Rado graph), and the group G is the group of automorphisms of this graph. We call this the *graph symmetry*.

FM set. Consider first the *cumulative hierarchy* of sets with data values, which is a hierarchy of sets indexed by ordinal numbers and defined as follows. The empty set is the unique set of rank 0. A set of rank α is any set whose elements are sets of rank smaller than α, or data values. A permutation π of data values can be applied to a set X in the hierarchy, by renaming the elements of X, and the elements of elements of X, and so on. The resulting set, which has the same rank, is denoted by $X \cdot \pi$.
A set C of data values is said to be a *support* of a set X in the cumulative hierarchy if $X \cdot \pi = X \cdot \sigma$ holds for every permutations π, σ in the group from the data symmetry which agree on elements of C. A set is called *finitely supported* if it has some finite support. We use the name *FM set* for a set in the cumulative hierarchy which is hereditarily finitely supported, which means that it is finitely supported, the sets in it are finitely supported, and so on[4].
The support of an FM set is not unique, e.g. supports are closed under adding data values. A set with empty support is called *equivariant*.
In many respects, FM sets behave like normal sets. For instance, if X, Y are FM sets, then $X \times Y$, $X \cup Y$, X^* and the finite powerset of X are all FM sets. Another example is the family of subsets of X that have finite supports. The appropriate notion of a function between FM sets X and Y is that of a *finitely supported function*, which is a function from X to Y whose graph is an FM set. Observe that FM sets in the classical symmetry are simply sets (equipped with the only possible action). Therefore the classical symmetry corresponds to classical set theory, without data values.

Orbit-finite FM sets. Suppose that X is an FM set. For a finite set $C \subseteq \mathbb{D}$, define G_C to be the subgroup of G which contains the bijections that are the identity on the set C and the C-orbit of an element $x \in X$ to be the set $\{x \cdot \pi : \pi \in G_C\}$. If C supports X, then the C-orbits form a partition of X. The set X is called orbit-finite if it the partition into C-orbits has finitely parts, for some C which supports X. For some data symmetries, including the classical, equality and graph symmetries discussed in this paper, the notion of orbit-finiteness does not depend on the choice of support [1]. In other words, for these data symmetries, if two sets C and D support an FM set X, then X has finitely many C-orbits if and only if it has finitely many D-orbits.
In this paper, we are mostly interested in FM sets that are orbit-finite.

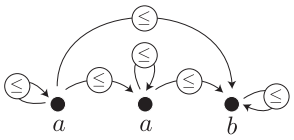
[4] The definition here is based on Definition 10.6 in [4], except that we use the name *FM set* for what [4] calls elements of $\mathcal{HFS}$.

3 Relational Structures

The discussion in this section – and the next Section 4 – makes sense in any data symmetry. Fix some data symmetry $(\mathbb{D}, G)$ for this section and the next.
One of the key ideas of finite model theory in computer science is that a combinatorial object, such as a word, tree, or graph, can be treated as a model for a logical formula. For instance, in the case of words over an alphabet $\{a, b\}$, a word with n positions can be interpreted as a relational structure where the domain is the set of positions $\{1, \ldots, n\}$, there are two unary predicates $a(x)$ and $b(x)$ for labels, and there is a binary predicate $x \leq y$ for the order on positions. Using this interpretation, one can define properties of words using first-order logic, e.g. the set of words that end with b is defined by the formula

$$\forall x \exists y \quad x \leq y \wedge b(y).$$

The goal of this paper is to define a similar notion of logic for combinatorial objects that contain data values. In particular, our definition should cover data words and data trees.
In standard sets, not FM sets, a relational structure can be seen as a hyper-edge colored directed hypergraph. For instance, the relational structure corresponding to the word aab is the following hypergraph.



We adapt this definition to FM sets as follows. For instance a binary predicate will not just say yes/no to each directed edge, but it can also color the edge, e.g. by a data value.
An *FM predicate* R consists of an orbit-finite FM set $\text{colors}(R)$ and a natural number $\text{arity}(R)$. An *FM signature* Σ is a finite set of FM predicates. An *FM relational structure* $\mathfrak{A}$ over Σ consists of:

- A set $\text{dom}(\mathfrak{A})$, called the *domain* of the structure, which is a normal set (meaning an FM set where the action of G is trivial).
- For every predicate R in the signature, a finitely supported partial function

$$R^{\mathfrak{A}} : \text{dom}(\mathfrak{A})^{\text{arity}(R)} \dot{\to} \text{colors}(R),$$

  called the *interpretation of* R.

For a fixed FM signature Σ, the set of FM relational structures over Σ is itself a FM set, because an FM relational structure is nothing other than a domain (which has empty support, since it is equipped with the trivial action) and a finite tuple of finitely-supported partial functions[5]. Therefore, if $\mathfrak{A}$ is a FM relational

[5] Formally speaking, this is an FM class, because all the domains do not form a set.

structure and $\pi \in G$, then also $\mathfrak{A} \cdot \pi$ is a FM relational structure. Both $\mathfrak{A}$ and $\mathfrak{A} \cdot \pi$ have the same domains, only different interpretations. If R is a predicate of arity n, and $x_1, \ldots, x_n$ are in the domain of $\mathfrak{A}$, then

$$R^{\mathfrak{A}\cdot\pi}(x_1, \ldots, x_n) = (R^{\mathfrak{A}}(x_1, \ldots, x_n)) \cdot \pi.$$

In particular, either both sides of the equality above are defined, or neither are (recall that interpretations are partial functions.)
An FM relational structure is called finite if its domain is finite. In such a case, an interpretation is a finite tuple of colors. A tuple of objects taken from an FM set necessarily has finite support, and therefore in the case of finite relational structures, the requirement on finitely supported interpretations is redundant.

Example 3.1. Data words can be modeled as FM relational structures. We use the name *FM alphabet* for any orbit-finite set A. To an FM alphabet A, we associate an FM signature Σ_A with two predicates:

- The alphabet predicate R_A, which has arity 1 and colors A.
- The order predicate $R_<$, which has arity 2 and only one color $\{<\}$.

For a data word $w = a_1 \cdots a_n \in A^*$, we define a corresponding FM relational structure $\mathfrak{A}_w$ over the signature Σ_A as follows. The domain $dom(\mathfrak{A}_w)$ is the set $\{1, \ldots, n\}$ of positions. The interpretation of the alphabet predicate maps each position to its label, and the interpretation of the order predicate maps a pair (i, j) to $<$ if and only if $i < j$.

Example 3.2. Edge-labelled directed data graphs can be modeled as FM relational structures. To an FM alphabet A, we associate an FM signature Σ_A with one predicate R_A with arity 2 and colors A, called the edge label predicate. A structure over this FM signature describes a directed graph, where edges are labelled by A. Because the interpretation is a partial function, every ordered pair of nodes is connected by zero or one edge.

4 Logic

In this section, we define a variant of first-order logic which is used to define properties of FM relational structures. Before giving the actual definition, we enumerate the postulates it should satisfy:

1. The set of formulas is itself an FM set, and the satisfaction relation is equivariant. That is

$$\mathfrak{A} \models \varphi \quad \text{iff} \quad \mathfrak{A} \cdot \pi \models \varphi \cdot \pi$$

   holds for every $\pi \in G$, FM relational structure $\mathfrak{A}$ and formula φ.
2. Orbit-finite disjunction is allowed. That is, if Γ is an orbit-finite FM set of formulas, then also $\bigvee \Gamma$ is a formula.

Below, in Section 4.1, we give a definition which satisfies the above postulates.

4.1 Definition of FM first-order logic

To choose the predicates for our logic, we use a semantic approach: every isomorphism-closed property of a tuple of elements is going to be a predicate. An isomorphism between two FM relational structures $\mathfrak{A}$ and $\mathfrak{B}$ is a finitely supported bijection

$$f : \mathrm{dom}(\mathfrak{A}) \to \mathrm{dom}(\mathfrak{A})$$

such that for every k-ary predicate R in the signature, we have

$$R^{\mathfrak{A}}(a_1, \ldots, a_k) = R^{\mathfrak{B}}(f(a_1), \ldots, f(a_k)) \qquad \text{for every } a_1, \ldots, a_k \in \mathrm{dom}(\mathfrak{A}).$$

Example 4.1. Consider the equality symmetry, and a signature with one unary predicate P, whose colors are $\mathbb{D}$. Suppose that $1, 2$ are data values. Let $\mathfrak{A}$ be a structure whose domain is $\{1\}$, and where the interpretation is $P^{\mathfrak{A}}(1) = 1$. Let $\mathfrak{B}$ be a structure whose domain is $\{2\}$, and where the interpretation is $P^{\mathfrak{B}}(2) = 1$. Then $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic.

An *atomic type of arity n* is (the isomorphism type of) a structure $\mathfrak{A}$, together with an n-tuple of elements, such that every element of the domain of $\mathfrak{A}$ appears in the tuple. The domain of the atomic type has at most n elements, but might be smaller if the tuple contains repetitions. We write $\mathrm{atoms}_n(\Sigma)$ for the set of atomic types of arity n. If $\mathfrak{A}$ is a structure, and $\bar{a}$ is a (possibly repeating) tuple of elements in $\mathrm{dom}(\mathfrak{A})$, then we define $\mathfrak{A}|\bar{a}$ to be the atomic type obtained from $\mathfrak{A}$ by only keeping the elements from $\bar{a}$.

Fact 1 *In the classical, equality and graph symmetries, the set* $\mathrm{atoms}_n(\Sigma)$ *is orbit-finite.*

A *basic type of arity n* is defined to be any finitely supported subset of $\mathrm{atoms}_n(\Sigma)$.

Example 4.2. Consider the FM alphabet $\binom{\mathbb{D}}{2}$, which is defined to be the family of two-element subsets of $\mathbb{D}$. Consider data words over this alphabet, as in Example 3.1. A basic type $\mathcal{B}$ of arity 2 could say that that the set in the label of the first distinguished position has non-empty intersection with the set in the label in the second distinguished position. This basic type is not only finitely supported, but also equivariant.

Example 4.3. This example also concerns data words over $\binom{\mathbb{D}}{2}$. In this example, as in subsequent examples, we assume that the data values are natural numbers. A non-equivariant basic type $\mathcal{B}_9$ of arity 2 says that the sets in the label of the first and second distinguished position both contain the data value $9 \in \mathbb{D}$.

We define *FM first-order logic for a relational signature Σ* as follows. As predicates, we use basic types in the following sense: a basic type $\mathcal{B}$ of arity n is a predicate of arity n, with the semantics

$$\mathfrak{A}, \bar{a} \models \mathcal{B}(x_1, \ldots, x_n) \qquad \text{iff} \qquad \mathfrak{A}|\bar{a} \in \mathcal{B}.$$

Furthermore, formulas can use boolean combinations $\{\vee, \wedge, \neg\}$ as well as quantifiers $\{\forall, \exists\}$. We will add one more connective, but to define this connective we need to discuss the action of G on formulas. When applying a permutation $\pi \in G$ to a formula φ, the structure of the formula, the connectives $\vee, \wedge, \neg, \forall, \exists$ as well as the variables are not changed. The only thing that changes is the basic types: a set of atomic types $\mathcal{B}$ is mapped to the set $\mathcal{B} \cdot \pi$.

Example 4.4. Consider the basic type $\mathcal{B}_9$ from Example 4.3, and the formula

$$\exists x \exists y \; x \neq y \wedge \mathcal{B}_9(x, y).$$

This formula, call it φ_9, says that the data value 9 appears in the label of at least two positions. Consider a permutation $\pi \in G$, which maps 9 to 8. Then the formula $\varphi_9 \cdot \pi$ says that the data value 8 appears in the label of at least two positions.

It is not difficult to see that every formula has finite support. The reason is that every formula uses a finite number of basic types, and each basic type has finite support by definition.
We now define the remaining connective, which is called *orbit-finite disjunction*. Consider an orbit-finite FM set of already defined formulas Γ. We allow a disjuction over this set $\bigvee \Gamma$, with the expected semantics. Orbit-finite disjunction is the last connective of the logic, and the definition of FM first-order logic is now complete.

Example 4.5. Consider the formula φ_9 in Example 4.4. This formula can be defined for any data value d, not just 9, and it is easy to see that the set $\{\varphi_d : d \in \mathbb{D}\}$ is an orbit-finite FM set of formulas. Therefore, we can use the orbit-finite disjunction

$$\bigvee \{\varphi_d : d \in \mathbb{D}\} \qquad \text{also written as} \qquad \bigvee_{d \in \mathbb{D}} \varphi_d.$$

The disjunction above says that some data value appears in the label of at least two positions. Observe that the above formula can be expressed, without orbit-finite disjunction, by using the predicate $\mathcal{B}$ from Example 4.2:

$$\exists x \exists y \; x \neq y \wedge \mathcal{B}(x, y).$$

Example 4.6. In the previous case, the set $\{\varphi_d : d \in \mathbb{D}\}$ was equivariant. One can also use non-equivariant sets, such as

$$\bigvee_{d \in \mathbb{D} - \{9\}} \varphi_d.$$

Non-equivariant sets are useful for nesting formulas, e.g.

$$\bigwedge_{e \in \mathbb{D}} \bigvee_{d \in \mathbb{D} - \{d\}} \varphi_d.$$

The formula above says that there are two data values that appear in the label of at least two positions.

5 Elimination of Orbit-Finite Disjunction

Recall that in Example 4.5, we were able to eliminate orbit-finite disjunction. The technique was to push the disjunction into the basic types. This technique can fail, e.g. in the graph symmetry, as shown by the following theorem.

Theorem 5.1. *Consider the graph symmetry. Let $L \subseteq \mathbb{D}^*$ be the set of words $d_1 \cdots d_n$ such that n is even, all letters are distinct, and for every $i, j \in \{1, \ldots, n\}$ there is no graph edge from d_i to d_j. This set is definable in FM first-order logic with orbit-finite disjunction, but not by a formula without orbit-finite disjunction.*

The main technical result of this paper is the following theorem, which says that orbit-finite disjunction can be eliminated in the equality symmetry.

Theorem 5.2. *Consider the equality symmetry. Every formula of FM first-order logic is equivalent to a formula that does not use orbit-finite disjunction.*

The proof can be found in the full version of the paper. It uses a notion of functionality, which we believe to be of independent interest, and which is discussed in Section 6. When the colors used by the predicates are just the set $\mathbb{D}$ of data values, Theorem 5.2 is straightforward. The main difficulty is dealing with non-standard sets of colors. We illustrate this with the following examples which show Theorem 5.2 in action for increasingly complicated sets of colors.

Example 5.3. Consider data words over the alphabet $A = \{a, b\} \times \mathbb{D}$. If a position carries the letter $(\sigma, d) \in A \times \mathbb{D}$, then we say that that it has label σ and data value d. Consider the language: "some data value appears only on positions with label a". This language is expressed by the formula

$$\bigvee_d \exists x \; d(x) \wedge \forall x \; d(x) \Rightarrow a(x),$$

where $d(x)$ is the basic type which holds for positions where the data value is d. The orbit-finite disjunction in this formula can be eliminated by encoding the data value d by a position y:

$$\exists y \forall x \; x \sim y \Rightarrow a(x),$$

where $x \sim y$ says that x and y carry the same data value, also a basic type. The same trick, of encoding a data value by a position, works for every formula over this alphabet.

Example 5.4. For $k \in \mathbb{N}$, consider the alphabet

$$B_k = \{a, b\} \times P_{\leq k}(\mathbb{D}),$$

which is like in the previous example, except that the second coordinate is now not a single data value, but an (unordered) set of at most k data values. Let us

first study the case of $k = 2$. Consider the language "some data value appears (in the set) only on positions with label a".

$$\bigvee_d \exists x\ (d \in x) \wedge \forall x\ (d \in x) \Rightarrow a(x),$$

where $d \in x$ is a unary basic type, which selects positions that contain d. Let us use the name *witness* for a data value d which satisfies the formula $\forall x\ (d \in x) \Rightarrow a(x)$. The trick from Example 5.3 was to encode a witness by a position. This trick does not always work for the alphabet B. Consider for instance the word

$$(a, \{1\})(b, \{2, 3\})(a, \{1, 3\})(a, \{2, 4\})(a, \{5, 6\})(a, \{5, 6\})$$

The witnesses are the data values $1, 4, 5, 6$. The witness 1 can be defined in terms of the first position: it is the unique data value in the set $\{1\}$, which appears in the first position. The witness 4 can be defined in terms of two positions: it is the unique data value which appears in the set $\{2, 4\}$ on the fourth position but not in the set $\{2, 3\}$ on the second position. Finally, witnesses 5 and 6 can only be defined as a set of size two; they cannot be distinguished. One can see that these three types of witnesses are the only possible ones for $k = 2$. All of these three types can be captured by the following formula, which does not use orbit-finite disjunction:

$$\exists y_1 \exists y_2 \forall x (\emptyset \subsetneq y_1 \cap y_2 \subseteq x) \Rightarrow a(x),$$

where $\emptyset \subsetneq y_1 \cap y_2 \subseteq x$ is a basic type, which says that the intersection of the sets of data values in y_1 and y_2 is non-empty and included in the set in x.
For $k > 2$, one needs more complicated expressions to define some data values, such as: "the data value that appears in positions five, six and seven, but not eight and two". Also, one can have sets of up to k data values that cannot be distinguished from each other.

5.1 Standard data words

Consider the special case where the models are data words as in Example 3.1 and the alphabet is of the form $A_{\text{fin}} \times \mathbb{D}$, where A_{fin} is a finite set. As mentioned in the introduction, there is an established logic for words over this kind of alphabet, which has a predicate $x < y$ for the position order, a predicate $x \sim y$ for equal data values, and a label predicate $a(x)$ for every $a \in A_{\text{fin}}$ (note that this logic does not allow orbit-finite disjnuctions). A simple consequence of Theorem 5.2 is the following theorem:

Theorem 5.5. *Let A_{fin} be a finite set. Let L be an equivariant language over the alphabet $A_{\text{fin}} \times \mathbb{D}$. The following conditions are equivalent:*

1. *L is definable by a formula of FM first-order logic, possibly including orbit-finite disjunction.*
2. *L is definable by a formula of the standard first-order logic for data words, which has predicates for the position order $x < y$, equal data value $x \sim y$, and the labels $\{a(x)\}_{a \in A_{\text{fin}}}$.*

6 Functionality and locality

In this section we define a key concept for the proof of Theorem 5.2. Our proof technique is to encode data values in elements of the domain of the relational structure (which corresponds to positions in the case of data words). As illustrated in Example 5.4:

1. Sometimes, more than one element is needed to define a data value;
2. Sometimes, a data value can only be defined in combination with some indistinguishable other data value;
3. Sometimes, both problems above hold simultaneously.

This section is devoted to a study of how one can define a data value, or more generally an element of some orbit finite set, in terms of a relational structure.

Functionality. In normal sets, without data values and group actions, the expression "f is a function of g" makes sense only when both f and g are functions with a common domain. For instance, one can say "the area of a circle is a function of its radius", which is formalized as two functions on the domain of circles, the area and radius functions. Another example: "a person's taste in football is a function of their sympathy for Real Madrid".
With data values, the notion of functionality makes sense for arbitrary objects. Suppose that x is an FM set or a data value, likewise y. Let C be a finite set of data values. We say that *y is a C-supported function of x* if there is a C-supported function

$$f : X \to Y \qquad \text{such that } x \in X \text{ and } y \in Y$$

which maps x to y. In the spacial case of $C = \emptyset$, we say that y is an equivariant function of x. (In the classical symmetry, which corresponds to normal sets, one can always take $X = \{x\}$, $Y = \{y\}$. In this case, every y is an equivariant function of every x, which is why the definition is not interesting.)

Example 6.1. The data value 2 is an equivariant function of the three-letter data word 123. In this case X is the set of data words of length three, Y is $\mathbb{D}$, and f maps a word to its second letter (there are other choices for X, Y and f). The data value 2 is an equivariant function of $\{2\} \in P(\mathbb{D})$. The data value 2 is not an equivariant function of the set $\{1, 2, 3\} \in P(\mathbb{D})$, or of the empty set $\emptyset \in P(\mathbb{D})$. The data value 2 is a $\{2\}$-supported function, but not a $\{3\}$-supported function, of the data value 1.

We can now state the main theorem of this section which concerns the first issue in the list at the beginning of this section: more than one element might be needed to define a data value.

Theorem 6.2 (Local Functionality Theorem). *Let X be an orbit-finite FM set, and Σ an FM relational signature. Let C be a finite set of data values that supports X and Σ. There is some $k \in \mathbb{N}$ such that for every $x \in X$ and every FM relational structure $\mathfrak{A}$, the following conditions are equivalent*

– *x is a C-supported function of $\mathfrak{A}$;*
– *x is a C-supported function of $\mathfrak{A}|\bar{a}$, for some tuple $\bar{a} \in (\mathrm{dom}(\mathfrak{A}))^k$.*

The point of Theorem 6.2 is that the bound k depends on X, Σ and C, but not on $\mathfrak{A}$. When proving Theorem 5.2, we use this result in the following form: if a parameter $i \in I$ of an orbit-finite disjunction $\bigvee_{i \in I} \varphi_i$ is an equivariant function of a model, then it is an equivariant function of a small tuple $\bar{a}$, and the tuple can be captured using k existential quantifiers.

7 Conclusions

We have defined a notion of first-order logic for models that talk about data values. The main technical result is that orbit-finite disjunction can be eliminated in the equality symmetry. Possibilities of future work include:

– Using orbit-finite disjunction, one gets a natural notion of star-free languages of data words. Is this notion equivalent to FM first-order logic?
– Elimination of orbit-finite disjunction works in the equality symmetry, but not in the graph symmetry. In which symmetries does it work? We conjecture that it also works in the total order symmetry and the forest order symmetry, see [1]. We intend to investigate this issue further.
– Can one use the syntax of FM first-order logic to define new fragments of first-order logic on data words that have decidable satisfiability?

References

1. Mikolaj Bojanczyk, Laurent Braud, Bartek Klin, and Slawomir Lasota. Towards nominal computation. In *POPL*, pages 401–412, 2012.
2. Mikolaj Bojanczyk, Bartek Klin, and Slawomir Lasota. Automata with group actions. In *LICS*, pages 355–364, 2011.
3. M. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Asp. Comput.*, 13(3-5):341–363, 2002.
4. Murdoch James Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bulletin of Symbolic Logic*, 17(2):161–229, 2011.
5. Andrew M. Pitts. Nominal logic, a first order theory of names and binding. *Inf. Comput.*, 186(2):165–193, 2003.
6. Luc Segoufin. Automata and logics for words and trees over an infinite alphabet. In *CSL*, pages 41–57, 2006.

Part I of the Appendix.

Theorems 5.5 and 5.1

A Proof of Theorem 5.1

Theorem 5.1. *Consider the graph symmetry. Let $L \subseteq \mathbb{D}^*$ be the set of words $d_1 \cdots d_n$ such that n is even, all letters are distinct, and for every $i, j \in \{1, \ldots, n\}$ there is no graph edge from d_i to d_j. This set is definable in FM first-order logic with orbit-finite disjunction, but not by a formula without orbit-finite disjunction.*

We use the name *nonrepeating edge-free word* for a word $d_1 \cdots d_n$ where all letters are distinct, and for every $i, j \in \{1, \ldots, n\}$ there is no edge from d_i to d_j. Consider the following lemma:

Lemma A.1. *Over nonrepeating edge-free words:*

1. *FM first-order logic has the same expressive power as monadic second-order logic.*
2. *Every formula that does not use orbit-finite disjunction is equivalent to one that uses only the position order predicate, and not the label predicates.*

Theorem 5.1 immediately follows from Lemma A.1. By the first item, it follows that even length can be defined in FM first-order logic, by using the standard formula of MSO for even length. By the second item, it follows that even length cannot be defined in normal form, because a formula of first-order logic with only the order predicate cannot define even length.

Proof (of Lemma A.1). Consider the first item. For every finite set $d_1, \ldots, d_n$ of nodes in the random graph, one can use a single node e of the random graph to represent an arbitrary subset of $d_1, \ldots, d_n$. This is because for every subset $X \subseteq \{d_1, \ldots, d_n\}$ there is some $e \in \mathbb{D}$ such that

$$X = \{d_1, \ldots, d_n\} \cap \{d \in \mathbb{D} : \text{there is an edge from } d \text{ to } e\}.$$

Consider now the second item, which talks about formulas without orbit-finite disjunction. Consider a basic type

$$\mathcal{B}(x_1, \ldots, x_n).$$

Let C be the constants of $\mathcal{B}$, which is a finite set of data values. We claim that over nonrepeating edge-free words $\mathcal{B}$ is equivalent to a formula that uses the following finite set of predicates:

- A predicate for equality of positions (not of their labels).
- For each $c \in C$, a predicate which is true in positions whose labels are data values that have an edge to (respectively, from) c.
- For each $c \in C$, a predicate which is true in positions whose labels are c.

Indeed, consider a nonrepeating edge-free data word $d_1 \cdots d_k$ and two valuations

$$\eta_1, \eta_2 : \{x_1, \ldots, x_n\} \to \{1, \ldots, n\}.$$

We say that these valuations have the C-types if they satisfy the same quantifier-free formulas that use the predicates listed above.
In a nonrepeating edge-free word, if two valuations η_1, η_2 have the same C-types, then the tuples of selected letters

$$(a_{\eta_1(x_1)}, \ldots, a_{\eta_1(x_n)}) \qquad (a_{\eta_2(x_1)}, \ldots, a_{\eta_2(x_n)})$$

are in the same orbit under the action of G_C. Because $\mathcal{B}$ has constants C, it follows that it cannot distinguish between two valuations of the same equality-type. Therefore, $\mathcal{B}$ is equivalent to a quantifier-free formula that only uses the predicates listed above. □

B Proof of Theorem 5.5

Theorem 5.5. *Let A_{fin} be a finite set. For a set of data words over the alphabet $A_{\text{fin}} \times \mathbb{D}$, the following conditions are equivalent:*

1. *L is definable by an equivariant formula of nominal first-order logic.*
2. *L is definable by a formula of first-order logic, which uses the position order $x < y$, the equal data predicate $x \sim y$, and the label predicates $\{a(x)\}_{a \in A_{\text{fin}}}$.*

Proof. For the direction from 2 to 1 it is clear that the order, the equal data predicate and the label predicates can be expressed by basic types in nominal first-order logic. The result follows.
For the other direction consider the following lemma:

Lemma B.1. *Let $L \subseteq A^*$ be an orbit-finite and finitely supported set of words. Then L is definable by a formula of first-order logic as in Item 2.*

Proof. We only consider the case when L has actually only one orbit, for more orbits, we simply use finite disjunction. The language L is the orbit, under the action of G, of some word

$$(a_1, d_1) \cdots (a_n, d_n).$$

It is easy to see that a word

$$(b_1, e_1) \cdots (b_k, e_k)$$

belongs to L if and only if

$$n = k \quad \text{and} \quad a_1 = b_1, \ldots, a_n = b_n \quad \text{and} \quad \bigwedge_{i,j \in \{1,\ldots,n\}} (d_i = d_j) \iff (e_i = e_j)$$

The above properties are easily seen to be expressible in first-order logic. The first one can be done with the order, the second one with the label predicates and the last one with the equal data predicate. □

It follows from Lemma B.1 that any equivariant basic type of nominal first-order logic can be expressed in first-order logic. From Theorem 5.2, we know that any formula of nominal first-order logic can be assumed to be without orbit-finite disjunction. Moreover an equivariant formula of nominal first-order logic that does not use orbit-finite disjunction can only use equivariant basic types. The implication from 1 to 3 follows. □

Part II of the Appendix.

Theorem 5.2

In this part we prove our main Theorem, Theorem 5.2. We proceed as follows, in Appendix C, we introduce additional definitions and notations that are used in the proof. In Appendix D we prove the Local Functionality Theorem of Section 6. Finally, in Appendix E, we use this Theorem to prove Theorem 5.2.

C Notations and Definitions

C-stabilizer and x-orbit. Let X be an FM set and C a set of data values. The *C-stabilizer* of $x \in X$ is the group of bijections σ in G_C such that $x \cdot \sigma = x$. If x, y are elements of two FM sets, X, Y, the *(x, C)-orbit* of y is the set $\{y \cdot \sigma \mid \sigma \text{ in the } C\text{-stabilizer of } x\}$.

Least support. Let X be an FM set. An element $x \in X$ is said to have least support iff there exists a support C of x which is contained in every support of x. In that case we write $\mathrm{mem}(x)$ this support C. Moreover, a symmetry $(\mathbb{D}, G)$ admits least support iff every element of every FM set has least support. It is known that every Fraïssé symmetry admits least support. In particular this is the case for the equality symmetry (see [2]).

Finite subsets. Let X be an FM set. We write $P_{\mathrm{fin}}(X)$ the finite subsets of X. Observe that $P_{\mathrm{fin}}(X)$ is an FM set as well.

D Local Functionality Theorem

In this appendix, we prove Theorem 6.2 as well as a corollary. Note that the proof of Theorem 5.2 in Appendix E actually uses the corollary rather than Theorem 6.2 itself.

D.1 Proof of Theorem 6.2

Theorem 6.2 (Local Functionality Theorem). *Let X be an orbit finite set, and Σ a nominal relational signature, both supported by a finite set $C \subseteq \mathbb{D}$. There is some $k \in \mathbb{N}$ such that for every $x \in X$ and structure $\mathfrak{A}$, the following conditions are equivalent*

- *x is a C-supported function of $\mathfrak{A}$;*
- *x is a C-supported function of some $\mathfrak{B} \in \mathrm{substructures}_{\leq k}(\mathfrak{A})$.*

Before proving Theorem 6.2, we state the following lemma that provides alternate characterizations for the functionality notion.

Lemma D.1. *Let X, Y be FM sets, both supported by a finite set $C \subseteq \mathbb{D}$. For $x \in X$ and $y \in Y$, the following conditions are equivalent:*

1. *$y \in Y$ is a C-supported function of $x \in X$;*
2. *the C-stabilizer of x is a subgroup of the C-stabilizer of y;*

3. the (x, C)-orbit of y is $\{y\}$;.

Proof. For the implication from 1 to 2, let $f : X \dot{\to} Y$ be a C-supported partial function with $f(x) = y$. Suppose that $\pi \in G_C$ stabilizes x. Then

$$y \cdot \pi = f(x) \cdot \pi = f(x \cdot \pi) = f(x) = y,$$

and therefore π also stabilizes y. Implication 2 to 3 is by definition. For implication 3 to 1, we observe that the (x, C)-orbit of y is a C-supported function of x, and that y is a function of $\{y\}$. □

We are now ready to prove Theorem 6.2:

Proof (of Theorem 6.2). We need to exhibit a bound k depending only on X, Σ and C such that for every structure $\mathfrak{A}$ and every $x \in X$ that is a C-supported function of $\mathfrak{A}$, one can extract a substructure $\mathfrak{B}$ of size at most k such that x is still a C-supported function of $\mathfrak{B}$.
Set m as the maximal arity of predicates in Σ. Also observe that since X is orbit finite $\sup_{z \in X} |\mathrm{mem}(z)|$ is well defined. Set

$$k_1 \stackrel{\text{def}}{=} m \cdot \sup_{z \in X} |\mathrm{mem}(z)|$$

Notice that k_1 depends only on Σ and X. Now set

$$k_2 \stackrel{\text{def}}{=} m \cdot k_1!$$

Finally, set $k = k_1 + k_2$ and observe that k only depends on Σ and X.
Let $\mathfrak{A}$ be a structure and $x \in X$ such that x is a C-supported function of $\mathfrak{A}$. We extract a substructure $\mathfrak{B}$ as in the statement of the theorem. We proceed in two steps. First, we extract a substructure $\mathfrak{A}|I$ with $|I| \leq k_1$ and such that $\mathrm{mem}(x) \setminus C \subseteq \mathrm{mem}(\mathfrak{A}|I)$. Then we show that it suffices to add k_2 of positions to I in order to get the desired substructure $\mathfrak{B}$ of size k.

By definition, a permutation π stabilizes a relational structure $\mathfrak{A}$ if it stabilizes every label assigned by every interpretation to every tuple of positions. Therefore, we have,

$$\mathrm{mem}(\mathfrak{A}) = \bigcup_{\substack{I \subseteq \mathrm{dom}(\mathfrak{A}) \\ |I| \leq m}} \mathrm{mem}(\mathfrak{A}|I)$$

Since x is a C-supported function of $\mathfrak{A}$, it follows that $\mathrm{mem}(x) \setminus C$ is a subset of $\mathrm{mem}(\mathfrak{A})$. Therefore, for every data value $d \in \mathrm{mem}(x) \setminus C$ there is some subset I_d of size at most m such that d belongs to $\mathrm{mem}(\mathfrak{A}|I_d)$. Define I to be the union

of I_d ranging over $d \in \mathrm{mem}(x) \setminus C$. By definition we have $|I| \leq m \cdot \mathrm{mem}(x) \leq k_1$ and

$$\mathrm{mem}(x) \setminus C \subseteq \mathrm{mem}(\mathfrak{A}|I)$$

Set $D = \mathrm{mem}(\mathit{structa}|I)$. For a set K such that $I \subseteq K \subseteq \mathrm{dom}(\mathfrak{A})$, define

$$S_K \stackrel{\mathrm{def}}{=} \{\pi|_D : \pi \text{ is in the } C\text{-stabilizer of } \mathfrak{A}|K\}.$$

This is a subgroup of the bijections of C, because the stabilizer of $\mathfrak{A}|K$ is contained in the stabilizer of $\mathfrak{A}|I$, and the latter stabilizer preserves $D = \mathrm{mem}(\mathit{structa}|I)$ as a set. It is not difficult to see that

$$S_{\mathrm{dom}(\mathfrak{A})} = \bigcap_{\substack{K \subseteq \mathrm{dom}(\mathfrak{A}) \\ |K| \leq m}} S_{I \cup K}.$$

There are at most $|D|!$ possible values for $S_{I \cup K}$. Therefore, there is a family $\mathcal{K}$ of at most $|D|!$ sets such that

$$S_{\mathrm{dom}(\mathfrak{A})} = \bigcap_{K \in \mathcal{K}} S_{I \cup K}.$$

Define J as $\cup_{K \in \mathcal{K}} K$. By definition $|J| \leq m \cdot k_1! \leq k_2$. Finally set $\mathfrak{B} = \mathfrak{A}|(I \cup J)$. By construction $\mathfrak{B}$ is of size k.
It remains to prove that x is a C-supported function of $\mathfrak{B}$. By Lemma D.1, it is enough to prove that the C-stabilizer of $\mathfrak{B}$ is included in the C-stabilizer of x. Suppose then that $\pi \in G_C$ stabilizes $\mathfrak{B}$. It follows that π stabilizes $\mathfrak{A}|(I \cup K)$ for every $K \in \mathcal{K}$. By choice of $\mathcal{K}$, it follows that $\pi|_D$ belongs to $S_{\mathrm{dom}(\mathfrak{A})}$. By definition of $S_{\mathrm{dom}(\mathfrak{A})}$, there is some $\sigma \in G_C$ which stabilizes $\mathfrak{A}$ and such that $\sigma|_D = \pi|_D$. By Lemma D.1 and the assumption on x being a C-supported function of $\mathfrak{A}$, we know that σ also stabilizes x. Because π and σ agree on D, and D contains the least support of x, also π stabilizes x. □

D.2 A corollary of Theorem 6.2

Recall the list of issues given in Section 6:

1. Sometimes, more than one element is needed to define a data value;
2. Sometimes, a data value can only be defined in combination with some indistinguishable other data value;
3. Sometimes, both problems above hold simultaneously.

Theorem 6.2 can be seen as dealing with the first issue, since it says that if a data value requires several elements to be defined, then it actually only requires a small and bounded number of elements. To deal with the second and third issues, we use the following corollary.

Corollary D.2. *Let X be an orbit finite set and Σ a nominal relational signature both supported by a finite set $C \subseteq \mathbb{D}$. There is some $k \in \mathbb{N}$ and a finite set of C-supported functions*

$$H \subseteq \text{atoms}_k(\Sigma) \to P_{\text{fin}}(X)$$

such that for every structure $\mathfrak{A}$ and $x \in X$, the following conditions are equivalent

- *the $(\mathfrak{A}, C)$-orbit of x is finite;*
- *the $(\mathfrak{A}, C)$-orbit of x is $h(\mathfrak{B})$ for some $h \in H$ and $\mathfrak{B} \in \text{substructures}_{\leq k}(\mathfrak{A})$.*

The corollary is a consequence of the following lemma which bounds the size of finite (x, C)-orbits when $x \in X$.

Lemma D.3. *There is some $n \in \mathbb{N}$ such that for every $\mathfrak{A}$ and $x \in X$, if the $(\mathfrak{A}, C)$-orbit of x is finite, then it has at most n elements.*

Proof. Let $l = \sup_{z \in X} |\text{mem}(z)|$ and m the maximal arity of predicates in the signature Σ. Set $n = l^{lm}$ and let $x \in X$ such that the $(\mathfrak{A}, C)$-orbit of x is finite. We show that it has less than n elements.

First observe that $\text{mem}(x) \setminus C \subseteq \text{mem}(\mathfrak{A})$. Indeed assume by contradiction that $\text{mem}(x) \setminus C \not\subseteq \text{mem}(\mathfrak{A})$. Therefore there exists some data value

$$d \in \text{mem}(x) \setminus (\text{mem}(\mathfrak{A}) \cup C)$$

Since $\text{mem}(\mathfrak{A}) \cup C$ is a finite set of data values, there exists infinitely many data values $d_1, d_2, \ldots$ such that

$$d_i = \pi_i(d) \qquad \text{for some } \pi_i \in G \text{ with } \pi_i|_{(\text{mem}(\mathfrak{A}) \cup C)} = id|_{(\text{mem}(\mathfrak{A}) \cup C)}.$$

For every i, the permutation π_i stabilizes $\mathfrak{A}$ and C and is therefore in G_C. However for every i, the least support of each element $x \cdot \pi$ contains d_i, and therefore the are infinitely many elements among $x \cdot \pi_1, x \cdot \pi_2, \ldots$ This contradicts that the $(\mathfrak{A}, C)$-orbit of x is finite.

Since every tuple in $\mathfrak{A}$ that can be assigned a label found in a substructure of size at most m, every $d \in \text{mem}(x) \setminus C$ can be found in a substructure of size at most m. Therefore, there exists a substructure $\mathfrak{B}$ of size at most ml such that $\text{mem}(x) \setminus C \subseteq \text{mem}(\mathfrak{B})$. By definition, the $(\mathfrak{A}, C)$-orbit of x is included in its $(\mathfrak{B}, C)$-orbit. Therefore it remains to show that the $(\mathfrak{B}, C)$-orbit of x has at most $n = l^{lm}$ elements.

Since, $\mathrm{mem}(\mathfrak{B})$ is finite, it follows that the family

$$\{\pi|_{\mathrm{mem}(x)} : \pi \in G_c \text{ and in the stabilizer of } \mathfrak{B}\}$$

is a finite set at most $\mathrm{mem}(x)^{\mathrm{mem}(\mathfrak{B})}$ functions. Since $x \cdot \pi$ is uniquely determined by $\pi|_{\mathrm{mem}(x)}$, it follows that the $(\mathfrak{B}, C)$-orbit of x is finite and of size at most $\mathrm{mem}(x)^{\mathrm{mem}(\mathfrak{B})} \leq l^{lm}$. □

We can now prove Corollary D.2:

Proof (of Corollary D.2). Apply Lemma D.3 to X, yielding some n such that for every $x \in X$ and structure $\mathfrak{A}$, the $(\mathfrak{A}, C)$-orbit of x has size at most n elements. The family of subsets of X of size at most n is an orbit-finite set. For every $x \in X$, the $(\mathfrak{A}, C)$-orbit of x is a C-supported function of $\mathfrak{A}$. Apply Theorem 6.2 to the family of subsets of X of size at most n, yielding a number k. Define H to be the set of functions from structures of size at most k to subsets of X of size at most n.
Both the structures of size at most k and subsets of X of size at most n being orbit-finite sets there are finitely many C-supported functions from one set into another. Therefore the set H is finite. By Theorem 6.2, we know that H satisfies the equivalence in the statement of the proposition. □

E Proof of Theorem 5.2

Theorem 5.2. *Consider the equality symmetry. For every formula of nominal first-order logic there is an equivalent formula, which does not use orbit-finite disjunction.*

First observe, that an orbit finite set of formulas may only depend on finitely many different variables. Therefore, it makes sense to talk about the *arity* of an orbit finite set of formulas.
We prove Theorem 5.2 as a special case of the following more general proposition:

Proposition E.1. *Let $k \in Nat$. Let Γ be an orbit finite FM set of formulas with arity n and a function*

$$f : \mathrm{atoms}_k(\Sigma) \to P_{\mathrm{fin}}(\Gamma)$$

both with support C. Then there exists a formula $\varphi'(\bar{x}, \bar{y})$ with arity $n + k$ and support C but without orbit-finite disjunction, such that for every structure $\mathfrak{A}$ and every valuation,

$$\mathfrak{A} \models \varphi'(\bar{x}, \bar{y}) \qquad \textit{iff} \qquad \bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \varphi(\bar{x}) \tag{1}$$

holds, provided that $f(\mathfrak{A}|\bar{y})$ is a $(\mathfrak{A}, C)$-orbit.

Before proving Proposition E.1, we use it to finish the proof of Theorem 5.2. Let φ be a FM first-order logic formula (possibly with orbit-finite disjunction) and let C be its support. We construct an equivalent formula without orbit finite disjunctions. Set $k = 0$ and consider the orbit finite FM set of formulas $\Gamma = \{\varphi\}$ with support C. Finally set

$$f : \text{atoms}_0(\Sigma) \to P_{\text{fin}}(\Gamma)$$

to be the constant function with image $\{\varphi\}$ and support C. Applying Proposition E.1, we get a formula φ' without orbit-finite disjunctions such that

$$\mathfrak{A} \models \varphi'(\bar{x}) \qquad \text{iff} \qquad \mathfrak{A} \models \varphi(\bar{x})$$

and we are finished.

It remains to prove Proposition E.1. The rest of this appendix is now devoted to this proof.

Proof (of Proposition E.1). Let k, Γ, f and C be as in the statement of the Proposition E.1. We begin by stating two lemmas which we use to simplify set of formulas Γ:

Lemma E.2. *Without loss of generality, we may assume that Γ has only one C-orbit.*

Proof. By hypothesis, Γ has finitely many C-orbits. Therefore orbit finite disjunction over Γ can be replaced by a finite disjunction of orbit finite disjunctions over each C-orbit. □

Because of Lemma E.2 we assume that Γ has only one C-orbit. Therefore, the parse tree (the structure of connectives) of every formula $\varphi \in \Gamma$ is the same; the only difference is in the basic types used as predicates.

Lemma E.3. *Without loss of generality, we may assume that for any orbit-finite disjunction occuring in the parse tree of Γ, the associated set of formulas can be indexed by $\mathbb{D}$.*

Proof. It simple to see that any orbit-finite set can be indexed by $\mathbb{D}^n$ for some n. If $\Psi = \{\psi_{d_1,\ldots,d_n}\}_{(d_1,\ldots,d_n)\in\mathbb{D}^n}$ is indexed by $\mathbb{D}^n$ then the formula

$$\bigvee_{(d_1,\ldots,d_n)\in\mathbb{D}^n} \psi_{(d_1,\ldots,d_n)}$$

is equivalent to the following formula satisfying the property:

$$\bigvee_{d_1\in\mathbb{D}} \cdots \bigvee_{d_n\in\mathbb{D}} \psi_{(d_1,\ldots,d_n)}$$

□

Because the parse tree of every formula $\varphi \in \Gamma$ is the same; it makes sense to use induction on the structure of the formula $\varphi \in \Gamma$.
By DeMorgan laws, it is sufficient to consider only the cases of basic types, binary disjunction, existential quantification, negation, and orbit-finite disjunction.

Basic type. Suppose each formula $\varphi \in \Gamma$ for $\mathcal{B}$. We want a formula $\varphi'(\bar{x}, \bar{y})$ which says:

$$\bigvee_{\mathcal{B} \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \mathcal{B}(\bar{x}). \tag{2}$$

Observe that the property (2) does not depend on the whole structure $\mathfrak{A}$ but only on the substructure in the nodes $\bar{x}$ and $\bar{y}$. Moreover, the property is supported by C. Therefore, it is captured by a basic type, which has variables $\bar{x}$ and $\bar{y}$ and support C.

Binary disjunction. Suppose that each formula $\varphi \in \Gamma$ is of the form

$$\varphi = \theta_1 \vee \theta_2$$

We write Θ_1, Θ_2 the corresponding families of formulas. Both families have support C. Therefore, we can apply the induction hypothesis to both families together with the function f yielding formulas θ_1' and θ_2', and define

$$\varphi'(\bar{x}, \bar{y}) \stackrel{\text{def}}{=} \theta_1'(\bar{x}, \bar{y}) \vee \theta_2'(\bar{x}, \bar{y}).$$

It is not difficult to see that this formula satisfies (1).

Existential quantification. Suppose that each formula $\varphi \in \Gamma$ is of the form

$$\varphi(\bar{x}) = \exists z \ \theta(\bar{x}, z)$$

Let Θ be the familiy of formulas θ, it has finite support C. Therefore, we can apply the induction hypothesis to this family together with the function f, yielding a formula $\theta'(\bar{x}, z, \bar{y})$ and define

$$\varphi'(\bar{x}, \bar{y}) \stackrel{\text{def}}{=} \exists z \ \theta'(\bar{x}, z, \bar{y})$$

Again it is not difficult to see that this formula satisfies (1).

Negation. Suppose that each formula $\varphi \in \Gamma$ is of the form

$$\varphi = \neg\theta$$

Apply the induction assumption to f and the family Θ of formulas θ (with support C), and define

$$\varphi'(\bar{x}, \bar{y}) = \neg(\theta'(\bar{x}, \bar{y})).$$

Let us prove the left-to-right implication in (1). Suppose that that φ' holds in $\mathfrak{A}$ for some valuation $\bar{x}, \bar{y}$ such that $f(\mathfrak{A}|\bar{y})$ is a $(C, \mathfrak{A})$-orbit. By induction assumption, we know that

$$\neg\Big(\bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \neg\varphi(\bar{x})\Big).$$

This means that

$$\bigwedge_{\varphi \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \neg\varphi(\bar{x}). \tag{3}$$

The problem is that Proposition E.1 requires a disjunction, not a conjunction:

$$\bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \neg\varphi(\bar{x}). \tag{4}$$

However, we use the fact $f(\mathfrak{A}|\bar{y})$ is a $(C, \mathfrak{A})$-orbit together with the following lemma to prove that the conjunction (3) and the disjunction (4) are equivalent.

Lemma E.4. *Let $\Gamma' \subseteq \Gamma$ a $(C, \mathfrak{A})$-orbit of Γ then:*

$$\mathfrak{A} \models \varphi_1(\bar{x}) \qquad \textit{iff} \qquad \mathfrak{A} \models \varphi_2(\bar{x}) \qquad \textit{for every } \varphi_1, \varphi_2 \in \Gamma'$$

Proof. Because Γ' is a $(C, \mathfrak{A})$-orbit, there exists a permutation $\sigma \in G_C$ such that $\varphi_2 = \varphi_1 \cdot \sigma$ and $\mathfrak{A} \cdot \sigma = \mathfrak{A}$. The result then follows from equivariance of the satisfaction relation $\models$. □

The converse implication in (1) is proved by reversing the reasoning above.

Orbit-finite disjunction. This is the most difficult case, suppose that each formula $\varphi \in \Gamma$ is of the form

$$\varphi = \bigvee \Theta_\varphi$$

where Θ_φ is an orbit finite set of formulas. The general idea is to use functionality in order to eliminate the orbit finite disjunction. However, we have to distinguish two cases. For a formula to be a function of a structure, the support of the formula need to be included in the structure. For this reason, e define $\varphi'(\bar{x}, \bar{y})$ as the disjunction of two formulas $\alpha(\bar{x}, \bar{y})$ and $\beta(\bar{x}, \bar{y})$. The formula α searches for formulas with a support included in the support of the structure:

$$\mathfrak{A} \models \alpha(\bar{x}, \bar{y}) \qquad \text{iff} \qquad \bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \bigvee_{\substack{\psi \in \Theta_\varphi \\ \mathrm{mem}(\psi) \setminus C \subseteq \mathrm{mem}(\mathfrak{A})}} \mathfrak{A} \models \psi(\bar{x}) \tag{5}$$

provided that $f(\mathfrak{A}|\bar{y})$ is a $(C, \mathfrak{A})$-orbit.
The formula β searches for formulas with a support that is not included in the support of the structure:

$$\mathfrak{A} \models \beta(\bar{x}, \bar{y}) \qquad \text{iff} \qquad \bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \bigvee_{\substack{\psi \in \Theta_\varphi \\ \mathrm{mem}(\psi) \setminus C \not\subseteq \mathrm{mem}(\mathfrak{A})}} \mathfrak{A} \models \psi(\bar{x}) \tag{6}$$

provided that $f(\mathfrak{A}|\bar{y})$ is a $(C, \mathfrak{A})$-orbit.
By definition φ' satisfies (1). It remains to prove that such formulas α and β can be defined. We begin with β.

Construction of β. This is where we the assumption we made in Lemma E.3 is crucial. We know that all orbit-finite disjunctions used in the parse-tree of our formulas can be indexed by $\mathbb{D}$. We fix such an index and for all $d \in \mathbb{D}$ and all $\varphi \in \Gamma$ we write $\psi_{(\varphi,d)}$ the formula with index d in Θ_φ. Observe, that by hypothesis, the formula $\psi_{(\varphi,d)}$ has support $C \cup \{d\}$. This means that $\mathrm{mem}(\psi_{\varphi,d}) \setminus C \not\subseteq \mathrm{mem}(\mathfrak{A})$ is equivalent to saying that $d \notin \mathrm{mem}(\mathfrak{A}) \cup C$. Therefore, the formula (6) can be rewritten:

$$\mathfrak{A} \models \beta(\bar{x}, \bar{y}) \qquad \text{iff} \qquad \bigvee_{\varphi \in f(\mathfrak{A}|\bar{y})} \bigvee_{d \notin \mathrm{mem}(structa) \cup C} \mathfrak{A} \models \psi_{\varphi,d}(\bar{x})$$

We proceed as follows: first we swap the two orbit-finite dijunctions in (6) and apply the induction hypothesis to obtain an orbit-finite set of formulas that do not use orbit-finite disjuntion. Then we construct the desired formula β from this set.
For all $d \in \mathbb{D}$ consider the family Γ_d of formulas $\psi_{\varphi,d}$ for $\varphi \in \Gamma$. By definition each set Γ_d is supported by $D = C \cup \{d\}$, has only one D-orbit. Moreover the parse-tree of the formulas is smaller than the parse-tree of the formulas of Γ. Therefore, we can apply the induction hypothesis to these sets for the integer k and the function f, and obtain a family of formulas not using orbit-finite disjunction:

$$\{\varphi'_d(\bar{x}, \bar{y})\}_{d \in \mathbb{D}}$$

with support C and such that for all $d \in \mathbb{D}$:

$$\mathfrak{A} \models \varphi'_d(\bar{x}, \bar{y}) \qquad \text{iff} \qquad \bigvee_{i \in f(\mathfrak{A}|\bar{y})} \mathfrak{A} \models \varphi_{(i,d)}(\bar{x}) \tag{7}$$

provided that $f(\mathfrak{A}|\bar{y})$ is an $(D, \mathfrak{A})$-orbit. We now use the following lemma to construct β out of this set:

Lemma E.5. *Let $\{\varphi'_d(\bar{x})\}_{d \in \mathbb{D}}$ be a family of formulas which does not use orbit-finite disjunction and has finite support C. Then there exists a formula β supported by C and not using orbit-finite dijunction such that*

$$\mathfrak{A} \models \beta(\bar{x}) \qquad \textit{iff} \qquad \bigvee_{d \notin \mathrm{mem}(\mathfrak{A}) \cup C} \mathfrak{A} \models \varphi'_d(\bar{x})$$

Proof. The proof is by induction on the structure of the formulas, using the same arguments as above. There is one important thing to point out: when doing the step of negation, in order to use Lemma E.4 we need the set $\mathbb{D} \backslash \mathrm{mem}(\mathfrak{A}) \cup C$ to be a $(C, \mathfrak{A})$-orbit. This fact is particular to the equality symmetry, it fails for instance in the graph symmetry. This is the only place where the proof of Theorem 5.2 depends on the equality symmetry. Note that in this case the formulas in the family are assumed to be without orbit-finite disjunction. Therefore the case of orbit-finite disjunction does not occur in the induction. □

Applying Lemma E.5 to the family $\{\varphi'_d\}_{d \in \mathbb{D}}$ we get a formula $\beta(\bar{x}, \bar{y})$ such that

$$\mathfrak{A} \models \beta(\bar{x}, \bar{y}) \qquad \text{iff} \qquad \bigvee_{d \notin \mathrm{mem}(\mathfrak{A}) \cup C} \mathfrak{A} \models \varphi'_d(\bar{x}, \bar{y})$$

By combining this with property (7) above we get exactly the needed property (6).

We now move to the construction of the formula α in (5).

Construction of α. This is the case for which functionality is needed. The general idea is to merge the sets Θ_φ and apply the induction hypothesis on the new resulting set.
Let Δ be the set $\bigcup_{\varphi \in \Gamma} \Theta_\varphi$. By construction Δ remains orbit finite and supported by C. Therefore, we can apply Corollary D.2 to Δ. This yields $k' \in \mathbb{N}$ and a finite set H of functions such that for every structure $\mathfrak{A}$ and $\psi \in \Delta$ the following conditions are equivalent:

- the $(C, \mathfrak{A})$-orbit of ψ is finite;
- the $(C, \mathfrak{A})$-orbit of ψ is $h(\mathfrak{B})$ for some $h \in H$ and $\mathfrak{B} \in \mathrm{substructures}_{\leq k'}(\mathfrak{A})$.

We proceed as follows: we prove that the right side of the equality in (5) is equivalent to a set of three properties which we all show to be definable in FM first-order logic (using the induction hypothesis for one of them).
Consider the right side of the equality (5). All formulas ψ that are in the disjunction iff they verify the two following properties:

– $\psi \in \Theta_\varphi$ for $\varphi \in f(\mathfrak{A}|\bar{y})$.
– $\mathrm{mem}(\psi) \setminus C \subseteq \mathrm{mem}(\mathfrak{A})$.

By definition, this means that $(C, \mathfrak{A})$-orbit of ψ is finite. By construction with Corollary D.2, there exists some $h \in H$ and $\mathfrak{B} \in \mathrm{substructures}_{\leq k'}(\mathfrak{A})$ such that the $(C, \mathfrak{A})$-orbit of ψ is $h(\mathfrak{B})$. Therefore, a formula ψ is in the disjunction of (5) iff there exists some $h \in H$ and some $\mathfrak{B} \in \mathrm{substructures}_{\leq k'}(\mathfrak{A})$ such that:

– $h(\mathfrak{B}) \subseteq \bigcup_{\varphi \in f(\mathfrak{A}|\bar{y})} \Theta_\varphi$.
– $h(\mathfrak{B})$ is a $(C, \mathfrak{A})$-orbit of Δ.

This means that the right side of (5) is equivalent to saying that for some $h \in H$ and some $\mathfrak{B} \in \mathrm{substructures}_{\leq k'}(\mathfrak{A})$,

1. $h(\mathfrak{B})$ is a $(C, \mathfrak{A})$-orbit of Δ,
2. $h(\mathfrak{B}) \subseteq \bigcup_{\varphi \in f(\mathfrak{A}|\bar{y})} \Psi_\varphi$.
3. for some $\psi \in h(\mathfrak{B})$, we have $\mathfrak{A} \models \psi(\bar{x})$.

It remains to show that this can be expressed in nominal first-order logic. The disjunction over $h \in H$ is a finite disjunction. Every substructure $\mathfrak{B}$ is of the form $\mathfrak{B} = \mathfrak{A}|\bar{z}$, for some k'-tuple $\bar{z}$ of elements, so the disjunction over the substructure $\mathfrak{B}$ can be captured existentially quantifying over $\bar{z}$. It remains to show that the three properties 1,2,3 can be expressed by formulas with support C, with free variables $\bar{x}$, $\bar{y}$ and $\bar{z}$ that do not use orbit-finite disjunction.

Item 1. We need to construct a formula $\varphi_{1,h}(\bar{z})$ that defines the property: $h(\mathfrak{A}|\bar{z})$ is a $(C, \mathfrak{A})$-orbit.

Lemma E.6. *Let J be an orbit finite set, C a finite set of data values and let:*

$$h : \mathrm{atoms}_{k'}(\Sigma) \to P_{\mathrm{fin}}(J)$$

be an equivariant function. There exists a formula $\varphi_{1,h}$, with k' free variables and support C, such that for every structure $\mathfrak{A}$ and every valuation:

$$\mathfrak{A} \models \varphi_{1,h}(\bar{z}) \quad \textit{iff} \quad h(\mathfrak{A}|\bar{z}) \textit{ is the } (C, \mathfrak{A})\textit{-orbit of some } j \in J$$

Proof. Let m be the maximal arity of predicates in the signature. The two following properties are equivalent:

1. $h(\mathfrak{A}|\bar{z})$ is the $(C, \mathfrak{A})$-orbit of some $j \in J$.
2. For all $\mathfrak{B} \in \mathrm{substructures}_{\leq m}(\mathfrak{A})$, $h(\mathfrak{A}|\bar{z})$ is the $(C, \mathfrak{B})$-orbit of some $j \in J$.

This is because, by definition a permutation π stabilizes a relational structure $\mathfrak{A}$ if it stabilizes every label assigned by every interpretation to every tuple, i.e., it stabilizes all substructures of size m.
If $\bar{y}$ is a valuation of m elements over $\mathfrak{A}$ the property $h(\mathfrak{A}|\bar{z})$ is the $(\mathfrak{A}|\bar{y})$-orbit under G_C of some $j \in J$ depends only on the positions $\bar{y}, \bar{z}$, and has support C. Therefore, it can be captured by a basic type with variables $\bar{y}, \bar{z}$ and support C. Property 2 can then be expressed by quantifying universally over all valuations $\bar{y}$ which yields a formula for Property 1. ◻

Item 2. This is a condition that only talks about $\mathfrak{B} = \mathfrak{A}|\bar{z}$ and $\mathfrak{A}|\bar{y}$, so it can be expressed by a basic type referring to the positions $\bar{z}$ and $\bar{y}$. This yields a formula $\varphi_2(\bar{y}, \bar{z})$ with support C.

Item 3. Consider the orbit-finite set Δ with support C. We apply the induction hypothesis to this set, the integer k' and the function h. This yields a formula $\varphi_{3,h}(\bar{x}, \bar{z})$ without orbit-finite disjunctions and with support C such that for every structure $\mathfrak{A}$ and every valuation:

$$\mathfrak{A} \models \varphi_{3,h}(\bar{x}, \bar{z}) \qquad \text{iff} \qquad \bigvee_{\psi \in h(\mathfrak{A}|\bar{z})} \mathfrak{A} \models \psi(\bar{x})$$

holds, provided that $h(\mathfrak{A}|\bar{z})$ is an $(C, \mathfrak{A})$-orbit.

We can now construct φ':

$$\varphi'(\bar{x}, \bar{y}) = \bigvee_{h \in H} \exists \bar{z} \quad \varphi_{1,h}(\bar{z}) \wedge \varphi_2(\bar{y}, \bar{z}) \wedge \varphi_{3,h}(\bar{x}, \bar{z})$$

By definition φ' is with support C without orbit-finite disjunctions and verifies the conditions of Proposition E.1. □