AUTOMATA THEORY IN NOMINAL SETS

MIKOŁAJ BOJAŃCZYK, BARTEK KLIN, AND SŁAWOMIR LASOTA

University of Warsaw
e-mail address: bojan@mimuw.edu.pl

University of Warsaw
e-mail address: klin@mimuw.edu.pl

University of Warsaw
e-mail address: sl@mimuw.edu.pl

Abstract. We study languages over infinite alphabets equipped with some structure that can be tested by recognizing automata. We develop a framework for studying such alphabets and the ensuing automata theory, where the key role is played by an automorphism group of the alphabet. In the process, we generalize nominal sets due to Gabbay and Pitts.

Contents

1998 ACM Subject Classification: MANDATORY list of acm classifications.
Key words and phrases: MANDATORY list of keywords.
OPTIONAL comment concerning the title, e.g., if a variant or an extended abstract of the paper has appeared elsewehere.

LOGICAL METHODS
IN COMPUTER SCIENCE
DOI:10.2168/LMCS-???

1. Introduction

We study languages and automata over infinite alphabets. Each alphabet comes with some structure that can be accessed by recognizing devices such as automata. Examples of such structures include:

- *Equality.* There is an infinite set $\mathbb{D}$ whose elements are called *data values*. Words are elements of $\mathbb{D}^*$, or in some cases $(\Sigma \times \mathbb{D})^*$, for some finite set Σ. There is no structure on the data values except for equality. A typical language is
$$\{d_1 \cdots d_n \in \mathbb{D}^* : d_{i+1} \neq d_i \text{ for all } i \in \{1, \ldots, n-1\}\}.$$
- *Total order.* The set of data values is equipped with a total order. A typical language is
$$\{d_1 \cdots d_n \in \mathbb{D}^* : d_{i+1} > d_i \text{ for all } i \in \{1, \ldots, n-1\}\}.$$

One could also consider data values equipped with a graph structure (where, e.g., the language of finite paths can be considered), a partial order etc.

Note that the above descriptions do not determine the data values uniquely. One of the themes in this paper is the use of "universal" alphabets to obtain well-behaved notions of automata.

A device can only access data values through the given structure (e.g. the equality or order relation). For instance, in the case of data values with equality, an automaton that accepts a two-letter word de with $d \neq e$, will also necessarily accept the word de' for any $e' \neq d$.

The notion of structure on an alphabet is naturally captured by the group of its automorphisms. For example, in the case of unordered data values, the group consists of all bijections on $\mathbb{D}$. In the case of totally ordered data values, it is the group of all monotone bijections on $\mathbb{D}$.

In general, we work with a set of data values $\mathbb{D}$, together with a group G of bijections of $\mathbb{D}$, which need not be the group of all bijections of $\mathbb{D}$. Such a pair $(\mathbb{D}, G)$ is called a *data symmetry*. We then study sets X which are acted upon by the group G. A key example is the set $X = \mathbb{D}^*$, where G acts separately on each letter. As far as languages are concerned, we work with languages $L \subseteq \mathbb{D}^*$ that are closed under actions of the group G.

1.1. **Contribution.** We now outline the main contributions of this paper.

Nominal sets for arbitrary symmetries. When working with a data symmetry $(\mathbb{D}, G)$ and a set X with an action of G, we pay attention to the interplay between the canonical action of G on $\mathbb{D}$ and the action of G on X. An example of this interplay is the definition of a nominal set. A set X is called nominal wrt. the symmetry if for every $x \in X$ there exists a *finite* set of data values $C \subseteq \mathbb{D}$, called a *support of* x, such that every $\pi \in G$ satisfies

$$\forall c \in C.\ \pi(c) = c \qquad \Rightarrow \qquad x \cdot \pi = x.$$

The left side of this implication uses the canonical action of π on $\mathbb{D}$, and the right side uses an action of π on X. The intuition is that x depends only on data values from C.

An example of a nominal set is $\mathbb{D}^*$, regardless of G: a support of a word can be chosen as the set of letters that appear in the word. In the case of data values with equality, where $\mathbb{D}$ is a countably infinite set and G is the group of all bijections on $\mathbb{D}$, the theory of nominal sets was developed by Gabbay and Pitts [15]. One of the contributions of this paper is a concept of nominal sets in different symmetries.

Automata theory in arbitrary symmetries. We study the theory of automata in various symmetries. For basic definitions of automata and languages, we transfer classical definitions to the world of nominal sets. A crucial aspect here is an appropriate choice of the notion of 'finiteness'. As far as nominal sets are considered, the appropriate notion is *orbit finiteness*. Thus the abstract definitions we work with are just the classical definitions, in which the requirement of finiteness (of alphabet, state space, etc.) is relaxed to orbit finiteness.

It turns out that, in the cases of unordered and ordered data values, the abstract definitions are expressively equivalent with existing definitions of finite memory automata [13, 11] and register automata over totally ordered data [3, 12]. Some minor adjustments to finite memory automata are needed; in fact, they help to make the automaton model robust. For instance, independently of the data symmetry, our models admit minimization of deterministic automata. As one of our contributions, we provide an infinite-alphabet counterpart of the Myhill-Nerode theorem, thus concluding previous work on this theme [14, 3].

Effective representation. Our framework can be applied far beyond the theory of deterministic automata. We introduce a method of representing orbit finite nominal sets, together with relations and functions on them. We prove that an effective representation is possible in any symmetry of a certain form. As a result we obtain a toolkit which may be used to define and study nominal nondeterministic or alternating automata, context-free grammars, pushdown automata, Petri nets, Turing machines or many other natural models of computation.

1.2. **Background.** We briefly overview some related work on nominal sets in the context of automata theory.

Nominal sets. The theory of nominal sets originates from the work of Fraenkel in 1922, further developed by Mostowski in the 1930s. At that time, nominal sets were used to prove independence of the axiom of choice and other axioms. In Computer Science, they have been rediscovered by Gabbay and Pitts in [15], as an elegant formalism for modeling name binding. Since then, nominal sets have become a lively topic in semantics. They were also independently rediscovered by the concurrency community, as a basis for syntax-free models of name-passing process calculi, see [23, 21].

Automata for infinite alphabets. Languages over infinite alphabets are a lively topic in the automata community. Two principal sources of motivation are XML and verification. An XML document is often modeled as a tree with labels from the (infinite) set of all Unicode strings that can appear as attribute values. In software verification, the infinite alphabet can refer to pointers or function parameters.

Many automata models have been developed for infinite alphabets, including: finite memory automata [13], automata for ordered data values [3], two-way automata and automata with pebbles [22], alternating register automata [11], data automata [6], etc. See [25] for a survey. There is no consensus as to which one is the "real" analogue of regular languages in the case of infinite alphabets. This question is a topic of debate, see e.g. [22] or [4].

Nominal sets and HD-automata. Nominal sets, studied until now in the case of unordered data values, are a convenient tool for capturing name generation and binding. They were introduced by Gabbay and Pitts [15] as a mathematical model of name-binding and α-conversion.

A fruitful line of research starting from [23] (see also [21] for an overview) uses a category equivalent to nominal sets for defining history-dependent (HD) automata, a syntax-free model of process calculi that create and pass names, like π-calculus. These are closely related to the notions of automata studied here. In fact, our representation of nominal sets, and consequently our notions of automata, are inspired by, and generalize, similar results for Gabbay-Pitts nominal sets as developed in [16, 26]. An initial connection between HD-automata and finite memory automata was made in [10].

Data monoids. The idea to use group actions in formal language theory for infinite alphabets appeared in [5], which is the closest relation to our current work. That paper already includes: a group action of bijections of data values on languages, a central role of finite supports, Myhill-Nerode congruence in the monoid setting. However, the main focus of [5] is the development of a monoid theory, including Green's relations and an effective characterization of first-order definable word languages. The present paper has a more fundamental approach. In particular we study: the connection with the literature on nominal sets, different kinds of alphabets, algorithms and methods of representing sets.

1.3. **Structure of the paper.** The remainder of this paper is divided in two parts. The first part, comprising Sections 2–8, is about nominal sets in an arbitrary data symmetry and the basics of automata theory developed in orbit finite nominal sets, in place of finite classical sets. In the last two sections we briefly venture beyond orbit regular languages: we define context-free nominal languages and pushdown automata, prove them equivalent,

and discuss possible further work and other models of computation that can be expressed in nominal sets. The second part of the paper, spanning Sections 9 to 12, introduces finite representations for orbit finite nominal sets, with an application to deterministic automata.

One can also view this paper as an interleaving of two main threads. The first one comprises Sections 2, 4 and 9-11. In this thread, we study nominal sets for arbitrary symmetries and prove finite representation theorems for orbit finite nominal sets, without a reference to automata theory except as a source of examples. Under progressively stronger assumptions on the symmetries involved, we are able to obtain more concrete representations, culminating in the notion of a well-behaved Fraïssé symmetry in Section 11.

The second thread is the development of rudiments of automata theory in nominal sets, which is done is Sections 3, 5-8 and 12. There, we definite the notion of nondeterministic finite automaton in nominal sets, prove the Myhill-Nerode theorem for deterministic automata, relate our notion to finite memory automata of Kaminsky and Francez [13, 11], and finally apply finite representation theorem for orbit finite automata in Section 12.

This paper is an extended and revised version of [8]. We are grateful to Thomas Colcombet for suggesting that we use Fraïssé limits, and to Tomasz Wysocki for noticing Lemma 10.8(3).

Part 1. Nominal sets and automata

2. Group actions and data symmetries

Group actions. A (right) action of a group G on a set X is a function $\cdot : X \times G \to X$, written infix, subject to axioms

$$x \cdot e = x \qquad x \cdot (\pi\sigma) = (x \cdot \pi) \cdot \sigma$$

for $x \in X$ and $\pi, \sigma \in G$, where e is the neutral element of G. A set equipped with such an action is called a *G-set*.

Example 2.1. Any set X is a G-set with a trivial action defined by $x \cdot \pi = x$. The set G can be seen as a G-set either with the composition action ($\pi \cdot \sigma = \pi\sigma$) or with the conjugacy action ($\pi \cdot \sigma = \sigma^{-1}\pi\sigma$). For any G-sets X, Y, the Cartesian product $X \times Y$ and the disjoint union $X + Y$ are G-sets with actions defined point-wise and by cases, respectively.

For further examples, we introduce the following:

Definition 2.2. A *data symmetry* $(\mathbb{D}, G)$ is a set $\mathbb{D}$ of *data*, together with a subgroup $G \leq \mathrm{Sym}(\mathbb{D})$ of the symmetric group on $\mathbb{D}$, i.e., the group of all bijections of $\mathbb{D}$.

Example 2.3. We give names to a few important symmetries:

- the *classical symmetry*, where $\mathbb{D} = \emptyset$ and G is the trivial group,
- the *equality symmetry*, where $\mathbb{D}$ is a countably infinite set, say the natural numbers, and $G = \mathrm{Sym}(\mathbb{D})$ is the group of all bijections of $\mathbb{D}$,
- the *total order symmetry*, where $\mathbb{D} = \mathbb{Q}$ is the set of rational numbers, and G is the group of monotone bijections[1],

[1] In Section 11 it will become apparent why we chose the rational numbers and not some other totally ordered set.

- the *integer symmetry*, where $\mathbb{D} = \mathbb{Z}$ is the set of integers, and G is the group of translations $i \mapsto i + c$, isomorphic to the additive group of integers. We shall use this symmetry as a source of pathological counterexamples.

Example 2.4. For any data symmetry $(\mathbb{D}, G)$, a simple example of a G-set is the set $\mathbb{D}$ itself, with the action defined by $d \cdot \pi = \pi(d)$. The action of G on $\mathbb{D}$ extends pointwise to actions of G on tuples $\mathbb{D}^n$, words $\mathbb{D}^*$, infinite words $\mathbb{D}^\omega$, or sets $\mathcal{P}(\mathbb{D})$.

Other interesting G-sets include

$$\mathbb{D}^{(n)} = \{(d_1, \ldots, d_n) : d_i \neq d_j \text{ for } i \neq j\},$$
$$\tbinom{\mathbb{D}}{n} = \{C \subseteq \mathbb{D} : |C| = n\},$$

with G-actions inherited from $\mathbb{D}$. For a subset $C \subseteq \mathbb{D}$, there is a G-set

$$\mathbb{D}^C = \{\pi|C \ : \ \pi \in G\}.$$

In other words, this is the set of all injective functions from C to $\mathbb{D}$ that extend to some permutation from G. The action is by composition:

$$(\pi|C) \cdot \rho = (\pi\rho)|C.$$

For the total order symmetry, one may also consider e.g.

$$\mathbb{D}^{(<n)} = \{(d_1, \ldots, d_n) : d_i < d_{i+1} \text{ for } 1 \leq i < n\},$$

and for the integer symmetry,

$$\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$$

with action $k \cdot m = (k + m) \mod n$.

Orbits. For any x in a G-set X, the set

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\} \subseteq X$$

is called the *orbit* of x. Any G-set is partitioned into orbits in a unique way. We will mostly be interested in *orbit finite* sets, i.e., those that have a finite number of orbits. In the world of G-sets these play the role of finite sets.

Example 2.5. In the equality symmetry, elements of the powerset $\mathcal{P}(\mathbb{D})$ are in the same orbit if and only if they have the same cardinality. As a result, $\mathcal{P}(\mathbb{D})$ is not orbit finite.

In the equality symmetry, the set $\mathbb{D}^2$ has two orbits:

$$\{(d, d) : d \in \mathbb{D}\} \qquad \{(d, e) : d \neq e \in \mathbb{D}\}$$

In the total order symmetry, $\mathbb{D}^2$ has three orbits:

$$\{(d, d) : d \in \mathbb{D}\} \qquad \{(d, e) : d < e \in \mathbb{D}\} \qquad \{(d, e) : e < d \in \mathbb{D}\}$$

In the integer symmetry, $\mathbb{D}^2$ is not orbit finite. Indeed, for any $y \in \mathbb{D}$, the set

$$\{(x, x + y) : x \in \mathbb{D}\}$$

is a separate orbit.

In any symmetry, the set $\mathbb{D}^C$ has one orbit.

Equivariant relations and functions. Suppose that X is a G-set. A subset $Y \subseteq X$ is called *equivariant* if it is preserved under group actions, i.e. $Y \cdot \pi = Y$ holds for every $\pi \in G$. In other words, Y is a union of orbits in X. This definition extends to the notion of an *equivariant relation* $R \subseteq X \times Y$, by using the action of G on the Cartesian product, or to relations of greater arity, by using the point-wise action of G. In the special case when $R \subseteq X \times Y$ is a function f, this definition says that

$$f(x \cdot \pi) = f(x) \cdot \pi \qquad \text{for } x \in X,\ \pi \in G,$$

where the action on the left is taken in X and on the right in Y. The identity function on any G-set is equivariant, and the composition of two equivariant functions is again equivariant, therefore for any group G, G-sets and equivariant functions form a category, called G-**Set**.

If a singleton subset $\{x\}$ of a G-set is equivariant, we speak of an equivariant element x of the G-set. In other words, an equivariant element is one that is preserved under the action every element from G. Again in other words, an equivariant element is one that has a singleton orbit under the action of G.

Example 2.6. In the equality symmetry, the only equivariant function from $\mathbb{D}$ to $\mathbb{D}$ is the identity; there are exactly two equivariant functions from $\mathbb{D}^2$ to $\mathbb{D}$ (the projections), and exactly one from $\mathbb{D}$ to $\mathbb{D}^2$ (the diagonal function $d \mapsto (d, d)$). Also, the mapping $(d, e) \mapsto \{d, e\}$ is the only equivariant function from $\mathbb{D}^{(2)}$ to $\binom{\mathbb{D}}{2}$.

Note that there is no equivariant function from $\binom{\mathbb{D}}{2}$ to $\mathbb{D}^{(2)}$. For any $d, e \in \mathbb{D}$, assume that a function maps $\{d, e\}$ to (d, e). The uniquely induced equivariant relation

$$\{(\{d, e\} \cdot \pi, (d, e) \cdot \pi) : \pi \in G\}$$

is not a function, since the permutation $\pi = (d\ e)$ that swaps d and e leaves $\{d, e\}$ intact in $\binom{\mathbb{D}}{2}$, but changes (d, e) into (e, d) in $\mathbb{D}^{(2)}$.

Languages. The classical notion of a language directly generalizes to the world of G-sets. An *alphabet* is any orbit finite G-set A. Examples of alphabets in the symmetries mentioned so far include the set of data values $\mathbb{D}$, any finite set Σ, or a product $\Sigma \times \mathbb{D}$ where Σ is finite. When A is an alphabet, the set of strings A^* is treated as a G-set, with the point-wise action of G. A G-language is any equivariant subset $L \subseteq A^*$.

Example 2.7. In the examples below assume $A = \mathbb{D}$. In the equality symmetry, exemplary G-languages are:

$$\bigcup_{d \in \mathbb{D}} d \cdot \mathbb{D}^* \cdot d \qquad \bigcup_{d,e \in \mathbb{D}} (d\,e)^* \qquad \{d_1 \dots d_n : n \geq 0, d_i \neq d_j \text{ for } i \neq j\}$$

or palindromes over $\mathbb{D}$. In the total order symmetry, all monotonic words

$$\{d_1 \dots d_n : n \geq 0, d_1 < \dots < d_n\}$$

is a G-language.

3. G-automata

The notion of G-automaton, to be introduced now, is an obvious generalization of classical automata to G-sets. The definition is exactly like the classical one, except that

- the notion of finiteness is relaxed: *orbit finite* sets are considered instead of *finite* ones, and

- the components of the automaton, such as the initial and accepting states, or the transition relation, are required to be equivariant.

Our main observation in this section is that the Myhill-Nerode theorem may be lifted to the general setting of G-automata. This is the first step in the program that we develop later in Sections 5–7.

For the rest of this section we fix some data symmetry $(\mathbb{D}, G)$.

Definition 3.1. A *nondeterministic G-automaton* consists of

- an orbit finite G-set A, called the input alphabet,
- a G-set Q, the set of states,
- equivariant subsets $I, F \subseteq Q$ of initial and accepting states,
- an equivariant transition relation

$$\delta \subseteq Q \times A \times Q.$$

We say that the automaton is orbit finite if the set of states Q is so.

To define acceptance, we extend the single-step transition relation δ to the multi-step relation

$$\delta^* \subseteq Q \times A^* \times Q$$

in the usual way. A word $w \in A^*$ is accepted by an automaton if $(q_I, w, q_F) \in \delta^*$ for some initial state q_I and accepting state q_F. Note that δ^* is equivariant, similarly as I and F, and thus the set of words accepted by a G-automaton is a G-language.

3.1. **Deterministic G-automata.** From now on, unless stated otherwise, we only consider *deterministic G-automata*, the special case of a nondeterministic ones where the transition relation is a function

$$\delta : Q \times A \to Q,$$

and where the set of initial states is a singleton $\{q_I\}$. A deterministic G-automaton is called *reachable* if every state is equal to $\delta^*(q_I, w)$ for some $w \in A^*$.

Example 3.2. In this example assume the equality symmetry $G = \mathrm{Sym}(\mathbb{D})$. We describe a deterministic G-automaton recognizing the language

$$\{def : f \in \{d, e\}\}.$$

Its states are $\bot, \top$, as well as tuples of data values of size at most two:

$$Q = \{\top, \bot, \epsilon\} \cup \mathbb{D} \cup \mathbb{D}^2.$$

The state space Q has six orbits: three singleton orbits

$$\{\bot\},\ \{\top\},\ \{\epsilon\},$$

and three infinite orbits

$$\{d : d \in \mathbb{D}\},\ \{(d, d) : d \in D\},\ \{(d, e) : d \neq e \in \mathbb{D}\}.$$

The idea is that the automaton, when reading the first two letters of its input, simply stores them in its state. Then, after the third letter, it has state $\top$ or $\bot$ depending on

whether its input belongs to L or not. Formally, the transition function $\delta : Q \times \mathbb{D} \to Q$ is defined by cases:

$$\begin{aligned} \delta(\epsilon, d) &= d \\ \delta(d, e) &= (d, e) \\ \delta((d, e), f) &= \begin{cases} \top & \text{if } f \in \{d, e\} \\ \bot & \text{otherwise} \end{cases} \\ \delta(\top, d) &= \delta(\bot, d) = \bot \end{aligned}$$

This function is easily seen to be equivariant. The only accepting state is $\top$, and ϵ is the initial one.

Example 3.3. Consider the same group G and the same language as in the previous example. We describe a different automaton for the language. Its states are $\bot, \top$, as well as nonempty *sets* of data values of size at most two:

$$Y = \{\top, \bot, \epsilon\} \cup \mathbb{D} \cup \tbinom{\mathbb{D}}{1} \cup \tbinom{\mathbb{D}}{2}.$$

(In the above, $\binom{\mathbb{D}}{k}$ refers to subsets of $\mathbb{D}$ that have size exactly k.) One can give an equivariant transition function on these states by analogy to the above example, so that the resulting automaton recognizes the same language. The idea is that a state $d \in \mathbb{D}$ represents a word d of one letter, and a state $\{d\} \in \binom{\mathbb{D}}{1}$ represents a word dd of two letters, where the letters happen to be equal. Compared to the automaton from the previous example, the change is that instead of the orbit

$$O_1 = \{(d, e) : d \neq e \in \mathbb{D}\}$$

we have an orbit

$$O_2 = \{\{d, e\} : d \neq e \in \mathbb{D}\}.$$

In particular, both automata have six orbits of states. However, the new automaton is smaller in the following sense: there is an equivariant surjection from O_1 to O_2, but there is no equivariant function from O_2 to O_1.

Categorical perspective. Viewing an element of Q as a function from a singleton set $1 = \{\star\}$ to Q and a subset of Q as a function from Q to a two-element set 2, one can depict an automaton using a diagram:

$$\begin{array}{ccccc} & & 1 & & \\ & & \downarrow{\scriptstyle \iota} & & \\ Q \times A & \xrightarrow[\delta]{} & Q & \xrightarrow[\alpha]{} & 2. \end{array} \tag{3.1}$$

In the categorical approach to automata theory (see e.g. [2] and references therein), it is standard to define various kinds of sequential automata by instantiating this diagram in suitable categories. In this paper, we study the case of the category G-**Set**; this amounts to interpreting all objects in (3.1) as G-sets and arrows as equivariant functions. We consider the trivial G-action on the sets 1 and 2. This means that the initial state is a singleton orbit, and the set of accepting states is a union of orbits.

Just as Q and A are typically assumed to be finite sets in the classical case, we will typically require them to be orbit finite. This again follows from abstract categorical principles, as orbit finite G-sets are exactly finitely presentable objects in G-**Set**, just as finite sets are

finitely presentable in the category **Set** of sets and functions (see e.g. [1] for information on locally finitely presentable categories).

We note, however, that the Cartesian product of two orbit finite G-sets is not always orbit finite. A counterexample, in the integer symmetry, has been provided in Example 2.5. In particular, even if both A and Q are orbit finite, the domain $Q \times A$ of the transition function of a G-automaton is not always orbit finite. This inconvenience will be avoided when we restrict to Fraïssé symmetries in Section 11.

3.2. **Myhill-Nerode Theorem.** The Myhill-Nerode equivalence relation makes sense for any alphabet A, including infinite alphabets. That is, we consider two words $w, w' \in A^*$ to be equivalent with respect to a language $L \subseteq A^*$, denoted by $w \equiv_L w'$, if

$$wv \in L \Leftrightarrow w'v \in L \qquad \text{for every } v \in A^*.$$

Lemma 3.4. *If L is equivariant then $\equiv_L$ is equivariant too.*

Proof. We need to show:

$$w \equiv_L w' \quad \text{implies} \quad w \cdot \pi \equiv_L w' \cdot \pi. \tag{3.2}$$

Indeed, to prove the above observation, suppose that $w \equiv_L w'$. By unraveling the definition of $\equiv_L$, we need to show that, for all $v \in A^*$, the following equivalence holds.

$$(w \cdot \pi) \cdot v \in L \Leftrightarrow (w' \cdot \pi) \cdot v \in L$$

By acting on both sides by π^{-1}, this is equivalent to

$$((w \cdot \pi) \cdot v) \cdot \pi^{-1} \in L \cdot \pi^{-1} \Leftrightarrow ((w' \cdot \pi) \cdot v) \cdot \pi^{-1} \in L \cdot \pi^{-1}$$

By equivariance of L, this is equivalent to

$$((w \cdot \pi) \cdot v) \cdot \pi^{-1} \in L \Leftrightarrow ((w' \cdot \pi) \cdot v) \cdot \pi^{-1} \in L$$

By equivariance of concatenation in A^*, this is equivalent to

$$w \cdot (v \cdot \pi^{-1}) \in L \Leftrightarrow w' \cdot (v \cdot \pi^{-1}) \in L$$

The above is implied by $w \equiv_L w'$, which completes the proof of (3.2). □

Below we will use the property that the quotient of a G-set by an equivariant equivalence relation has a natural structure of G-set:

Lemma 3.5. *Let X be a G-set and let $R \subseteq X \times X$ be an equivalence relation that is equivariant. Then the quotient X/R is a G-set, under the action*

$$[x]_R \cdot \pi = [x \cdot \pi]_R$$

of G, and the abstraction mapping

$$x \mapsto [x]_R \ : \ X \to X/R$$

is an equivariant function.

Proof. Relying on equivariance of R, both well-definedness of the action of G, as well as equivariance of the abstraction mapping, are routinely checked. □

As usual, the equivalence $\equiv_L$ is a congruence with respect to appending new letters, i.e. if $w \equiv_L w'$ then $wa \equiv_L w'a$ holds for every letter $a \in A$. Thus one can define a transition function on equivalence classes

$$\delta_L : \quad A^*/\equiv_L \quad \times \quad A \quad \to \quad A^*/\equiv_L$$

such that:

$$\delta_L([w]_{\equiv_L}, a) = [w\,a]_{\equiv_L}. \tag{3.3}$$

If A is a G-set and L is a G-language then $\equiv_L$ is an equivariant relation on A^*. We call it the *syntactic congruence of* L.

Suppose that A is orbit finite and $L \subseteq A^*$ is a G-language. We define the *syntactic automaton* of L as follows: its states are equivalence classes of A^* under Myhill-Nerode equivalence $\equiv_L$, the transition function is δ_L, its initial state is the equivalence class of the empty word ε, and accepting states are equivalence classes of the words in L.

Lemma 3.6. *The syntactic automaton of a G-language is a reachable deterministic G-automaton.*

Proof. Note that we do not claim the syntactic automaton to be orbit finite.

By Lemma 3.4 the congruence $\equiv_L$ is equivariant, and thus Lemma 3.5 applies. Thus we can define an action of G on equivalence classes of $\equiv_L$ by

$$[w]_{\equiv_L} \cdot \pi = [w \cdot \pi]_{\equiv_L}. \tag{3.4}$$

So far, we have defined the structure of a G-set on the state space of the syntactic automaton. To complete the proof of the lemma, we need to show that the various components of the syntactic automaton are equivariant. It is easy to see that the initial state is a singleton orbit:

$$[\epsilon]_{\equiv_L} \cdot \pi \overset{(3.4)}{=} [\epsilon \cdot \pi]_{\equiv_L} = [\epsilon]_{\equiv_L}.$$

By equivariance of L, the set of final states is also equivariant:

$$[w]_{\equiv_L} \in F \Leftrightarrow w \in L \Leftrightarrow w \cdot \pi \in L \Leftrightarrow [w \cdot \pi]_{\equiv_L} \in F \Leftrightarrow [w]_{\equiv_L} \cdot \pi \in F.$$

Finally, the transition function in the syntactic automaton is equivariant:

$$\delta_L([w]_{\equiv_L}, a) \cdot \pi \overset{(3.3)}{=} [w \cdot a]_{\equiv_L} \cdot \pi \overset{(3.4)}{=} [(w \cdot \pi) \cdot (a \cdot \pi)]_{\equiv_L} \overset{(3.3)}{=} \delta_L([w]_{\equiv_L} \cdot \pi, a \cdot \pi).$$

□

For the language in Example 3.2, the syntactic automaton is the one in Example 3.3, and not the one in Example 3.2.

Homomorphisms of automata. Suppose that we have two deterministic G-automata

$$\mathcal{A} = (Q, A, q_I, F, \delta) \qquad \mathcal{A}' = (Q', A, q'_I, F', \delta')$$

over the same input alphabet A. An equivariant function

$$f : Q \to Q'$$

is called an automaton homomorphism if it maps q_I to q'_I, it maps F to F', and it commutes with the transition functions δ and δ' in the following sense:

$$f(\delta(q, a)) = \delta'(f(q), a) \qquad \text{for every } q \in Q \text{ and } a \in A.$$

It is easy to see that two automata related by a homomorphism recognize the same language. If there is a surjective homomorphism from $\mathcal{A}$ to $\mathcal{A}'$ then we call $\mathcal{A}'$ a homomorphic image of $\mathcal{A}$.

Myhill-Nerode theorem. In Theorem 3.8 below we state an abstract counterpart of the Myhill-Nerode theorem for infinite alphabets. The proof relies on Lemma 3.6 and on the following fact:

Lemma 3.7. *Let L be a G-language. The syntactic automaton of L is a homomorphic image of any reachable deterministic G-automaton that recognizes L.*

Proof. Consider a reachable deterministic G-automaton that recognizes L, over the alphabet A, with the initial state q_I and the transition function δ. We claim that the mapping

$$\delta^*(q_I, w) \longmapsto [w]_{\equiv_L}, \quad \text{for } w \in A^*,$$

is a homomorphism. It is total as the automaton is reachable, and well defined as $\delta^*(q_I, w) = \delta^*(q_I, v)$ implies $w \equiv_L v$. The mapping is easily shown equivariant using Lemmas 3.4 and 3.5. It commutes with the transition functions by the very definition of the syntactic automaton. The initial state q_I is mapped to the initial one $[\varepsilon]_{\equiv_L}$. Finally, the accepting states are mapped to accepting states, as $\delta^*(q_I, w)$ or $[w]_{\equiv_L}$ is accepting exactly when $w \in L$. □

Theorem 3.8 (Myhill-Nerode theorem for G-sets)**.** *Let A be an orbit finite G-set, and let $L \subseteq A^*$ be a G-language. The following conditions are equivalent:*

1. *the set of equivalence classes of Myhill-Nerode equivalence $\equiv_L$ is orbit finite;*
2. *L is recognized by a deterministic orbit finite G-automaton.*

Proof. The implication (1) $\implies$ (2) follows by Lemma 3.6. For the opposite implication, we observe that if L is recognized by a deterministic G-automaton $\mathcal{A}$ then without loss of generality one may assume that $\mathcal{A}$ is reachable, and then use Lemma 3.7. □

3.3. **Bisimulation.** We now return to the study of nondeterministic G-automata. The property that a deterministic G-automaton may be quotiented by the Myhill-Nerode equivalence, as stated in Lemma 3.7, is a special case of the quotient of nondeterministic G-automata with respect to bisimulation equivalence.

Consider a nondeterministic G-automaton $\mathcal{A}$ over an alphabet A, with transition relation δ and state space Q. We do not assume Q to be orbit finite. Except for the treatment of accepting states, the following definition is entirely standard [20]:

Definition 3.9. A binary relation $R \subseteq Q \times Q$ is a *bisimulation* if for every $(q, p) \in R$ and $a \in A$, the following conditions hold:

- whenever $(q, a, q') \in \delta$ there is some p' with $(p, a, p') \in \delta$ and $(q', p') \in R$,
- whenever $(p, a, p') \in \delta$ there is some q' with $(q, a, q') \in \delta$ and $(q', p') \in R$,
- q is accepting if and only if p is accepting.

The union of all bisimulation relations, being itself a bisimulation, we call *bisimulation equivalence*. It will be denoted by $\sim$.

Below we show that the bisimulation quotient is a legal operation in nominal sets. First, observe that even if a bisimulation is not required to be equivariant, the greatest bisimulation is always so:

Fact 3.10. The bisimulation equivalence is equivariant.

Proof. It is sufficient to show that if R is a bisimulation and $\pi \in G$ then

$$R \cdot \pi = \{(q \cdot \pi, p \cdot \pi) : (q, p) \in R\}$$

is a bisimulation too. We will only show the first condition of Definition 3.9, the others are shown similarly. Assume $(q \cdot \pi, a, q') \in \delta$. By equivariance of δ we have $(q, a \cdot \pi^{-1}, q' \cdot \pi^{-1}) \in \delta$. Now as R is a bisimulation we get some p' with $(p, a \cdot \pi^{-1}, p') \in \delta$ and $(q' \cdot \pi^{-1}, p') \in R$. After translating back via π we obtain $(p \cdot \pi, a, p' \cdot \pi) \in \delta$ with $(q', p' \cdot \pi) \in R$, as required. □

The bisimulation quotient of a G-automaton $\mathcal{A}$ is defined in a standard way. The states are equivalence classes $Q/\!\sim$ of bisimulation equivalence. The transition relation contains a triple

$$([q]_\sim, a, [p]_\sim) \tag{3.5}$$

if there is some $p' \sim p$ with $(q, a, p') \in \delta$. The state $[q]_\sim$ is accepting if q is so.

Fact 3.11. The bisimulation quotient of a G-automaton is a G-automaton.

Proof. The action of G on the set of equivalence classes of $\sim$, given by

$$[x]_\equiv \cdot \pi = [x \cdot \pi]_\equiv,$$

is well-defined by equivariance of $\sim$ (see Lemma 3.5). It only remains to show that the quotient transition relation (3.5) is equivariant. Assume $([q]_\sim, a, [p]_\sim)$ is in the quotient transition relation, i.e.,

$$(q, a, p') \in \delta, \quad \text{for some } p' \sim p.$$

Choose any $\pi \in G$. By equivariance of δ we obtain:

$$(q \cdot \pi, a \cdot \pi, p' \cdot \pi) \in \delta.$$

By equivariance of $\sim$ we have $p' \cdot \pi \sim p \cdot \pi$, thus $([q \cdot \pi]_\sim, a \cdot \pi, [p \cdot \pi]_\sim)$ is in the quotient transition relation. This means that $([q]_\sim \cdot \pi, a \cdot \pi, [p]_\sim \cdot \pi)$ is in the quotient transition relation, as required for equivariance thereof. □

Finally, the mapping $q \mapsto [q]_\sim$ is simultaneously an equivariant function and a bisimulation over the disjoint union of $\mathcal{A}$ and its bisimulation quotient.

4. Nominal G-sets

The notion of G-automaton presented in Section 3 is quite abstract. When working with a model of computation, one expects it to have some kind of concrete presentation, e.g., in terms of control states and memory. Such a presentation makes it easier to understand what the automaton does, and is necessary to design algorithms that work with automata, e.g., minimization algorithms. Although we have defined some particular automata by finite means (e.g. Example 3.2), it is not clear how an arbitrary automaton can be presented.

One of the goals of this paper is to give a concrete presentation for orbit finite G-sets, equivariant functions and algebraic structures such as automata. This, however, cannot be done in full generality even for the equality symmetry (see Example 2.3), for rather fundamental reasons:

Fact 4.1. For a countably infinite $\mathbb{D}$ and $G = \mathrm{Sym}(\mathbb{D})$, there are uncountably many non-isomorphic single-orbit G-sets.

Proof. This proof is best deferred until Proposition 9.7, after some basic representation machinery is introduced. □

Another problem with G-sets is that Cartesian product on them does not preserve orbit finiteness in general:

Example 4.2. Consider $G = \mathrm{Sym}(\mathbb{D})$ for a countably infinite $\mathbb{D}$, and let $X \subseteq \mathcal{P}(\mathbb{D})$ be the set of all those subsets of $\mathbb{D}$ that are neither finite nor cofinite. It is easy to see that X is a single-orbit G-set. However, X^2 has infinitely many orbits. Indeed, for any $n \in \mathbb{N}$ one can choose $(C_n, D_n) \in X^2$ such that $|C_n \cap D_n| = n$, and pairs (C_n, D_n) and (C_m, D_m) are in different orbits of X^2 if $n \neq m$.

Due to these difficulties, since the equality symmetry $G = \mathrm{Sym}(\mathbb{D})$ is one of the most important cases we want to consider, we need to restrict attention to some class of well-structured G-sets. To this end, we introduce the notion of a G-nominal set. Observe that so far, we have only used the group G, and we have ignored the fact that G is a group acting on some data values $\mathbb{D}$. The definition of a G-nominal sets is where the data values start to play a role.

From now on, we focus on G-sets for groups arising from data symmetries. Consider a data symmetry $(\mathbb{D}, G)$ (cf. Definition 2.2).

Definition 4.3. A set $C \subseteq \mathbb{D}$ *supports* an element $x \in X$ if $x \cdot \pi = x$ for all $\pi \in G$ that act as identity on C. A G-set is *nominal* in the symmetry $(\mathbb{D}, G)$ if its every element has a finite support.

Note that the definition of support mentions two group actions of G: an action on X, and the canonical one on $\mathbb{D}$. By abuse of notation, we usually leave the set of data values $\mathbb{D}$ implicit, and simply talk about nominal G-sets.

Nominal G-sets and equivariant functions between them form a category G-**Nom**.

Example 4.4. For any data symmetry, $\mathbb{D}$ is a nominal G-set, since every element $d \in \mathbb{D}$ is supported by $\{d\} \subseteq \mathbb{D}$. Similarly $\{d_1, \ldots, d_k\}$ supports $(d_1, \ldots, d_k) \in \mathbb{D}^k$, hence $\mathbb{D}^k$ is also a nominal G-set. The same works for $\mathbb{D}^*$, but not for $\mathbb{D}^\omega$ or $\mathcal{P}(\mathbb{D})$ if $\mathbb{D}$ is infinite.

If X, Y are nominal G-sets then so are the Cartesian product $X \times Y$ and the disjoint union $X + Y$. Indeed, if C supports $x \in X$ and D supports $y \in Y$ then $C \cup D$ supports $(x, y) \in X \times Y$, and also C supports $x \in X + Y$ and D supports $y \in X + Y$. A set X equipped with the trivial G-action is always nominal, with every element supported by the empty set.

Example 4.5. For the equality symmetry (see Examples 2.3), nominal G-sets are exactly nominal sets introduced by Gabbay and Pitts [15]. Assuming $\mathbb{D} = \mathbb{N}$, the sets $\{0, 1, 2, 3\}$ and its complement $\mathbb{N} \setminus \{0, 1, 2, 3\}$, considered as elements of $\mathcal{P}(\mathbb{D})$, are both supported by $\{0, 1, 2, 3\}$. In the equality symmetry, an element of $\mathcal{P}(\mathbb{D})$ has finite support if and only if it is finite or cofinite. In particular, there are countably many finitely supported elements in $\mathcal{P}(\mathbb{D})$.

Example 4.6. Consider the total order symmetry, where $\mathbb{D} = \mathbb{Q}$, and the element $x \in \mathcal{P}(\mathbb{Q})$ that is the union of two intervals $[0; 1] \cup [2; 3)$. It is easy to see that this element is supported by the set $\{0, 1, 2, 3\}$. More generally, an element of $\mathcal{P}(\mathbb{Q})$ has a finite support if and only if it is a finite Boolean combination of intervals.

Example 4.7. Consider the integer symmetry. If a translation $i \mapsto i + j$ preserves any single integer, then it is necessarily the identity. Therefore, any element of any set with an action of integers is supported by $\{5\}$ or $\{8\}$, etc. In the integer symmetry, all G-sets are nominal.

Suppose that we change a symmetry $(\mathbb{D}, G)$ by keeping the set of data values $\mathbb{D}$, but considering a subgroup $H \leq G$. What happens to the nominal sets? If X is a G-set (and therefore also a H-set), then every G-support of $x \in X$ is also an H-support of x, therefore every nominal G-set is a nominal H-set. On the other hand, under the smaller group H, more sets might become nominal (see Examples 4.5 and 4.6).

A basic property of equivariant functions is that they preserve supports:

Lemma 4.8. *For any equivariant $f : X \to Y$, $x \in X$ and $C \subseteq \mathbb{D}$, if C supports x then C supports $f(x)$.*

Proof. For any $\pi \in G$, if $x \cdot \pi = x$ then $f(x) \cdot \pi = f(x \cdot \pi) = f(x)$. □

Similarly, action of the group preserves supports in the following sense:

Lemma 4.9. *If C supports x then πC supports $x \cdot \pi$, for any $\pi \in G$.*

Proof. Assume an arbitrary $\rho \in G$ to be identity on πC. Then $\pi\, \rho\, \pi^{-1}$ is identity on C, and thus preserves x,

$$x \cdot (\pi\, \rho\, \pi^{-1}) = x,$$

from which we obtain:

$$(x \cdot \pi) \cdot \rho = x \cdot \pi$$

as required. □

The problem signified by Fact 4.1 disappears for nominal G-sets:

Fact 4.10. For the equality symmetry $(\mathbb{D}, G)$, there are only countably many non-isomorphic single-orbit nominal G-sets.

Proof. This will follow from the more general Corollary 10.18. □

However, other problems persist and we shall not be able to distill a satisfactory representation of nominal G-sets and automata for arbitrary data symmetries. As a pathological example, consider the integer symmetry (see Example 2.3).

Integer pathologies. As far as single-orbit nominal sets are concerned, the integer symmetry has a promisingly simple structure. As we mentioned in Example 4.7, all G-sets are nominal in this case. One example of a single-orbit G-set is $\mathbb{Z}$. Another example is the finite cyclic group $\mathbb{Z}_n$, for any nonzero $n \in \mathbb{N}$. It turns out that these are all the single-orbit sets:

Fact 4.11. Every single-orbit nominal set in the integer symmetry is isomorphic to $\mathbb{Z}$ or to $\mathbb{Z}_n$ for some $n \in \mathbb{N}$.

Proof. This will easily follow from Proposition 9.2. □

Equivariant functions between single-orbit sets are also simple. If the domain in $\mathbb{Z}$, these are all translations, possibly modulo n if the co-domain is $\mathbb{Z}_n$. If the domain is $\mathbb{Z}_n$, the co-domain must be necessarily $\mathbb{Z}_m$ for m a divisor of n.

The problems with the integer symmetry appear as soon as Cartesian products of nominal sets are considered. This has bad consequences for automata. Suppose that we are interested in automata where the set of states is $\mathbb{Z}$ and the input alphabet is also $\mathbb{Z}$. Both sets are single-orbit and nominal, so these are among the simplest automata in the integer symmetry. The transition function is any equivariant function

$$\delta : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}.$$

What functions δ can we expect? Suppose that δ is defined for arguments of the form $(0, i)$. Then, by equivariance, this definition extends uniquely to all arguments:

$$\delta(i, j) = \delta((0, j - i) \cdot i) = \delta(0, j - i) + i.$$

However, there is no restriction on the value of $\delta(0, i)$, call it $g(i)$. It is not difficult to show that for any function $g : \mathbb{Z} \to \mathbb{Z}$, the function δ_g defined by

$$\delta_g(i, j) = g(j - i) + i$$

is equivariant. In particular, there are uncountably many equivariant functions $\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$.

Wishing to disregard the integer symmetry and other pathological cases, we shall require some desirable properties of data symmetries such as the existence of least supports.

Definition 4.12. A symmetry $(\mathbb{D}, G)$ *admits least supports* if each element of every nominal G-set has the least finite support, or, equivalently, if finite supports of each element are closed under intersection.

We shall study the existence of least supports in some detail in Section 10. For now, we simply state some examples:

Example 4.13. The equality symmetry and the total order symmetry both admit least supports. This will be proved as Corollaries 10.4 and 10.5; for the equality symmetry, it was proved already in [15].

The integer symmetry does not admit least supports, as is evident from Example 4.7.

Later, in Section 10, we shall restrict attention to those data symmetries that admit least supports and enjoy some other desirable properties, to achieve a finitary representation of nominal sets. For instance, Fact 4.10 holds for any data symmetry $(\mathbb{D}, G)$ that admits least supports, assumed that $\mathbb{D}$ itself is a countable set, as we show in Corollary 10.18. For now, however, we shall continue the study of nominal automata in an abstract setting, in the following Sections 5–8.

We conclude this section with a simple fact that gives a feeling of a kind of finitary representation we mean above. Recall from Example 2.4 that $\mathbb{D}^C$ is the set of all injective functions $C \to \mathbb{D}$ that extend to a permutation from G.

Lemma 4.14. *Every single orbit nominal set X is an image, under an equivariant function, of $\mathbb{D}^C$, for some $C \subseteq_{fin} \mathbb{D}$.*

Proof. Choose any $x \in X$ and any finite support C of x. Because this is a support it follows that

$$\pi|_C = \sigma|_C \quad \Rightarrow \quad x \cdot \pi = x \cdot \sigma \qquad \text{for every } \pi, \sigma \in G.$$

Therefore, the relation defined by

$$f = \{(\pi|_C, x \cdot \pi) : \pi \in G\}$$

is actually an equivariant function from $\mathbb{D}^C$ to X. The function is clearly surjective, because every element of X is of the form $x \cdot \pi$ for some $\pi \in G$. □

In the integer symmetry every injective function $C \to \mathbb{D}$ extends to a permutation, thus we obtain:

Corollary 4.15. *In the equality symmetry, every single orbit nominal set is an image, under an equivariant function, of $\mathbb{D}^{(n)}$, for some $n \in \mathbb{N}$.*

5. Nominal G-automata

This section is a continuation of our theory of G-automata initiated in Section 3, but now we restrict attention to nominal G-sets. We call a G-automaton *nominal* if both the alphabet A and the state space Q are nominal G-sets.

Note that if A is nominal then A^* is so as well, as every finite word is supported by the union of supports of individual letters. Thus every G-language over a nominal alphabet is automatically nominal.

The restriction to nominal sets will have little or no impact on the expressive power of automata. In particular, in any data symmetry $(\mathbb{D}, G)$:

Proposition 5.1. *In any reachable deterministic G-automaton over a nominal alphabet A, the G-set of states is nominal.*

Proof. Reachability of an automaton (see Definition 3.1 and the following paragraphs) means that the function $w \mapsto \delta^*(q_I, w)$ is an equivariant function from the nominal set A^* onto the state space of the automaton. By Lemma 4.8, the image of a nominal set under an equivariant function is a nominal set. □

As before, for the rest of this section we fix some infinite set $\mathbb{D}$ and a group $G \leq \mathrm{Sym}(\mathbb{D})$. The deterministic orbit finite nominal G-automata we call shortly G-DFA; similarly, the nondeterministic orbit finite nominal G-automata we call G-NFA.

5.1. Myhill-Nerode theorem revisited.

Assume the alphabet A is an orbit finite nominal G-set. By Proposition 5.1 every reachable deterministic automaton over A is nominal. As a conclusion, the syntactic automaton is always nominal. Thus in condition (2) in Theorem 3.8 one may equivalently require that the automaton is nominal:

Theorem 5.2 (Myhill-Nerode theorem for nominal G-sets)**.** *Let A be an orbit finite nominal G-set, and let $L \subseteq A^*$ be a G-language. The following conditions are equivalent:*

1. *the set of equivalence classes of Myhill-Nerode equivalence $\equiv_L$ is orbit finite;*
2. *L is recognized by a G-DFA.*

5.2. **Nondeterministic G-automata.** In the sequel we investigate some basic properties of classical NFA, and verify which of them still hold for G-NFA.

Determinization fails. In the world of nominal sets, one cannot in general determinize finite automata. One reason is that complementation fails for nondeterministic finite automata. Perhaps a more suggestive explanation is that the powerset of an orbit finite set can have infinitely many orbits, as illustrated in Example 2.5 in the case of the equality symmetry. This means that applying the subset construction to a nominal nondeterministic finite automaton yields a nominal deterministic automaton, but not necessarily one with an orbit finite state space.

Elimination of ε-transitions. Consider nominal G-automata as in Definition 3.1, but which also have additionally ϵ-transitions, described by an equivariant relation

$$\delta_\epsilon \subseteq Q \times Q.$$

Lemma 5.3. *The expressive power of G-NFA is not changed if ε-transitions are allowed.*

Proof. The standard proof works. After eliminating ε-transitions, we should have transitions of the form $(p, a, q) \in Q \times A \times Q$ such that

$$(p_1, p_2), \ldots, (p_{n-1}, p_n) \in \delta_\varepsilon \qquad (p_n, a, q_1) \in \delta \qquad (q_1, q_2), \ldots, (q_{m-1}, q_m) \in \delta_\varepsilon$$

holds for some

$$p_1, \ldots, p_n, q_1, \ldots, q_m \in Q \qquad p_1 = p \qquad q_m = q.$$

It is not difficult to see that the new set of transitions is equivariant. □

Union and intersection. It is easy to see that languages recognized by G-NFA are closed under union (because orbit finite sets are closed under disjoint union) and concatenation (disjoint union again, and using Lemma 5.3). They also contain all orbit finite subsets of A^*. This raises the question of regular expressions and a Kleene Theorem, but we do not discuss these issues in this paper.

Closure under intersection is a bit more subtle, as it does not hold in an arbitrary symmetry. The essential reason is that orbit finite nominal sets are not stable under Cartesian product, as shown in Example 2.5 in the case of the integer symmetry. However, if one restricts to well-behaved symmetries only, as we do in Sections 10-12, the closure under products is recovered, and, as a consequence, the closure of G-NFA under intersection is recovered as well.

Complementation. For closure under complementation, the situation is much worse, as the closure fails essentially in every symmetry. The proof below works for the equality symmetry, but with minor changes it can be adapted to other symmetries.

Lemma 5.4. *In the equality symmetry, languages recognized by G-NFA are not closed under complementation.*

Proof. Anticipating Section 6, we follow that same lines as the proof that finite memory automata of Francez and Kaminsky are not closed under complementation. Consider the words over $\mathbb{D}$ which contain some data value twice:

$$L = \bigcup_{d \in \mathbb{D}} \mathbb{D}^* \cdot d \cdot \mathbb{D}^* \cdot d \cdot \mathbb{D}^*.$$

The complement of this language is the set of words where all letters are distinct. Suppose that the complement of L is recognized by a G-NFA $\mathcal{A}$, with states Q and transitions δ. For each $q \in Q$, let $C_q \subseteq \mathbb{D}$ be some chosen finite support of q. By Lemma 4.9, the sets C_q may be chosen so that the size of C_q depends only on the orbit of q, and therefore

$$\max_{q \in Q} |C_q|$$

is a finite number, since there are finitely many orbits in Q. Choose $n \in \mathbb{N}$ to be bigger than this finite number. Consider a word

$$d_1 \cdots d_{2n} \notin L.$$

This word should be accepted by $\mathcal{A}$, so there should be an accepting run

$$q_0, \ldots, q_{2n} \qquad \text{such that } (q_{i-1}, d_i, q_i) \in \delta \text{ for all } i \in \{1, \ldots, 2n\}.$$

Because the least support C_{q_n} of q_n has fewer than n data values it follows that

$$d_i, d_j \notin C_{q_n} \qquad \text{for some } i \in \{1, \ldots, n\} \text{ and some } j \in \{n+1, \ldots, 2n\}.$$

Let π be the transposition which swaps d_i and d_j. By equivariance of the transition relation, we see that the sequence

$$q_0 \cdot \pi, \ldots, q_n \cdot \pi$$

is a run over the prefix

$$(d_1 \cdots d_n) \cdot \pi.$$

Because π does not move the support of q_n, it follows that $q_n \cdot \pi = q_n$. Therefore, the sequence

$$q_0 \cdot \pi, \ldots, q_n \cdot \pi, q_{n+1}, \ldots, q_{2n}$$

is an accepting run over the word

$$((d_1 \cdots d_n) \cdot \pi) \cdot d_{n+1} \cdots d_{2n}.$$

However, the above word contains the data value d_j twice, so it should be rejected by $\mathcal{A}$. □

6. Relationship with finite memory automata

In this section, we take a detour from the discussion of automata theory in general symmetries, and we discuss the special case of the equality symmetry $(\mathbb{D}, G)$. In this case, for alphabets of a special form, the abstract model of nominal finite automata coincides with an existing automaton model, namely the finite memory automata of Francez and Kaminski. A connection between finite memory automata and nominal sets was first made in [10], in the related framework of named sets and history-dependent automata. However, no comparison of the expressive power of automata was proved there.

6.1. **Finite memory automata.** We begin by defining *finite memory automata* [13], known also under the name *register automata* [11].

Partial data tuples. Consider a finite set N of names. A partial data tuple over N is a partial function from N to $\mathbb{D}$. We write $(\mathbb{D} \cup \bot)^N$ for the set of partial data tuples. An equality constraint over N is an element

$$(r, \tau, r') \in N \times \{=, \neq\} \times N$$

We say a partial tuple t satisfies the constraint if $t(r)$ is defined, and $t(r')$ is defined, and their data values are related by τ. For instance, the completely undefined tuple is the unique partial tuple that satisfies no constraints.

Lemma 6.1. *Every equivariant subset of* $(\mathbb{D} \cup \bot)^N$ *is equivalent to a boolean combination of equality constraints.*

Proof. Fix an arbitrary orbit of $(\mathbb{D} \cup \bot)^N$ and an arbitrary element x of the orbit. Consider the set of equality constraints satisfied by x. A crucial but easy observation is that precisely the same constraints are satisfied by all other elements of the orbit. On the other side, any two tuples that satisfy the same equality constraints are related by some permutation π. Thus the orbit is equivalent to a conjunction of equality constraints. □

There are only finitely many equality constraints, as long as N is finite, thus by the above lemma $(D \cup \bot)^N$ is an orbit finite nominal set.

Definition 6.2. A *nondeterministic finite memory automaton* consists of

- a finite set A_{fin} of input labels;
- a finite set C of control states;
- a finite set N of register names;
- sets of initial $I \subseteq C$ and final $F \subseteq C$ control states;
- a transition relation, which is a subset of

$$\delta \subseteq C \times A_{\text{fin}} \times \mathit{bool}(\Phi) \times C$$

where Φ is the set of equality constraints over the following set of names:

$$N' = \quad \{\text{before}\} \times N \quad \cup \quad \{\text{input}\} \quad \cup \quad \{\text{after}\} \times N$$

and $\mathit{bool}(\Phi)$ stands for the boolean combinations of constraints from Φ.

Such an automaton $\mathcal{A}$ is used to accept or reject words over the alphabet $A_{\text{fin}} \times \mathbb{D}$, and works as follows. After reading a prefix of the input word, the configuration of the automaton consists of a control state from C together with a partial valuation from registers to data values. In other words, a configuration is an element of the set

$$Q_{\mathcal{A}} = C \times (\mathbb{D} \cup \bot)^N,$$

Suppose that the automaton is in a configuration

$$(c, d_1, \ldots, d_n) \in Q_{\mathcal{A}}$$

and that it reads an input letter $(a, d) \in A$. The automaton can nondeterministically choose any new configuration

$$(c', d'_1, \ldots, d'_n) \in Q_{\mathcal{A}}$$

provided that there is a transition

$$(c, a, \phi, c') \in \delta$$

such that the partial tuple

$$(d_1, \ldots, d_n, d, d'_1, \ldots, d'_n),$$

interpreted as a partial tuple over N', satisfies the boolean combination of equality constraints given by ϕ.

Lemma 6.3. *Consider an alphabet $A = A_{\text{fin}} \times \mathbb{D}$, where A_{fin} is a finite set. Then the following conditions are equivalent for every language $L \subseteq A^*$:*

1. *L is recognized by a finite memory automaton.*
2. *L is recognized by a G-NFA, where*
   - *The state space is $C \times (\mathbb{D} \cup \bot)^n$ for some finite set C and $n \in \mathbb{N}$.*
   - *There is only one initial state.*

Proof. The implication from 1 to 2 follows immediately from the definition. Note that $\bot$ is a singleton orbit in $\mathbb{D} \cup \bot$. For the converse implication, we use Lemma 6.1. The assumption on one initial state guarantees that the initial state is $(c, \bot^n)$ for some $c \in C$. □

6.2. **Equivalence for nondeterministic automata.** In this section, we prove a stronger version of Lemma 6.3, namely:

Theorem 6.4. *Consider an alphabet $A = A_{\text{fin}} \times \mathbb{D}$, where A_{fin} is a finite set. Then the following conditions are equivalent for every language $L \subseteq A^*$:*

1. *L is recognized by a finite memory automaton.*
2. *L is recognized by a G-NFA.*

The implication from (1) to (2) has already been shown in Lemma 6.3. The rest of Section 6.2 is devoted to the implication from (2) to (1).

Corollary 6.5. *Every orbit finite nominal set is an image, under a partial equivariant function f, of a set of the form*

$$I \times (\mathbb{D} \cup \bot)^n \qquad \text{for some finite set } I \text{ and } n \in \mathbb{N}.$$

Proof. Suppose that X is a nominal set with k orbits. Recall from Example 2.4 that $\mathbb{D}^{(n)}$ is the set of non-repeating n-tuples of data values. By Corollary 4.15, X is an image of

$$\coprod_{i \in \{1 \ldots k\}} \mathbb{D}^{(n_i)}. \tag{6.1}$$

Let n be the maximal number among $\{n_i\}_{i \in \{1 \ldots k\}}$. It is not difficult to see that $\mathbb{D}^{(n_i)}$ is isomorphic to an orbit of $(\mathbb{D} \cup \bot)^n$. It follows that the disjoint union from (6.1) is isomorphic to an equivariant subset of $\{1 \ldots k\} \times (\mathbb{D} \cup \bot)^n$. □

We are now ready to prove Theorem 6.4. Consider a G-NFA $\mathcal{A} = (Q, A, I, F, \delta)$ with $A = A_{\text{fin}} \times \delta$ for some finite set A_{fin}. We assume that there is only one initial state, call it q_I. Otherwise, we add a new initial state, call it q_I, with a trivial action

$$q_I \cdot \pi = q_I,$$

and extend the set of transitions by the equivariant set of triples of the form

$$\{(q_I, a, q) : (p, a, q) \in \delta \text{ for some } p \in I\}.$$

Basing on Lemma 6.3, we only need to show that there is an equivalent G-NFA with a single initial state, whose state space is $C \times (\mathbb{D} \cup \bot)^n$ for some finite set C and $n \in \mathbb{N}$. Apply Corollary 6.5 to Q, yielding a partial surjective equivariant function

$$f : \quad C \times (\mathbb{D} \cup \bot)^n \quad \to \quad Q$$

for some finite set C and $n \in \mathbb{N}$. Because there is just one initial state, we may assume that

$$q_I = f(c_I, \bot^n)$$

for some $c_I \in C$. Define a G-NFA, call it $f^{-1}(\mathcal{A})$, with states $C \times (\mathbb{D} \cup \bot)^n$, initial state $(c_I, \bot^n)$, final states $f^{-1}(F)$ and transitions $f^{-1}(\delta)$. It is easy to see that the automata $\mathcal{A}$ and $f^{-1}(\mathcal{A})$ recognize the same language. This completes the proof of Theorem 6.4.

Local symmetry. Although finite memory automata and G-NFA have the same expressive power, the latter model is arguably richer and has more structure. Indeed, in contrast to Lemma 3.6, syntactic automata of G-languages are not necessarily finite memory automata. An example is the automaton from Example 3.3, which does not arise from any finite memory automaton. This is because G-NFA allow for a *local symmetry*[2], as illustrated in Example 3.3 where a G-NFA stores an unordered pair of data values instead of an ordered one; on the other hand finite memory automata do not allow any notion of local symmetry, or permutation, of registers. As a result, the Myhill-Nerode theorem fails, and finite memory automata do not minimize: the syntactic automaton is always a homomorphic image of a finite memory automaton, but it may not be isomorphic to one.

The importance of local symmetries for automata minimization was first noticed in the context of history-dependent automata, in [23].

6.3. **Equivalence for deterministic automata.** Recall that the set of configurations of a finite memory automaton $\mathcal{A}$ is $Q_{\mathcal{A}} = C \times (\mathbb{D} \cup \bot)^N$. The semantics of a nondeterministic finite memory automaton is given by a transition relation between configurations, being an equivariant subset of $Q_{\mathcal{A}} \times (A_{\text{fin}} \times \mathbb{D}) \times Q_{\mathcal{A}}$. A finite memory automaton is called deterministic if this relation is actually a function $Q_{\mathcal{A}} \times (A_{\text{fin}} \times \mathbb{D}) \to Q_{\mathcal{A}}$.

In this section, we prove a deterministic variant of Theorem 6.4:

Theorem 6.6. *Consider an alphabet $A = A_{\text{fin}} \times \mathbb{D}$, where A_{fin} is a finite set. Then the following conditions are equivalent for every language $L \subseteq A^*$:*

1. *L is recognized by a deterministic finite memory automaton.*
2. *L is recognized by a G-DFA.*

[2]The notion of local symmetry is introduced in its full generality in Section 11.

We do the same proof as for the nondeterministic automata. The only problem is that $f^{-1}(\delta)$ might not in general be deterministic. To solve this problem, we need two additional results.

Recall from Example 4.13 that the equality symmetry admits least supports. Our first observation is that in Corollary 6.5 there exists a function f that is not just equivariant, but also preserves least supports. (In general, an equivariant function might decrease the least support, see Lemma 4.8.) The second additional result is stated in Lemma 6.7.

Lemma 6.7. *Suppose that $f : X \to X'$ is an equivariant function that preserves least supports. Then for every nominal set A, and every equivariant function*

$$\delta' : X' \times A \to X'$$

there exists a function

$$\delta : X \times A \to X$$

such that the following diagram commutes

$$\begin{array}{ccc} X \times A & \xrightarrow{\delta} & X \\ {\scriptstyle f \times Id_A}\downarrow & & \downarrow{\scriptstyle f} \\ X' \times A & \xrightarrow[\delta']{} & X' \end{array} \tag{6.2}$$

Proof. Let $Y_1, \ldots, Y_n$ be the orbits of the set $X \times A$.

Consider some $i \in \{1, \ldots, n\}$. Pick a representative $(x_i, a_i) \in Y_i$. In the diagram (6.2), follow the $f \times \mathrm{Id}_A$ arrow, and then δ', yielding an element

$$x'_i = \delta'(f(x_i), a_i).$$

Because the above element is the result of applying two equivariant functions, the least support of x'_i is a subset of the least support of (x_i, a_i). Because the function f is surjective, there must be some $y_i \in X$ such that

$$f(y_i) = x'_i = \delta'(f(x_i), a_i).$$

Because the function f preserves least supports, the least support of y_i is equal to the least support of x'_i, which is included in the least support of (x_i, a_i). It follows that there is an equivariant function

$$\delta_i : Y_i \to X \qquad \text{such that } \delta_i(x_i, a_i) = y_i.$$

Do the construction above for all orbits $Y_1, \ldots, Y_n$, yielding functions $\delta_1, \ldots, \delta_n$. Define δ to be the union of these functions. We now prove that the diagram (6.2) commutes.

Pick some $(x, a) \in X \times A$. Because

$$(x_1, a_1), \ldots, (x_n, a_n)$$

represent all orbits of $X \times A$, it follows that

$$(x, a) = (x \cdot \pi, a \cdot \pi) \qquad \text{for some } i \in \{1, \ldots, n\} \text{ and some } \pi \in G.$$

Following the down-right path in the diagram (6.2) from (x, a) yields

$$\delta'(f(x), a) = \delta'(f(x_i \cdot \pi), a_i \cdot \pi) = \delta'(f(x_i), a_i) \cdot \pi.$$

Following the right-down path in the diagram (6.2) from (x, a) yields

$$f(\delta(x, a)) = f(\delta_i(x), a)) = f(\delta_i(x_i \cdot \pi, a_i \cdot \pi)) = f(\delta_i(x_i, a_i)) \cdot \pi = f(y_i) \cdot \pi,$$

which means that the diagram commutes because $f(y_i) = \delta'(f(x_i), a_i)$.

□

We now prove Theorem 6.6. Let X' be the state space of the G-DFA from item 2, and let δ' be its transition function. Apply Corollary 6.5 to X', yielding a partial surjective equivariant function $f : X \to X'$ where

$$X = C \times (\mathbb{D} \cup \bot)^n.$$

Let $Y \subseteq X$ be the domain of f. Because f preserves least supports, we can apply Lemma 6.7 for f, yielding a transition function $\delta : Y \times A \to Y$. Extend δ to an equivariant function $X \times A \to X$ in an arbitrary way. The rest of the proof is the same as in Theorem 6.4, using Lemma 6.3.

7. Nominal context-free languages

In this section, we discuss context-free languages in the world of nominal sets. The discussion in the section works for any data symmetry that guarantees that orbit finite sets are preserved by Cartesian product. For the rest of Section 7, fix some such symmetry $(\mathbb{D}, G)$. Our aim is to repeat the classical proof that pushdown automata are equivalent to context-free grammars. This is doable in nominal sets, because the proof does not use the subset construction.

7.1. **Pushdown automata.** Similarly as in Section 3, the definition of a nominal pushdown automaton is the same as the classical definition, except the word "finite" is replaced by "orbit finite", and elements and subsets are required to be equivariant.

Definition 7.1. A *nominal pushdown automaton* $\mathcal{A}$ consists of

- an input alphabet A, which is an orbit finite nominal set;
- a set of states Q, which is an orbit finite nominal set;
- a stack alphabet Γ, which is an orbit finite nominal set;
- an initial state $q_I \in Q$, which is equivariant;
- an initial stack symbol $\gamma_I \in Q$, which is equivariant;
- a set of transitions

$$\delta \subseteq Q \times \Gamma \times (A \cup \epsilon) \times Q \times \Gamma^*$$

  which is orbit finite and equivariant.

Unlike for nominal finite automata, we need to add the condition that the set of transitions is orbit finite, as the set

$$Q \times \Gamma \times (A \cup \epsilon) \times Q \times \Gamma^*$$

is not orbit finite, because of the Γ^* component. The orbit finiteness restriction prohibits a set of rules which can push arbitrarily large words onto the stack, because all words in the same orbit of Γ^* have the same length.

The semantics of the pushdown automaton is defined in the standard way (we use acceptance through the empty stack). A configuration of the automaton is a pair in $Q \times \Gamma^*$, where the first coordinate represents the control state and the second coordinate represents the stack contents. One defines a relation

$$\delta^* \subseteq (Q \times \Gamma^*) \times A^* \times (Q \times \Gamma^*)$$

which says how to go from one configuration to another reading a given input word. It is easy to see that the relation δ^* is equivariant. It follows that the language recognized by the automaton, which is defined as

$$\{w : ((q_I, \gamma_I), w, (p, \epsilon)) \in \delta^* \text{ for some } p \in Q\}$$

is a G-language.

Example 7.2. For an orbit finite alphabet A, consider the language of palindromes:

$$P = \{a_1 a_2 \cdots a_n a_n \cdots a_2 a_1, a_1 a_2 \cdots a_n \cdots a_2 a_1 : a_1, \ldots, a_n \in A\} \subseteq A^*$$

This language is recognized by a nominal pushdown automaton which works exactly the same way as the usual automaton for palindromes, with the only difference that the stack alphabet Γ is now A. For instance, in the case when $A = \mathbb{D}$, the automaton keeps a stack of data values during its computation. The automaton has two control states: one for the first half of the input word, and one for the second half of the input word.

Example 7.3. The automaton in Example 7.2 had two control states. In some cases, it might be useful to have a set Q of control states that is orbit finite, but not finite. Consider for example the set of odd-length palindromes where the middle letter is equal to the first letter:

$$P' = \{a_1 a_2 \cdots a_n a_1 a_n \cdots a_2 a_1 : a_1, \ldots, a_n \in A\} \subseteq A^*.$$

A natural automaton recognizing this language would be similar to the automaton for palindromes, except that it would store the first letter a_1 in its control state.

Another solution would be an automaton which would keep the first letter in every token on the stack. This automaton would have a stack alphabet of $\Gamma = A \times A$, and after reading letters $a_1 \cdots a_n$ its stack would be

$$(a_1, a_1), (a_1, a_2), \ldots, (a_1, a_n).$$

This automaton would only need two control states. Actually, using the standard construction, one can show that every nominal pushdown automaton can be converted into one that has one control state, but a larger stack alphabet.

A register model. Consider the special case of the equality symmetry, and an alphabet of the form $A = A_{\text{fin}} \times \mathbb{D}$, for A_{fin} finite. Using the same kind of techniques as in Section 6, one can show that there is an equivalent 'concrete' model, in the spirit of finite memory automata. This model uses:

- for the states Q: a finite set of control states and a finite set of register names;
- for the stack alphabet Γ: a finite set of stack labels and a finite set of stack register names.

Transitions are in the spirit of finite memory automata.

7.2. **Context-Free Grammars.** Once more, the definition of a nominal context-free grammar is obtained from the standard definition by replacing 'finite' with 'orbit finite', and requiring elements and subsets to be equivariant.

Definition 7.4. A *nominal context-free grammar* $\mathcal{G}$ consists of

- an input alphabet A, which is an orbit finite nominal set;
- a set of nonterminals $\mathcal{N}$, which is an orbit finite nominal set;
- a starting nonterminal, which is equivariant;
- a set of productions
$$\mathcal{P} \subseteq \mathcal{N} \times (\mathcal{N} \cup A)^*$$
which is orbit finite and equivariant.

As usual, we assume that the sets A and $\mathcal{N}$ are disjoint.

Example 7.5. Consider the palindrome language P from Example 7.2. This language is generated by the following grammar with one nonterminal N:

$$N \to aNa \qquad \text{for every } a \in \mathbb{A}.$$
$$N \to \epsilon$$

Example 7.6. In the previous example, the grammar had just one nonterminal. Sometimes, it is useful to have an orbit finite, but infinite, set of nonterminals. Consider the language P' from Example 7.3. For this language, we need a different set of nonterminals, with a starting nonterminal N as well as one nonterminal N_a for every $a \in A$. The rules of the grammar are:

$$N \to aN_aa \qquad \text{for every } a \in \mathbb{A}.$$
$$N_a \to bN_ab \qquad \text{for every } b \in \mathbb{A}.$$
$$N_a \to a \qquad \text{for every } b \in \mathbb{A}.$$

One can show that the language P' cannot be recognized by a nominal context-free grammar which has a finite set of nonterminals. Otherwise, the language P' would have the following property, which it does not have:

> For every sufficiently long word $w \in P'$, there is a decomposition $w = w_1w_2w_3$, with w_2 and w_1w_3 nonempty, such that
> $$w_1(w_2 \cdot \pi)w_3 \in P' \qquad \text{for every } \pi \in G.$$

7.3. **Equivalence of pushdown automata and context-free grammars.** In the examples so far, the languages P and P' could be recognized by nominal pushdown automata, and generated by nominal context-free grammars. This is no coincidence, since the two models are equivalent, as stated below.

Theorem 7.7. *Let A be an orbit finite alphabet. The following conditions are equivalent for a language $L \subseteq A^*$:*

(1) *L is recognized by a nominal pushdown automaton.*
(2) *L is generated by a nominal context-free grammar.*

The proof is essentially the same as the standard proof for classical sets, and is only sketched below.

Lemma 7.8. *Every nominal context-free grammar is equivalent to one in Chomsky normal form, which means that all rules have the form:*

$$N \to N_1 N_2 \quad \text{or} \qquad N \to a \qquad \text{for nonterminals } N, N_1, N_2 \text{ and input letters } a.$$

From a context-free grammar to a pushdown automaton. The classical construction works. The automaton keeps a stack of nonterminals. It begins with just the starting nonterminal, and accepts when all nonterminals have been used up. In a single transition, it replaces the nonterminal on top of the stack by the result of applying a rule. Here it is useful to assume that the grammar is in Chomsky normal form.

From a pushdown automaton to a context-free grammar. Suppose that the pushdown automaton is as in Definition 7.1. The nonterminals of the grammar are going to be

$$\mathcal{N} = Q \times \Gamma \times Q.$$

The language generated by a nonterminal (p, γ, q) is going to be the set of words which take the automaton from a configuration with state p and γ on top of the stack, to another configuration with state p and γ on top of the stack, such that during the run the symbol γ is not removed from the stack. The set $\mathcal{N}$ is orbit finite, by assumption on orbit finite sets being preserved under Cartesian products. The rules of the grammar are as in the classical construction; it is easy to see that the set of rules is orbit finite.

8. Other models and perspectives

In Sections 3 and 5 we defined and studied the nominal version of finite automata. In Section 7, we defined the nominal version of context-free languages. This approach could be pursued for a wide variety of models, such as two-way automata, alternating automata (cf. [7]), Turing machines, Petri nets, and so on. In each case one has to be careful to see which of the classical constructions or equivalences work, and which of them fail. Here we announce some results, which the reader can fill in:

- Nominal two-way G-NFA (G-DFA) are more powerful than one-way G-NFA (G-DFA). For instance, the language
$$L = \{d_1 \cdots d_n : \ n \in \mathbb{N} \text{ and all the letters } d_1, \ldots, d_n \text{ are different}\} \subseteq \mathbb{D}^*$$
is recognized by a two-way G-DFA.
- Nominal alternating finite automata are more powerful than nominal nondeterministic finite automata. For instance, the language L mentioned above is recognized by a nominal alternating finite automaton. In the spirit of Section 6, one makes a connection between nominal alternating finite automata, and models of alternating register automata known in the literature [11, 12]. This connection is investigated in [7].
- Nondeterministic and deterministic Turing machines have the same expressive power. For every Fraïssé symmetry, the nominal version of the P = NP question has the same answer as the standard version of the P = NP question.

In the theory of automata, the algorithmic aspect is of central importance. For instance, for every nondeterministic finite automaton, one can test emptiness in polynomial time. Another example: given a deterministic finite automaton, one can compute the minimal automaton in polynomial time. What is the nominal equivalent of these algorithms?

How is the input represented? Is there a programming language? How is running time measured? What is polynomial time? A first step toward answering these questions is made in the paper [7]. The latter paper builds on finite representations of orbit finite sets, to be developed in the following sections of this paper.

Part 2. Finite representations of nominal sets and automata

We shall now provide finite representation results for nominal G-sets and equivariant functions. A general goal of this kind of results is to generalize the development of Section 6, where a concrete understanding of deterministic and nondeterministic G-automata for the equality symmetry was provided. In the following sections we shall prove a sequence of progressively more concrete representations, under certain assumptions on the underlying data symmetry.

9. G-set representation

We begin with well-known results from group theory, regarding the structure of arbitrary G-sets for any group G, and we indicate why orbit finite G-sets cannot be presented by finite means in general.

Important examples of G-sets are provided by subgroups of G and their coset spaces. For a subgroup $H \leq G$, a (right) *coset* of H is a set of the form

$$H\pi = \{\sigma\pi \mid \sigma \in H\} \subseteq G,$$

for some $\pi \in G$. Note that $H\pi = H\theta$ if and only if $\pi\theta^{-1} \in H$. Right cosets of H define a partition of G, and the set of all such cosets is denoted G/H^r.

We shall now show a well-known representation result for single-orbit G-sets as coset spaces of subgroups of G.

Definition 9.1. A *subgroup representation* of a G-set is a subgroup $H \leq G$. Its *semantics* is the set

$$[\![H]\!]^{\mathsf{c}} = G/H^r,$$

with a G-action defined by $(H\pi) \cdot \sigma = H(\pi\sigma)$.

Proposition 9.2. *(1) For each $H \leq G$, $[\![H]\!]^{\mathsf{c}}$ is a single-orbit G-set. (2) Every single-orbit G-set X is isomorphic to some $[\![H]\!]^{\mathsf{c}}$.*

Proof. For (1), first check that the G-action on $[\![H]\!]^{\mathsf{c}}$ is well-defined under the choice of π; indeed, $H\pi = H\pi'$ implies $H(\pi\sigma) = H(\pi'\sigma)$. Further, every $H\pi, H\sigma \in [\![H]\!]$ are in the same orbit since $H\pi = H\sigma \cdot (\sigma^{-1}\pi)$.

(2) is known in the literature as the *orbit-stabilizer theorem*. For any x in a G-set X, the group

$$G_x = \{\pi \in G \mid x \cdot \pi = x\} \leq G$$

is called the *stabilizer* of x.

To prove (2), put $H = G_x$ for any $x \in X$. Define $f : X \to [\![G_x]\!]^{\mathsf{c}}$ by $f(x \cdot \pi) = G_x\pi$. The function f is well defined: if $x \cdot \pi = x \cdot \sigma$ then $\pi\sigma^{-1} \in G_x$, hence $G_x\pi = G_x\sigma$. It is easy to check that f is equivariant. It is also a bijection. For injectivity, if $f(x \cdot \pi) = f(x \cdot \sigma)$, which means $G_x\pi = G_x\sigma$, then $\pi\sigma^{-1} \in G_x$, hence $x \cdot \sigma = (x \cdot \pi\sigma^{-1}) \cdot \sigma = x \cdot \pi$. For surjectivity of f, for any $\pi \in G$ there is $f(x \cdot \pi) = G_x\pi$. □

Recall from group theory that subgroups $H, K \leq G$ are called *conjugate* if $K = \pi H \pi^{-1}$ for some $\pi \in G$.

Proposition 9.3. *For any $H, K \leq G$, $[\![H]\!]^{\mathsf{c}}$ and $[\![K]\!]^{\mathsf{c}}$ are isomorphic if and only if H and K are conjugate.*

Proof. For the *if* part, assume $K = \pi H \pi^{-1}$ and define

$$f(H\sigma) = K\pi\sigma.$$

This is well defined as a function from $[\![H]\!]^{\mathsf{c}}$ to $[\![K]\!]^{\mathsf{c}}$: if $H\sigma = H\theta$ then $\sigma\theta^{-1} \in H$, therefore $\pi\sigma\theta^{-1}\pi^{-1} = \pi\sigma(\pi\theta)^{-1} \in K$, hence $K\pi\sigma = K\pi\theta$. Moreover, f is obviously equivariant by Definition 9.1, and the mapping $K\sigma \mapsto H\pi^{-1}\sigma$ is its inverse.

For the *only if* part, assume an equivariant isomorphism $f : [\![H]\!]^{\mathsf{c}} \to [\![K]\!]^{\mathsf{c}}$ and take any $\pi \in G$ such that $f(He) = K\pi$, for e the neutral element of G. Now, for any $\sigma \in H$ there is

$$K\pi\sigma = f(He) \cdot \sigma = f(H\sigma) = f(He) = K\pi$$

hence $\pi\sigma\pi^{-1} \in K$; as a result, $H \leq \pi K \pi^{-1}$. For f^{-1} the inverse of f, there is $f^{-1}(K\pi) = He$, therefore by equivariance, $f^{-1}(Ke) = H\pi^{-1}$ and by repeating the previous argument, $K \leq \pi^{-1}H\pi$, hence $\pi K \pi^{-1} \leq H$. As a result, $H = \pi K \pi^{-1}$ as required. □

The subgroup representation can be extended to a representation of equivariant functions from single orbit G-sets:

Proposition 9.4. *Let $X = [\![H]\!]^{\mathsf{c}}$ and let Y be a G-set. Equivariant functions from X to Y are in bijective correspondence with elements $y \in Y$ for which $H \leq G_y$.*

Proof. Given an equivariant function $f : X \to Y$, let y be the image under f of the coset $He \in X$. Equivariant functions can only increase stabilizers, so $H = G_{He} \leq G_y$. On the other hand, given $y \in Y$, define a function $f : X \to Y$ by $f(H\pi) = y \cdot \pi$. This is well-defined if $H \leq G_y$; indeed, if $H\pi = H\sigma$ then $\pi\sigma^{-1} \in H \subseteq G_y$, hence $y \cdot \pi = y \cdot \sigma$.

It is easy to check that the two above constructions are mutually inverse. □

Corollary 9.5. *Equivariant functions from $X = [\![H]\!]^{\mathsf{c}}$ to $Y = [\![K]\!]^{\mathsf{c}}$ are in bijective correspondence with those cosets $K\pi$ for which $\pi H \subseteq K\pi$.*

Proof. By Proposition 9.4 unfolding Definition 9.1. Notice that the stabilizer of $Ke \in [\![K]\!]^{\mathsf{c}}$ is K itself, and the stabilizer of $K\pi$ is the conjugate subgroup $\pi^{-1}K\pi$. The condition $H \leq \pi^{-1}K\pi$ obtained from Proposition 9.4 is equivalent to $\pi H \subseteq K\pi$. □

Proposition 9.2 provides a way to represent single-orbit G-sets by subgroups. Together with Corollary 9.5, this representation can be rephrased concisely as an equivalence of two categories. Denote by G-$\mathbf{Set}^{\mathbf{1}}$ the category of single-orbit G-sets and equivariant function between them.

Theorem 9.6. *For any group G, G-$\mathbf{Set}^{\mathbf{1}}$ is equivalent to a category with:*

- *as objects, subgroups $H \leq G$,*
- *as morphisms from H to K, cosets $K\pi$ such that $\pi H \subseteq K\pi$.*

This theorem was formulated in [26] with essentially the same proof as above. We include it here for completeness.

Thanks to Proposition 9.3, one could refine Theorem 9.6 and represent single-orbit G-sets not by subgroups of G, but by conjugacy classes of those subgroups. For the sake of simplicity we choose not to do so.

The representation can be extended from single-orbit to arbitrary G-sets. To this end, note that the action of G on a set X acts independently, and can be defined separately, on each orbit. Formally, every G-set X is isomorphic to the disjoint union of its orbits understood as single orbit G-sets. As a result, a G-set can be represented by a *family* of subgroups of G, and equivariant functions are represented as suitable families of functions.

The subgroup representation exhibits some structure in the world of G-sets and equivariant functions. At the same time, it implies that it is impossible to present all orbit finite G-sets by finite means, as we shall now demonstrate.

By Propositions 9.2 and 9.3, the following proposition proves Fact 4.1.

Proposition 9.7. *For a countably infinite $\mathbb{D}$ and $G = \mathrm{Sym}(\mathbb{D})$, there are uncountably many non-conjugate subgroups of G.*

Proof. Fix an arbitrary family of pairwise-disjoint subsets $C_p \subseteq \mathbb{D}$, indexed by prime numbers p, such that $|C_p| = p$ for any p. Then, fix a family of permutations π_p, indexed also by prime numbers, such that each π_p acts as identity on $\mathbb{D} \setminus C_p$, and as a permutation of order p on C_p. For any subset I of prime numbers, let the group $H_I \leq G$ be generated by the family $\{\pi_p : p \in I\}$. One easily observes that H_I contains an element of a prime order p if and only if $p \in I$.

On the other hand, is is easy to show that for conjugate subgroups $H, K \leq G$, if H contains an element of some finite order, then K contains an element of the same order. Therefore, if $I \neq I'$ then H_I and $H_{I'}$ are not conjugate, and there are uncountably many different choices of I. □

Open subgroups. We shall now restrict the subgroup representation to nominal G-sets. For any $C \subseteq \mathbb{D}$, define $G_C \leq G$ by:

$$G_C = \{\pi \in G \mid \pi(c) = c \text{ for all } c \in C\}. \tag{9.1}$$

In other words, the subgroup G_C is the intersection of all stabilizers G_c for $c \in C$, in $\mathbb{D}$ considered as a G-set.

Definition 9.8. A subgroup $H \leq G$ is *open* if $G_C \leq H$ for some finite $C \subseteq \mathbb{D}$. If this is the case, we say that C *supports* H.

The name "open" is justified by considering $G \leq \mathrm{Sym}(\mathbb{D})$ as a topological group. This technique is well known, see e.g. [19] for an application in the context of sheaf theory closely related to nominal sets. For any set $\mathbb{D}$, the set of permutations $G \subseteq \mathrm{Sym}(\mathbb{D})$ can be equipped with a topology with basis given by *C-neighborhoods* of all $\pi \in G$:

$$\mathcal{B}_C(\pi) = \{\sigma \in G \mid \sigma|_C = \pi|_C\}. \tag{9.2}$$

It is not difficult to check that a subgroup $H \leq G$ is an open subset with respect to this topology if and only if it satisfies Definition 9.8.

Open subgroups of G are linked to nominal G-sets via the following result.

Proposition 9.9. *A single-orbit G-set $[\![H]\!]^{\mathsf{c}}$ is nominal if and only if H is open in G.*

Proof. Unfolding the definitions, it is easy to see that in a G-set X, a subset $C \subseteq \mathbb{D}$ supports an element $x \in X$ if and only if $G_C \leq G_x$. Then use (the proof of) Proposition 9.2(2). □

The above proof also implies that the notions of support in Definitions 4.3 and 9.8 coincide along the representation function $[\![-]\!]^{\mathsf{c}}$. We shall use both notions as convenient.

It is now straightforward to restrict the subgroup representation of Definition 9.1: nominal G-sets are represented by open subgroups of G. The representation of equivariant functions from nominal sets remains as in Proposition 9.4. In categorical terms, Theorem 9.6 restricts to:

Theorem 9.10. *For data symmetry $(\mathbb{D}, G)$, the category G-**Nom**$^{\mathbf{1}}$ is equivalent to a category with:*

- *as objects, open subgroups $H \leq G$,*
- *as morphisms from H to K, cosets $K\pi$ such that $\pi H \subseteq K\pi$.*

Here, G-$\mathbf{Nom}^{\mathbf{1}}$ denotes the category of single-orbit nominal sets and equivariant functions.

10. Well-behaved symmetries

Open subgroups of permutation groups are rather abstract entities, and it is not at all clear how to represent them by finite means. Much more concrete representations can be obtained under certain assumptions on the data symmetry involved, as we shall now demonstrate.

10.1. **Least supports.** An element of a nominal set always has a minimal support with respect to inclusion. As shown in Example 4.7, there may be many incomparable minimal supports (which means that there is no least support). Minimal supports of the same element might even have different cardinalities, as illustrated by the following example.

Example 10.1. For a permutation $\pi \in \mathrm{Sym}(\mathbb{N})$, let $\pi^2 \in \mathrm{Sym}(\mathbb{N} \times \mathbb{N})$ be the permutation

$$\pi^2(n, m) = (\pi(n), \pi(m)).$$

Let $\mathbb{D} = \mathbb{N} \times \mathbb{N}$ and let $G \leq \mathrm{Sym}(\mathbb{D}) = \{\pi^2 : \pi \in \mathrm{Sym}(\mathbb{N})\}$. Essentially, G contains all permutations of $\mathbb{N}$, extended coordinate-wise to $\mathbb{N} \times \mathbb{N}$. Consider the set $\mathbb{D}$ a nominal G-set, with the canonical action of G. The pair $(0, 1)$ has three minimal supports: the singleton $\{(0, 1)\}$, the singleton $\{(1, 0)\}$, and the two-element set $\{(0, 0), (1, 1)\}$.

The following fact follows immediately from the development of Section 9:

Fact 10.2. A symmetry $(\mathbb{D}, G)$ admits least supports if and only if for every subgroup $H \leq G$ and for every finite $C, D \subseteq \mathbb{D}$, if $G_C \leq H$ and $G_D \leq H$ then $G_{C \cap D} \leq H$ (see (9.1)).

We now give a convenient sufficient and necessary condition for $(\mathbb{D}, G)$ admitting least supports. It is easy to check that

$$C \subseteq D \text{ implies } G_C \geq G_D$$

and, as a result, for all $C, D \subseteq \mathbb{D}$,

$$G_{C \cap D} \geq G_C + G_D,$$

where the right-hand side denotes the subgroup of G generated by the union of G_C and G_D, i.e., the smallest subgroup of G that contains G_C and G_D. The opposite subgroup inclusion shall guarantee least supports for open subgroups of G. In fact it is not necessary to compare both sides as groups, but merely to check containment of their single orbits of $\mathbb{D}$, in the special case when both $C \setminus D$ and $D \setminus C$ are singleton sets.

Theorem 10.3. *For any symmetry $(\mathbb{D}, G)$, the following conditions are equivalent:*

(1) *For all finite $E \subseteq \mathbb{D}$ and $c, d \in \mathbb{D} \setminus E$ such that $c \neq d$,*

$$c \cdot G_E \subseteq c \cdot \left(G_{E\cup\{c\}} + G_{E\cup\{d\}}\right).$$

(2) *$(\mathbb{D}, G)$ admits least supports, i.e., if $G_C \leq H$ and $G_D \leq H$ then $G_{C\cap D} \leq H$, for any $H \leq G$ and any finite $C, D \subseteq \mathbb{D}$.*

Proof. (2)$\Longrightarrow$(1) is easy: take $C = E\cup\{c\}$, $D = E\cup\{d\}$ and $H = G_{E\cup\{c\}} + G_{E\cup\{d\}}$. Clearly $G_C \leq H$ and $G_D \leq H$, so by (2), $G_E \leq H$, hence $c \cdot G_E \subseteq c \cdot H$ for any $c \in \mathbb{D}$.

For (1)$\Longrightarrow$(2), we shall assume (1) and prove (2) by induction on the size of the (finite) set $C \cup D$.

If $C \subseteq D$ or $D \subseteq C$, then $C \cap D = C$ or $C \cap D = D$ and the conclusion follows trivially. Otherwise, consider any $c \in C \setminus D$ and $d \in D \setminus C$; obviously $c \neq d$. Define

$$E = (C \cup D) \setminus \{c, d\}.$$

We have $C \subseteq E \cup \{c\}$ and $D \subseteq E \cup \{d\}$, so

$$G_{E\cup\{c\}} \leq G_C \leq H \qquad\qquad G_{E\cup\{d\}} \leq G_D \leq H.$$

We shall now prove that $G_E \leq H$. To this end, consider any $\pi \in G_E$. By (1), there exists a permutation

$$\tau = \sigma_1\theta_1\sigma_2\theta_2 \cdots \sigma_n\theta_n$$

such that all $\sigma_i \in G_{E\cup\{c\}}$, $\theta_i \in G_{E\cup\{d\}}$, and $\tau(c) = \pi(c)$. Since $G_{E\cup\{c\}} \leq H$ and $G_{D\cup\{d\}} \leq H$, all $\sigma_i, \theta_i \in H$, hence also $\tau \in H$.

On the other hand, clearly $G_{E\cup\{c\}} \leq G_E$ and $G_{E\cup\{d\}} \leq G_E$, so all $\sigma_i, \theta_i \in G_E$, therefore $\tau \in G_E$. As a result, $\tau\pi^{-1} \in G_E$. Since $\tau\pi^{-1}(c) = c$, we obtain $\tau\pi^{-1} \in G_{E\cup\{c\}}$, therefore $\tau\pi^{-1} \in H$. Together with $\tau \in H$ proved above, this gives $\pi \in H$. Thus we have proved $G_E \leq H$.

It is now easy to show that $G_{C\cap D} \leq H$. Indeed, $|C \cup E| = |C \cup D| - 1$, so by the inductive assumption for C and E, we have $G_{C\setminus\{c\}} \leq H$ (note that $C \setminus \{c\} = C \cap E$). Further, $|(C \setminus \{c\}) \cup D| = |C \cup D| - 1$, so $G_{C\cap D} \leq H$ (note that $(C \setminus \{c\}) \cap D = C \cap D$). □

As an application:

Corollary 10.4. *The equality symmetry admits least supports.*

Proof. Consider any finite $E \subseteq \mathbb{D}$ and $c, d \notin E$ such that $c \neq d$. Take any $e \in c \cdot G_E = \mathbb{D} \setminus E$. We need to show some $\pi \in G_{C\cup\{c\}} + G_{C\cup\{d\}}$ such that $\pi(c) = e$.

There are two cases to consider. If $e \neq d$, put $\pi = (c\ e) \in G_{C\cup\{d\}}$. If $e = d$, take some fresh $d' \notin E \cup \{c, d\}$ and put $\pi = \sigma\theta$, where

$$\sigma = (c\ d') \in G_{C\cup\{d\}} \qquad \text{and} \qquad \theta = (d\ d') \in G_{C\cup\{c\}}.$$

Then use Theorem 10.3. □

Corollary 10.4 was first proved by Gabbay and Pitts [15, Prop. 3.4].

Corollary 10.5. *The total order symmetry admits least supports.*

Proof. Consider any finite $E \subseteq \mathbb{D}$ and $c, d \notin E$ such that $c \neq d$. Let l be the greatest element of E smaller than c, and let h be the smallest element of E greater than c, assuming they both exist. (The cases where c is smaller/greater than all elements of E are similar). Then $c \cdot G_E$ is the open interval of rational numbers (l, h). Take any $e \in (l, h)$; without loss of generality assume that $e > c$. We need to show some $\pi \in G_{C\cup\{c\}} + G_{C\cup\{d\}}$ such that $\pi(c) = e$.

The only interesting case is $d \in (c, e]$. In this case, take some $d' \in (c, d)$ and put $\pi = \sigma\theta$, where

- σ is some monotone permutation that acts as identity on $(-\infty, l] \cup [d, +\infty)$ (so $\sigma \in G_{E\cup\{d\}}$) and such that $\sigma(c) = d'$,
- θ is some monotone permutation that acts as identity on $(\infty, c] \cup [h, +\infty)$ (so $\theta \in G_{E\cup\{c\}}$) and such that $\theta(d') = e$.

Then use Theorem 10.3. □

10.2. **Fungibility.** In general, even if $G \leq \mathrm{Sym}(\mathbb{D})$ admits least supports, not every finite subset of $\mathbb{D}$ is the least support of some open subgroup of G. We now characterize those subsets that are.

For any $C \subseteq D$ and $G \leq \mathrm{Sym}(\mathbb{D})$, the restriction of G to C is defined by

$$G|_C = \{\pi|_C \mid \pi \in G,\ C \cdot \pi = C\} \leq \mathrm{Sym}(C).$$

Clearly if $H \leq G$ then $H|_C \leq G|_C$. On the other hand, for $S \leq \mathrm{Sym}(C)$, the G-extension of S is

$$ext_G(S) = \{\pi \in G \mid \pi|_C \in S\} \leq G.$$

Definition 10.6. A finite set $C \subseteq \mathbb{D}$ is *fungible* (wrt. G) if for every $c \in C$ there exists a $\pi \in G$ such that:

- $\pi(c) \neq c$, and
- $\pi(c') = c'$ for all $c' \in C \setminus \{c\}$.

We say that a data symmetry $(\mathbb{D}, G)$ is fungible if every finite $C \subseteq D$ is fungible.

Example 10.7. The equality symmetry and the total order symmetry are both fungible. The integer symmetry is not fungible, as the set $\{1, 2\}$ is not fungible in it: if $\pi(1) = 1$ then necessarily $\pi(2) = 2$, for $\pi \in G$.

Lemma 10.8.

(1) *For any open $H \leq G$, if the least support of H exists then it is fungible.*
(2) *If $(\mathbb{D}, G)$ admits least supports then every finite fungible $C \subseteq \mathbb{D}$ is the least support of $ext_G(S)$, for any $S \leq \mathrm{Sym}(C)$.*
(3) *If $(\mathbb{D}, G)$ is fungible then every finite $C \subseteq \mathbb{D}$ is the least support of $ext_G(S)$, for any $S \leq \mathrm{Sym}(C)$.*

Proof. For (1), it is not difficult to check that if C is not fungible then $G_{C\setminus\{c\}} = G_C$ for some $c \in C$, therefore whenever C supports H so does $C \setminus \{c\}$.

For (2), first show that C supports $ext_G(S)$; indeed, $G_C = \{\pi \in G \mid \pi|_C = e|_C\} \subseteq ext_G(S)$. In this part fungibility is not used. Since $(\mathbb{D}, G)$ admits least supports, if C is not

the least support then there must be some support properly contained in it. However, if C is fungible then no $C\setminus\{c\}$ supports $ext_G(S)$; indeed, the permutation π from Definition 10.6 is a witness for $G_{C\setminus\{c\}} \not\leq ext_G(S)$. Since supports of a given group are always closed under supersets, no $C' \subsetneq C$ supports $ext_G(S)$.

Note that in (3) the existence of least supports is not assumed, so it does not follow immediately from (2). For a proof of (3), first show that C supports $ext_G(S)$ as in (2) above. Then assume another support D of $ext_G(S)$. We shall show that necessarily $C \subseteq D$. To this end, assume to the contrary that some $c \in C \setminus D$ exists. By the assumption on $(\mathbb{D}, G)$ the set $C \cup D$ is fungible, so the permutation π from Definition 10.6 is a witness for $G_{C\cup D\setminus\{c\}} \not\leq ext_G(S)$. But $G_{C\cup D\setminus\{c\}} \leq G_D$, so $G_D \not\leq ext_G(S)$, contradicting the assumption on D. □

In general, there is no implication between fungibility and the existence of least supports, as the following two examples show.

Example 10.9. Let $\mathbb{D}$ be a countably infinite set with a distinguished element d, and let G be the group of all permutations π of $\mathbb{D}$ such that $\pi(d) = d$. The symmetry $(\mathbb{D}, G)$ is not fungible, as the set $\{d, e\}$ is not fungible for any $e \neq d$. The fact that $(\mathbb{D}, G)$ admits least supports can be proved along the lines of Corollary 10.4.

Example 10.10. Let $\mathbb{D} = \{0, 1\} \times \mathbb{N}$, and let G be the group of all bijections π on $\mathbb{D}$ that either preserve the first components of all elements, or negate the first components of all elements. Such a permutation may be presented by a triple (a, π, σ) with $a \in \{0, 1\}$ and $\pi, \sigma \in \mathrm{Sym}(\mathbb{N})$, acting on $\mathbb{D}$ as follows:

$$(0, n) \mapsto (a, \pi(n)) \qquad (1, n) \mapsto (1 - a, \sigma(n))$$

It is easy to check that $\mathbb{D}$ is fungible. Now consider the set $X = \{0, 1\}$ with an action of G defined by:

$$0 \cdot (a, \pi, \sigma) = a \qquad 1 \cdot (a, \pi, \sigma) = 1 - a$$

Note that this action disregards the π and σ components of a permutation in G. Now, $0 \in X$ is supported by any singleton $\{(0, n)\} \subseteq \mathbb{D}$, but not by the empty set. As a result, $(\mathbb{D}, G)$ does not admit least supports.

10.3. **Support representation.** From now on, we assume a data symmetry $(\mathbb{D}, G)$ that admits least supports.

Definition 10.11. A *support representation* is a pair (C, S), where $C \subseteq \mathbb{D}$ is finite and fungible, and $S \leq G|_C$. Its *subgroup semantics* is

$$[\![C, S]\!]^{\mathsf{e}} = ext_G(S).$$

Proposition 10.12. *(1) $[\![C, S]\!]^{\mathsf{e}}$ is an open subgroup of G. (2) Every open subgroup $H \leq G$ is equal to some $[\![C, S]\!]^{\mathsf{e}}$.*

Proof. For (1), use Lemma 10.8(2). For (2), Put $S = H|_C$ where C is the least support of H; obviously $H|_C \leq G|_C$ since $H \leq G$, and C is fungible by Lemma 10.8(1). Then calculate

$$\begin{aligned} ext_G(H|_C) &= \{\pi \in G \mid \pi|_C \in H|_C\} = \{\pi \in G \mid \exists \sigma \in H.\ \pi|_C = \sigma|_C, C \cdot \sigma = C\} \\ &\overset{(*)}{=} \{\pi \in H \mid C \cdot \pi = C\} \overset{(**)}{=} H \end{aligned}$$

Step $(*)$ above is valid since C supports H, as $\pi|_C = \sigma|_C$ iff $\pi \in \mathcal{B}_C(\sigma) \subseteq H$ for $\sigma \in H$ (see (9.2)). For step $(**)$, check that for any $\pi \in G$,

$$\{\sigma^{-1} \mid \sigma \in \mathcal{B}_C(\pi)\} = \mathcal{B}_{C\cdot\pi}(\pi^{-1}).$$

This implies that if C supports H then so does $C \cdot \pi$, for any $\pi \in H$. Since C is the least support of H, there must be $C \subseteq C \cdot \pi$ and hence by finiteness, $C \cdot \pi = C$. □

In the following we shall use a simple characterization of the subgroup relation in terms of representations:

Lemma 10.13. *$[\![C, S]\!]^{\mathsf{e}} \leq [\![D, T]\!]^{\mathsf{e}}$ if and only if $D \subseteq C$ and $S|_D \leq T$.*

Proof. First we prove that $[\![C, S]\!]^{\mathsf{e}} \leq [\![D, T]\!]^{\mathsf{e}}$ implies $D \subseteq C$. Indeed, assuming the former, C supports $[\![D, T]\!]^{\mathsf{e}}$ (as it supports $[\![C, S]\!]^{\mathsf{e}}$). However, the least support of $[\![D, T]\!]^{\mathsf{e}}$ is D by Lemma 10.8(2), therefore $D \subseteq C$.

Then, assuming $D \subseteq C$, unfold the definitions and check

$$\begin{array}{c} [\![C, S]\!]^{\mathsf{e}} \leq [\![D, T]\!]^{\mathsf{e}} \\ \Updownarrow \\ \forall \pi \in G.\ \pi|_C \in S \Longrightarrow \pi|_D \in T \\ \Updownarrow \\ \forall \pi \in G.\ \pi|_C \in S \Longrightarrow (\pi|_C)|_D \in T \\ \Updownarrow \\ \forall \tau \in S.\ \tau|_D \in T; \end{array}$$

the last step uses the assumption that $S \leq G|_C$. □

We now compose representations 9.1 and 10.11 to represent single-orbit nominal G-sets in terms of least supports.

Definition 10.14. The *G-set semantics* $[\![C, S]\!]^{\mathsf{ec}}$ of a support representation (see Definition 10.11) is the set of those injective functions $u : C \to \mathbb{D}$ that extend to a permutation from G, quotiented by the equivalence relation:

$$u \equiv_S v \Leftrightarrow \exists \tau \in S.\ \tau u = v. \tag{10.1}$$

An action of G on $[\![C, S]\!]^{\mathsf{ec}}$ is defined by composition:

$$[u]_S \cdot \pi = [u\pi]_S.$$

Here and in the following, by $[u]_S$ we denote the equivalence class of u under $\equiv_S$.

Proposition 10.15. *(1) $[\![C, S]\!]^{\mathsf{ec}}$ is a single-orbit nominal G-set. (2) Every single-orbit nominal G-set X is isomorphic to some $[\![C, S]\!]^{\mathsf{ec}}$.*

Proof. Both parts easily follow from Propositions 9.2 and 10.12 once we prove that

$$[\![C, S]\!]^{\mathsf{ec}} \cong [\![[\![C, S]\!]^{\mathsf{e}}]\!]^{\mathsf{c}}. \tag{10.2}$$

For this we need an equivariant bijection between $[\![C, S]\!]^{\mathsf{ec}}$ and the set of cosets of $H = [\![C, S]\!]^{\mathsf{e}}$ in G.

To this end, map a coset $H\sigma$ to $[\sigma|_C]_S$; this is well-defined since C supports H. Conversely, for any $u : C \to \mathbb{D}$, map $[u]_S$ to $H\sigma$ where $\sigma \in G$ is such that $\sigma|_C = u$. This is again well-defined under the choice of σ since C supports H. To check that it is also well-defined under the choice of u from $[u]_S$, assume $\tau u = v$ for some $\tau \in S$. Since $H = ext_G(S)$, there is

some $\pi \in H$ such that $\pi|_C = \tau$. Then $\sigma|_C = u$ and $\theta|_C = v$ implies $(\pi\sigma)|_C = \theta|_C$, therefore (since C supports H) $H\sigma = H\pi\sigma = H\theta$.

Finally, it is easy to check that the two constructions are equivariant and mutually inverse. □

It is also possible to represent equivariant functions between G-sets represented via least supports.

Proposition 10.16. *Let $X = [\![C, S]\!]^{\mathsf{ec}}$ and $Y = [\![D, T]\!]^{\mathsf{ec}}$ be single-orbit nominal sets. Equivariant functions from X to Y are in bijective correspondence with those injective functions $u : D \to C$ that extend to a permutation from G, such that $uS \subseteq Tu$, quotiented by $\equiv_T$ (see (10.1)).*

Proof. By Proposition 9.4 and by (10.2), equivariant functions from X to Y bijectively correspond to those elements $[u]_T \in [\![D, T]\!]^{\mathsf{ec}}$ (i.e., injective functions $u : D \to \mathbb{D}$ that extend to permutations from G, quotiented by $\equiv_T$) for which the condition

$$[\![C, S]\!]^{\mathsf{e}} \leq G_{[u]_T} \tag{10.3}$$

holds. Considering $[u]_T$ as a right coset of $[\![C, K]\!]^{\mathsf{e}}$, it is easy to show that $G_{[u]_T} = \pi^{-1}[\![D, T]\!]^{\mathsf{e}}\pi$, for any $\pi \in G$ that extends u. Further, it is easy to check that $\pi^{-1}[\![D, T]\!]^{\mathsf{e}}\pi = [\![D \cdot u, u^{-1}Tu]\!]^{\mathsf{e}}$ (here note that $D \cdot u$ is fungible whenever D is). As a result, (10.3) is equivalent to

$$[\![C, S]\!]^{\mathsf{e}} \leq [\![D \cdot u, u^{-1}Tu]\!]^{\mathsf{e}}$$

and, by Lemma 10.13, to

$$D \cdot u \subseteq C \qquad \text{and} \qquad S|_{D \cdot u} \leq u^{-1}Tu.$$

Equivalently, u is an injection from D to C such that $uS \subseteq Tu$, as in the conclusion. □

As before, Propositions 10.15 and 10.16 can be phrased in the language of category theory, by analogy to Theorem 9.10:

Theorem 10.17. *For any data symmetry $(\mathbb{D}, G)$ which admits least supports, the category G-**Nom**$^{\mathbf{1}}$ is equivalent to a category with:*

- *as objects, pairs (C, S) where $C \subseteq \mathbb{D}$ is finite and fungible and $S \leq G|_C$,*
- *as morphisms from (C, S) to (D, T), those injective functions $u : D \to C$ that extend to permutations from G, such that $uS \subseteq Tu$, quotiented by $\equiv_T$.*

This representation is much more concrete than those of Theorems 9.6 or 9.10. Indeed, pairs (C, S) are finite entities, and equivariant functions are also represented by finite functions. As an immediate application, we obtain:

Corollary 10.18. *For any data symmetry $(\mathbb{D}, G)$ with $\mathbb{D}$ countable, which admits least supports, there are only countably many non-isomorphic single-orbit nominal G-sets.*

Proof. Since $\mathbb{D}$ is countable, it has only countably many finite subsets C. Moreover, for any C, there are only finitely many choices of $S \leq \mathrm{Sym}(C)$. □

To obtain an even more appealing representation, we shall now restrict attention to symmetries arising from certain classes of finite relational structures.

11. Fraïssé symmetries

Two of the key symmetries studied in this paper: the equality and the total order symmetry, arise from a general construction of a Fraïssé limit known from standard model theory, to be defined in this section.

11.1. **Fraïssé limits.** A *signature* is a set of relation names together with (finite) arities. We shall now consider relational structures over some fixed finite signature. For two relational structures $\mathfrak{A}$ and $\mathfrak{B}$, an *embedding* $f : \mathfrak{A} \to \mathfrak{B}$ in an injective function from the carrier of $\mathfrak{A}$ to the carrier of $\mathfrak{B}$ that preserves and reflects all relations in the signature.

Definition 11.1. A class $\mathcal{K}$ of finite structures over some fixed signature is called a *Fraïssé class* if it:

- is closed under isomorphisms and substructures,
- has *amalgamation*: if $f_{\mathfrak{B}} : \mathfrak{A} \to \mathfrak{B}$ and $f_{\mathfrak{C}} : \mathfrak{A} \to \mathfrak{C}$ are embeddings and $\mathfrak{A}, \mathfrak{B}, \mathfrak{C} \in \mathcal{K}$ then there is a structure $\mathfrak{D} \in \mathcal{K}$ together with two embeddings $g_{\mathfrak{B}} : \mathfrak{B} \to \mathfrak{D}$ and $g_{\mathfrak{C}} : \mathfrak{C} \to \mathfrak{D}$ that agree on the images of $f_{\mathfrak{B}}$ and $f_{\mathfrak{C}}$, i.e., $g_{\mathfrak{B}} f_{\mathfrak{B}} = g_{\mathfrak{C}} f_{\mathfrak{C}}$.

Example 11.2. Examples of Fraïssé classes include, over the empty signature:

- all finite structures, i.e., sets,
- sets of size at most 7,

and over the signature with a single binary relation symbol:

- all finite structures, i.e., directed graphs,
- undirected graphs, undirected trees,
- equivalence relations,
- preorders, partial orders, total orders,
- forest orders (i.e., partial orders such that for every x, the set of elements smaller than x is a total order).

Classes that are *not* Fraïssé due to lack of amalgamation include, over the signature with a single binary relation symbol:

- total orders of size at most 7,
- directed acyclic graphs,
- undirected forests (i.e., sets of disjoint trees),
- planar graphs.

The following theorem is standard in model theory (see e.g. [17]):

Theorem 11.3. *For any Fraïssé class $\mathcal{K}$ there exists a unique, up to isomorphism, countable* universal structure $\mathfrak{U}_{\mathcal{K}}$*, called the Fraïssé limit of $\mathcal{K}$, such that:*

- *the class of structures isomorphic to finite substructures of $\mathfrak{U}_{\mathcal{K}}$ is exactly $\mathcal{K}$, and*
- $\mathfrak{U}_{\mathcal{K}}$ *is* homogenous*, i.e., any isomorphism between two finite substructures of $\mathfrak{U}_{\mathcal{K}}$ extends (not necessarily uniquely) to an automorphism of $\mathfrak{U}_{\mathcal{K}}$.*

For the rest of this section, fix a Fraïssé class $\mathcal{K}$. From $\mathcal{K}$ we obtain a data symmetry $(\mathbb{D}_{\mathcal{K}}, G_{\mathcal{K}})$, where $\mathbb{D}_{\mathcal{K}}$ is the carrier of $\mathfrak{U}_{\mathcal{K}}$ and $G_{\mathcal{K}} = \mathrm{Aut}(\mathfrak{U}_{\mathcal{K}}) \leq \mathrm{Sym}(\mathbb{D}_{\mathcal{K}})$ is its group of automorphisms. We shall call a data symmetry of this form a *Fraïssé symmetry*.

Example 11.4. The equality and total order symmetries (see Example 2.3), are both Fraïssé symmetries; the former arises from the class of all finite sets, the latter from the class of finite total orders.

Other Fraïssé symmetries of interest include:

- The *graph symmetry*, arising from the class of finite undirected graphs. The universal undirected graph is the so-called random graph [24], where vertices are natural numbers, and an edge $\{x, y\}$ is present if and only if the x-th bit in the binary representation of y is 1 (for $x < y$). In the graph symmetry, $\mathbb{D}$ is therefore the set of natural numbers, and G is the automorphism group of the random graph.
- The *partial order symmetry*, arising from the class of finite partial orders. The universal structure $\mathfrak{U}_\mathcal{K}$ is not easily described in this case (see e.g. [18]), except that it is partially ordered and homogenous.

Definition 11.5. A Fraïssé symmetry $(\mathbb{D}_\mathcal{K}, G_\mathcal{K})$ is *well-behaved* if it admits least supports and is fungible.

All symmetries in Example 11.4 are well behaved. However, not every Fraïssé symmetry admits least supports or is fungible. Indeed, symmetries in Examples 10.9 and 10.10 are both Fraïssé. The one from Example 10.9 arises from the class of all finite sets with a possibly distinguished single element, i.e., relational structures over a signature with a single unary predicate P, such that at most one element satisfies P. The symmetry from Example 10.10 arises from the class of finite equivalence relations with at most two equivalence classes.

11.2. **Structure representation.** We shall now refine the nominal set representation provided in Section 10 for well-behaved Fraïssé symmetries. Looking at Definitions 10.11 and 10.14, from the properties of Fraïssé limits it is easy to form the following definition:

Definition 11.6. A *structure representation* is a finite structure $\mathfrak{A} \in \mathcal{K}$ (the *shape*) together with a group of automorphisms $S \leq \mathrm{Aut}(\mathfrak{A})$ (the *local symmetry*). Its *semantics* $[\![\mathfrak{A}, S]\!]$ is the set of embeddings $u : \mathfrak{A} \to \mathfrak{U}_\mathcal{K}$, quotiented by $\equiv_S$ (see 10.1). A $G_\mathcal{K}$-action on $[\![\mathfrak{A}, S]\!]$ is defined by composition of embeddings with automorphisms of $\mathfrak{U}_\mathcal{K}$.

Proposition 11.7. *(1) $[\![\mathfrak{A}, S]\!]$ is a single-orbit nominal $G_\mathcal{K}$-set. (2) Every single-orbit nominal $G_\mathcal{K}$-set X is isomorphic to some $[\![\mathfrak{A}, S]\!]$.*

Proof. Easy from Proposition 10.15. Indeed, compare Definitions 11.6 and 10.11 and notice that $\mathrm{Aut}\,\mathfrak{A} = (G_\mathcal{K})|_C$, where C is the carrier of $\mathfrak{A}$, as $\mathfrak{U}_\mathcal{K}$ is homogenous. Moreover, embeddings of $\mathfrak{A}$ into $\mathfrak{U}_\mathcal{K}$ are exactly those injective functions from C to $\mathbb{D}_\mathcal{K}$ that extend to automorphisms of $\mathfrak{U}_\mathcal{K}$. As a result, $[\![\mathfrak{A}, S]\!] = [\![C, S]\!]^{\mathsf{ec}}$. □

Equivariant functions get a similar characterization:

Proposition 11.8. *Let $X = [\![\mathfrak{A}, S]\!]$ and $Y = [\![\mathfrak{B}, T]\!]$ be single-orbit nominal sets. Equivariant functions from X to Y are in bijective correspondence with those embeddings $u : \mathfrak{B} \to \mathfrak{A}$ for which $uS \subseteq Tu$, quotiented by $\equiv_T$.*

Proof. Easy from Proposition 10.16. □

As before, this induces an equivalence of categories:

Theorem 11.9. *In a well-behaved Fraïssé symmetry, the category G-**Nom**^{1} is equivalent to a category with:*

- *as objects, pairs $(\mathfrak{A}, S)$ where $\mathfrak{A} \in \mathcal{K}$ and $S \leq \mathrm{Aut}(\mathfrak{A})$,*
- *as morphisms from $(\mathfrak{A}, S)$ to $(\mathfrak{B}, T)$, those embeddings $u : \mathfrak{B} \to \mathfrak{A}$ for which $uS \subseteq Tu$, quotiented by $\equiv_T$.*

For $\mathcal{K}$ the class of all finite sets, this gives rise to the category of "named sets with symmetries" studied in the theory of history-dependent automata. In this special case, Theorem 11.9 was proved in [16, 26].

11.3. **Representation of Cartesian products.** Nominal automata as studied in Section 5 are algebraic structures that involve equivariant functions, or relations, between nominal sets that are Cartesian products of other sets. In Sections 7 and 8, Cartesian products naturally appeared in algebraic presentations of other computation models as well. To present such models by finite means, it is therefore necessary to calculate Cartesian products of nominal sets in terms their representations. We shall now do this for the case of well-behaved Fraïssé symmetries.

In the case of the equality symmetry, a somewhat less concrete representation, in terms of minimal spans of representation morphisms, was provided in [9].

First, consider a Cartesian product of the form

$$[\![\mathfrak{A}, 1]\!] \times [\![\mathfrak{B}, 1]\!]$$

for some finite structures $\mathfrak{A}, \mathfrak{B} \in \mathcal{K}$, where both representation symmetries are trivial groups. Recall that $[\![\mathfrak{A}, 1]\!]$ is the set of embeddings $f : \mathfrak{A} \to \mathfrak{U}_\mathcal{K}$, with $G_\mathcal{K}$-action defined by $f \cdot \pi = \pi \circ f$, and similarly for $[\![\mathfrak{B}, 1]\!]$.

For any pair of embeddings $f : \mathfrak{A} \to \mathfrak{U}_\mathcal{K}$, $g : \mathfrak{B} \to \mathfrak{U}_\mathcal{K}$, consider a relation $\rho_{(f,g)}$ between the carriers A, B of $\mathfrak{A}, \mathfrak{B}$ defined by:

$$\rho_{(f,g)}(a, b) \quad \Leftrightarrow \quad f(a) = g(b). \tag{11.1}$$

Since both f and g are embeddings, $\rho_{(f,g)}$ is a partial isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$. This isomorphism is invariant under the action of $G_\mathcal{K}$ on pairs of embeddings:

$$\rho_{(f,g)\cdot\pi} = \rho_{(f,g)} \tag{11.2}$$

for all $\pi \in G_\mathcal{K}$. Indeed, calculate:

$$\rho_{(f,g)\cdot\pi}(a, b) \Leftrightarrow (f \cdot \pi)a = (g \cdot \pi)a \Leftrightarrow \pi(f(a)) = \pi(g(b)) \Leftrightarrow f(a) = g(b) \Leftrightarrow \rho_{(f,g)}(a, b).$$

For a partial bijection ρ between A and B, the *amalgamated sum* $A \cup_\rho B$ is the disjoint union of A and B quotiented by ρ, together with canonical injections

$$A \xrightarrow{\ i\ } A \cup_\rho B \xleftarrow{\ j\ } B. \tag{11.3}$$

To save space, $A \cup_{\rho_{(f,g)}} B$ will be denoted by $A \cup_{(f,g)} B$.

Define a function $\gamma_{(f,g)} : A \cup_{(f,g)} B \to \mathbb{D}$ by cases:

$$\gamma_{(f,g)}(i(a)) = f(a) \qquad\qquad \gamma_{(f,g)}(j(b)) = g(b). \tag{11.4}$$

This is well defined by definition of $A \cup_{(f,g)} B$. Moreover, obviously

$$\gamma_{(f,g)\cdot\pi} = \pi \circ \gamma_{(f,g)}. \tag{11.5}$$

Let $\mathfrak{C}_{(f,g)}$ be the unique relational structure on the carrier $A \cup_{(f,g)} B$ that makes $\gamma_{(f,g)}$ an embedding into $\mathfrak{U}_{\mathcal{K}}$. By universality of $\mathfrak{U}_{\mathcal{K}}$, we have $\mathfrak{C}_{(f,g)} \in \mathcal{K}$. From (11.5) it is clear that

$$\mathfrak{C}_{(f,g)\cdot\pi} = \mathfrak{C}_{(f,g)} \tag{11.6}$$

for any $\pi \in G_{\mathcal{K}}$. Also, it is easy to see that $i : \mathfrak{A} \to \mathfrak{C}_{(f,g)}$ and $j : \mathfrak{B} \to \mathfrak{C}_{(f,g)}$ are embeddings.

In sum, embeddings $f : \mathfrak{A} \to \mathfrak{U}_{\mathcal{K}}$ and $g : \mathfrak{B} \to \mathfrak{U}_{\mathcal{K}}$ determine:

- a partial isomorphism $\rho_{(f,g)}$ between $\mathfrak{A}$ and $\mathfrak{B}$,
- a relational structure $\mathfrak{C}_{(f,g)}$ on $A \cup_{(f,g)} B$,
- an embedding $\gamma_{(f,g)} : \mathfrak{C}_{(f,g)} \to \mathfrak{U}_{\mathcal{K}}$;

moreover, by (11.2) and (11.6), $\rho_{(f,g)}$ and $\mathfrak{C}_{(f,g)}$ are invariant under the $G_{\mathcal{K}}$-action on (f, g). As a result, we obtain an equivariant function

$$[\![\mathfrak{A}, 1]\!] \times [\![\mathfrak{B}, 1]\!] \longrightarrow \coprod_{\rho,\mathfrak{C}} [\![\mathfrak{C}, 1]\!] \tag{11.7}$$

where ρ ranges over partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$, and $\mathfrak{C} \in \mathcal{K}$ over those relational structures on $A \cup_\rho B$ that make the inclusions i, j in (11.3) embeddings. In other words, $(\rho, \mathfrak{C})$ ranges over the indexing set:

$$I_{\mathfrak{A},\mathfrak{B}} \;=\; \{(\rho_{(f,g)}, \mathfrak{C}_{(f,g)}) : f : \mathfrak{A} \to \mathfrak{U}_{\mathcal{K}} \text{ and } g : \mathfrak{B} \to \mathfrak{U}_{\mathcal{K}}\}. \tag{11.8}$$

It is not difficult to define an inverse to (11.7): given ρ and $\mathfrak{C}$, simply precompose embeddings $\gamma : \mathfrak{C} \to \mathfrak{U}_{\mathcal{K}}$ with $i : \mathfrak{A} \to \mathfrak{C}$ and $j : \mathfrak{B} \to \mathfrak{C}$. Routine calculation shows that both constructions are mutually inverse, therefore (11.7) is an isomorphism of nominal sets.

If the relational signature of $\mathcal{K}$ is finite and the class $\mathcal{K}$ has decidable membership, then the collection of all possible ρ and $\mathfrak{C}$ is finite and can be effectively enumerated. As a result, we have obtained a way to compute representations of Cartesian products of the form $[\![\mathfrak{A}, 1]\!] \times [\![\mathfrak{B}, 1]\!]$.

We now adapt the above reasoning to the general case

$$[\![\mathfrak{A}, S]\!] \times [\![\mathfrak{B}, T]\!]$$

for arbitrary $S \leq \mathrm{Aut}(\mathfrak{A})$ and $T \leq \mathrm{Aut}(\mathfrak{B})$.

First, consider an action of the product group $S^{op} \times T$ on the set of partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$ defined by:

$$(\rho \cdot (\sigma, \tau))(a, b) \quad \Leftrightarrow \quad \rho(\sigma(a), \tau^{-1}(b));$$

equivalently, with ρ considered as a partial isomorphism *from* $\mathfrak{A}$ *to* $\mathfrak{B}$, this can be written as

$$\rho \cdot (\sigma, \tau) = \tau \circ \rho \circ \sigma.$$

For any $\sigma \in S$ and $\tau \in T$, there is a bijection

$$m_{\sigma,\tau} : A \cup_\rho B \to A \cup_{\rho\cdot(\sigma,\tau)} B$$

given by:

$$m_{\sigma,\tau}(i(a)) = i(\sigma^{-1}(a)) \qquad\qquad m_{\sigma,\tau}(j(b)) = j(\tau(b)). \tag{11.9}$$

This is well defined; indeed, calculate:

$$i(a) = j(b) \Leftrightarrow \rho(a, b) \Leftrightarrow (\rho \cdot (\sigma, \tau))(\sigma^{-1}(a), \tau(b)) \Leftrightarrow i(\sigma^{-1}(a)) = j(\tau(b)).$$

For any relational structure $\mathfrak{C}$ on $A \cup_\rho B$, let $\mathfrak{C} \cdot (\sigma, \tau)$ be the unique structure on $A \cup_{\rho\cdot(\sigma,\tau)} B$ that makes $m_{\sigma,\tau}$ into an isomorphism.

It is easy to check that we thus obtain a group action of $S^{op} \times T$ on the indexing set (11.8) of the disjoint union in (11.7). Pick a family of representatives $(\underline{\rho}, \underline{\mathfrak{C}})$ for each orbit of this action. For any representative, where $\underline{\mathfrak{C}} \in \mathcal{K}$ is a structure on $A \cup_{\underline{\rho}} B$, let $S \Cup T \leq \mathrm{Aut}(\underline{\mathfrak{C}})$ be the group of all those automorphisms of $\underline{\mathfrak{C}}$ that, roughly speaking, restrict to S on $\mathfrak{A}$ and to T on $\mathfrak{B}$. Formally,

$$S \Cup T \quad = \quad i^{-1}Si \ \cap \ j^{-1}Tj.$$

The following theorem is a generalization of (11.7).

Theorem 11.10. *There is an equivariant isomorphism*

$$[\![\mathfrak{A}, S]\!] \times [\![\mathfrak{B}, T]\!] \quad \cong \quad \coprod_{\underline{\rho}, \underline{\mathfrak{C}}} [\![\underline{\mathfrak{C}}, S \Cup T]\!]$$

where $\underline{\rho}, \underline{\mathfrak{C}}$ range over the chosen representatives as above.

Proof. For a function from left to right, take any $[f]_S \in [\![\mathfrak{A}, S]\!]$ and $[g]_T \in [\![\mathfrak{A}, T]\!]$. The embeddings $f : \mathfrak{A} \to \mathfrak{U}_{\mathcal{K}}$ and $g : \mathfrak{B} \to \mathfrak{U}_{\mathcal{K}}$ determine a partial isomorphism $\rho_{(f,g)}$, a relational structure $\mathfrak{C}_{(f,g)}$ and an embedding $\gamma_{(f,g)} : \mathfrak{C}_{(f,g)} \to \mathfrak{U}_{\mathcal{K}}$ as before.

Let $(\underline{\rho}, \underline{\mathfrak{C}})$ be the chosen representative of the $(S^{op} \times T)$-orbit of $(\rho_{(f,g)}, \mathfrak{C}_{(f,g)})$. In particular, there exists some $\sigma \in S$ and $\tau \in T$ such that

$$\rho_{(f,g)} = \underline{\rho} \cdot (\sigma, \tau) \qquad\qquad \mathfrak{C}_{(f,g)} = \underline{\mathfrak{C}} \cdot (\sigma, \tau). \tag{11.10}$$

Define an embedding $\gamma : \underline{\mathfrak{C}} \to \mathfrak{U}_{\mathcal{K}}$ by:

$$\gamma = \gamma_{(f,g)} \circ m_{(\sigma,\tau)}. \tag{11.11}$$

There may be many possible choices of σ, τ that satisfy (11.10), and they may yield different γ. However, all these γ's are $\equiv_{S \Cup T}$-equivalent. To see this, assume

$$\underline{\rho} \cdot (\sigma, \tau) = \underline{\rho} \cdot (\sigma', \tau') \qquad\qquad \underline{\mathfrak{C}} \cdot (\sigma, \tau) = \underline{\mathfrak{C}} \cdot (\sigma', \tau');$$

then it is easy to check

$$m_{(\sigma,\tau)} = m_{(\sigma',\tau')} \circ m_{(\sigma'^{-1}\sigma, \tau\tau'^{-1})},$$

and $m_{(\sigma'^{-1}\sigma, \tau\tau'^{-1})}$ is an automorphism of $\underline{\mathfrak{C}}$ that restricts to $\sigma'^{-1}\sigma \in S$ on $\mathfrak{A}$ and to $\tau\tau'^{-1} \in T$ on $\mathfrak{B}$. As a result,

$$m_{(\sigma,\tau)} \equiv_{S \Cup T} m_{(\sigma',\tau')}.$$

Moreover, for γ in (11.11), $[\gamma]_{S \Cup T}$ does not depend on the choice of representatives $f \in [f]_S$ and $g \in [g]_T$. To see this, notice that for any $\sigma \in S$ and $\tau \in T$:

$$\gamma_{(f \circ \sigma, g \circ \tau)} = \gamma_{(f,g)} \circ m_{(\sigma^{-1}, \tau)}$$

by (11.4) and (11.9).

As a result, we obtain a function that maps the pair $([f]_S, [g]_T)$ to $\underline{\rho}$, $\underline{\mathfrak{C}}$ and $[\gamma]_{S \Cup T}$. Equivariance of this function is checked routinely. As before, its inverse is obtained by precomposing embeddings $h : \underline{\mathfrak{C}} \to \mathfrak{U}_{\mathcal{K}}$ with injections $i : \mathfrak{A} \to \mathfrak{C}$ and $j : \mathfrak{B} \to \mathfrak{B}$. Both constructions are well-defined and mutually inverse up to $\equiv_S$, $\equiv_T$ and $\equiv_{S \Cup T}$. □

Example 11.11. In the equality symmetry, where $\mathcal{K}$ is the class of finite sets, there are no nontrivial relational structures, i.e., every structure is simply its carrier. Let

$$\mathfrak{A} = \{x, y\} \qquad \mathfrak{B} = \{z\}.$$

By Definition 11.6, there is

$$[\![\mathfrak{A}, 1]\!] \cong \mathbb{D}^{(2)} \qquad [\![\mathfrak{B}, 1]\!] \cong \mathbb{D}$$

(see Example 2.4). There are three partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$:

$$\rho_1 = \{(x, z)\} \qquad \rho_2 = \{(y, z)\} \qquad \rho_3 = \emptyset,$$

with the corresponding amalgamated sums $A \cup_{\rho_i} B$ having 2, 2 and 3 elements, respectively. By (11.7), there is an isomorphism

$$\mathbb{D}^{(2)} \times \mathbb{D} \cong \mathbb{D}^{(2)} + \mathbb{D}^{(2)} + \mathbb{D}^{(3)}.$$

In elementary terms:

$$\{(c, d) \mid c \neq d\} \times \mathbb{D} = \{(c, d, e) \mid c \neq d = e\} + \{(c, d, e) \mid e = c \neq d\} + \{(c, d, e) \mid c \neq d \neq e\}.$$

In general, the product $\mathbb{D}^{(n)} \times \mathbb{D}$ has $n + 1$ orbits.

Now consider a local symmetry $S = \mathrm{Aut}(\mathfrak{A}) = \{1, (x\ y)\}$. By Definition 11.6, there is

$$[\![\mathfrak{A}, S]\!] \cong \binom{\mathbb{D}}{2}$$

(see Example 2.4). Partial isomorphisms ρ_1 and ρ_2 form an orbit under the action of $S^{op} \times 1$, therefore by Theorem 11.10, the product of $[\![A, S]\!]$ and $[\![B, 1]\!]$ has only two orbits:

$$\binom{\mathbb{D}}{2} \times \mathbb{D} \cong [\![\{x, y\}, 1]\!] + [\![\{x, y, z\}, S \uplus 1]\!] \cong \mathbb{D}^{(2)} + \{(\{x, y\}, z) \mid z \neq x, y\};$$

here $S \uplus 1 = \{1, (x\ y)\}$.

Example 11.12. In the total order symmetry, where $\mathcal{K}$ is the class of finite total orders, there are no nontrivial local symmetries S in representations, since the only automorphism of a finite total order is the identity. Let

$$\mathfrak{A} = \{x < y\} \qquad \mathfrak{B} = \{z\}.$$

By Definition 11.6, there is

$$[\![\mathfrak{A}, 1]\!] \cong \mathbb{D}^{(<2)} \qquad [\![\mathfrak{B}, 1]\!] \cong \mathbb{D}$$

(see Example 2.4). As in Example 11.11, there are three partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$:

$$\rho_1 = \{(x, z)\} \qquad \rho_2 = \{(y, z)\} \qquad \rho_3 = \emptyset.$$

However, in this case there are three different total orders on $A \cup_{\rho_3} B = \{x, y, z\}$ that embed $\mathfrak{A}$ and $\mathfrak{B}$:

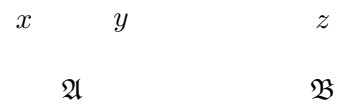
$$\mathfrak{C} = \{z < x < y\}, \qquad \mathfrak{C}' = \{x < z < y\}, \qquad \mathfrak{C}'' = \{x < y < z\},$$

each giving rise to a different orbit of the Cartesian product. As a result, there are five orbits:

$$\mathbb{D}^{(<2)} \times \mathbb{D} = \mathbb{D}^{(<2)} + \mathbb{D}^{(<2)} + \mathbb{D}^{(<3)} + \mathbb{D}^{(<3)} + \mathbb{D}^{(<3)}.$$

In general, the product $\mathbb{D}^{(<n)} \times \mathbb{D}$ has $2n + 1$ orbits.

Example 11.13. Consider the graph symmetry, where $\mathcal{K}$ be the class of all finite undirected graphs (see Example 11.4). By analogy to Example 11.11, let



$\mathfrak{A}$ $\mathfrak{B}$

be discrete graphs. There are three partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$:

$$\rho_1 = \{(x, z)\} \qquad \rho_2 = \{(y, z)\} \qquad \rho_3 = \emptyset,$$

with the corresponding amalgamated sums $A \cup_{\rho_i} B$ having 2, 2 and 3 elements, respectively. The sums corresponding to ρ_1 and ρ_2 have unique (discrete, isomorphic to $\mathfrak{A}$) graphs on them that embed $\mathfrak{A}$. The sum $A \cup_{\rho_3} B = \{x, y, z\}$ allows four graphs:

$\mathfrak{C}$ $\mathfrak{C}'$ $\mathfrak{C}''$ $\mathfrak{C}'''$

and as a result, the product $[\![\mathfrak{A}, 1]\!] \times [\![\mathfrak{B}, 1]\!]$ has six orbits:

$$[\![\mathfrak{A}, 1]\!] \times [\![\mathfrak{B}, 1]\!] = [\![\mathfrak{A}, 1]\!] + [\![\mathfrak{A}, 1]\!] + [\![\mathfrak{C}, 1]\!] + [\![\mathfrak{C}', 1]\!] + [\![\mathfrak{C}'', 1]\!] + [\![\mathfrak{C}''', 1]\!].$$

Now consider a local symmetry $S = \mathrm{Aut}(\mathfrak{A}) = \{1, (x\ y)\}$. This introduces three changes to the Cartesian product:

- Partial isomorphisms ρ_1 and ρ_2 form an orbit under the action of $S^{op} \times 1$,
- Structures $\mathfrak{C}'$ and $\mathfrak{C}''$, considered with ρ_3, are also in one orbit,
- The amalgamated groups $S \Cup 1$ on $\mathfrak{C}$ and $\mathfrak{C}'''$ are both nontrivial, $S \Cup 1 = \{1, (x\ y)\}$.

As a result, by Theorem 11.10, the product of $[\![\mathfrak{A}, S]\!]$ and $[\![\mathfrak{B}, 1]\!]$ has four orbits:

$$[\![\mathfrak{A}, S]\!] \times [\![\mathfrak{B}, 1]\!] = [\![\mathfrak{A}, 1]\!] + [\![\mathfrak{C}, S \Cup 1]\!] + [\![\mathfrak{C}', 1]\!] + [\![\mathfrak{C}''', S \Cup 1]\!].$$

12. Fraïssé automata

A deterministic orbit-finite nominal G-automaton, understood as in Section 5, is a simple combination of a few orbit-finite nominal G-sets and equivariant functions between them, involving a simple Cartesian product. It is therefore natural that an effective representation of nominal sets, equivariant functions and Cartesian products extends to a similar representation of automata. In this section we sketch the result of this extension for Fraïssé symmetries.

Fix for the rest of this section a class $\mathcal{K}$ of structures that induces a well-behaved Fraïssé symmetry $(\mathbb{D}_\mathcal{K}, G_\mathcal{K})$. Our goal is to apply Theorems 11.9 and 11.10 to develop a syntax (understood as a finite representation) for $G_\mathcal{K}$-DFA. At the risk of repeating some material from Section 11, we unravel below the definition of a deterministic orbit finite nominal $G_\mathcal{K}$-automaton.

For the sake of presentation, we only study automata over the alphabet $\mathbb{D}_\mathcal{K}$. The general case, when the alphabet is an arbitrary orbit finite nominal $G_\mathcal{K}$-set such as $(\mathbb{D}_\mathcal{K})^2$ or $\mathbb{D}_\mathcal{K} \uplus \mathbb{D}_\mathcal{K}$, may be dealt with in essentially the same way.

The basic intuition is that the class $\mathcal{K}$ describes all possible "memory shapes" of an automaton.

A Fraïssé $\mathcal{K}$-*automaton* has a finite set Q of states. Each state $q \in Q$ comes with a structure representation $(\mathfrak{A}_q, S_q)$. The set of configurations in state q is the nominal set $[\![\mathfrak{A}_q, S_q]\!]$. Elements of $\mathfrak{A}_q$ are called *registers* of state q, and the group S_q is the *register symmetry*. The set of all configurations of an automaton is:

$$X = \coprod_{q \in Q} [\![\mathfrak{A}_q, S_q]\!]. \tag{12.1}$$

A configuration consists of a state $q \in Q$, together with a valuation $\mathfrak{A}_q \to \mathfrak{U}_\mathcal{K}$ that maps registers to data values, and preserves and reflects the structure of $\mathfrak{A}_q$, with the proviso that valuations are considered equal if they differ only by a register symmetry.

The automaton has a set of accepting states, and an initial state. The structure of registers $\mathfrak{A}_q$ in the initial state must be empty.

The last ingredient of the Fraïssé automaton is a *symbolic transition function* $s = \{s_q\}_{q \in Q}$ that is used to represent an equivariant transition function

$$\delta_s : X \times \mathbb{D}_\mathcal{K} \to X. \tag{12.2}$$

The symbolic transition function is a representation of δ_s along the lines of Theorem 11.10 and Proposition 11.8. We define symbolic transition functions in terms of *annotations*, which enumerate orbits of the product $X \times \mathbb{D}_\mathcal{K}$ as explained in Theorem 11.10. An *annotation* of a representation $(\mathfrak{A}, S)$ is a structure of one of two kinds: either a conservative extension $\mathfrak{A}^* \in \mathcal{K}$ of $\mathfrak{A}$ by one element, denoted $*$; or the structure $\mathfrak{A}$ itself with additionally one distinguished element, that we denote by $*$ as well. In either case, we identify two annotations if they are related by an automorphism $\sigma \in S$ such that $\sigma(*) = *$. An annotation comes thus with its local symmetry, that is isomorphic either to the group S itself, or to its subgroup determined by the requirement $\sigma(*) = *$. There are finitely many possible annotations for every $\mathfrak{A}$, as the relational signature is assumed to be finite.

Intuitively speaking, annotations describe the ways in which the newly read input data value $(*)$ may compare to the data values in the registers. In other words, annotations formalize the tests an automaton on the input letters.

Note that an annotation of a structure $\mathfrak{A}$ uniquely determines:

- a partial isomorphism ρ between $\mathfrak{A}$ and a one-element structure $*$ (ρ is empty if the annotation extends $\mathfrak{A}$ with $*$, otherwise it identifies $*$ with the distinguished element of $\mathfrak{A}$),
- a relational structure on the amalgamated sum $A \cup_\rho \{*\}$.

In other words, by Theorem 11.10, annotations of $\mathfrak{A}_q$ correspond to orbits of the Cartesian product $[\![\mathfrak{A}_q, S_q]\!] \times \mathbb{D}_\mathcal{K}$.

The domain of s_q contains all possible annotations of $(\mathfrak{A}_q, S_q)$. For any annotation $\mathfrak{A}^*$, the value $s_q(\mathfrak{A}^*)$ is a state $p \in Q$ together with an embedding

$$s_q(\mathfrak{A}^*) : \mathfrak{A}_p \to \mathfrak{A}^* \tag{12.3}$$

that commutes with the local symmetries as prescribed by Proposition 11.8.

To sum up:

Definition 12.1. A Fraïssé $\mathcal{K}$-automaton consists of:

- a finite set of states Q;
- for each state $q \in Q$, a structure representation $(\mathfrak{A}_q, S_q)$ (see Definition 11.6);

- an initial state $q_I \in Q$ with $\mathfrak{A}_{q_I}$ the empty structure;
- a set of accepting states $F \subseteq Q$;
- a symbolic transition function $s = \{s_q\}_{q \in Q}$ as above.

Elements of $\mathfrak{A}_q$ are called *registers* of q.

These ingredients naturally induce a $G_{\mathcal{K}}$-automaton, with a transition function (12.2) defined as follows. Suppose that the state in the current configuration is $q \in Q$ and the valuation is represented, up to register symmetry, by $\eta : \mathfrak{A}_q \to \mathbb{D}_{\mathcal{K}}$. The automaton reads an input letter $d \in \mathbb{D}_{\mathcal{K}}$. Let η^* extend η by mapping $*$ to d, thus $\eta^* : \mathfrak{A}^* \to \mathbb{D}_{\mathcal{K}}$ is an embedding, for some annotation $\mathfrak{A}^* \in \mathcal{K}$. Apply s_q to $\mathfrak{A}^*$, yielding $p \in Q$ and a function (12.3). The new state is p, and the new valuation is obtained by composing $s_q(\mathfrak{A}^*)$ with the extended valuation η^*, that is $\eta^* \circ s_q(\mathfrak{A}^*) : \mathfrak{A}_p \to \mathbb{D}_{\mathcal{K}}$. The new valuation is an embedding, as a composition of embeddings, and its equivalence class depends only on the equivalence class of η, thanks to the assumption that s_q commutes with local symmetries.

Theorem 12.2. *For a well-behaved Fraïssé symmetry induced by a class $\mathcal{K}$, every reachable orbit finite deterministic nominal $G_{\mathcal{K}}$-automaton over the input alphabet $\mathbb{D}_{\mathcal{K}}$ is isomorphic to a Fraïssé $\mathcal{K}$-automaton.*

By Theorems 12.2 and 5.2 one directly obtains:

Corollary 12.3. *For a well-behaved Fraïssé symmetry induced by a class $\mathcal{K}$, the following conditions are equivalent for a $G_{\mathcal{K}}$-language $L \subseteq {\mathbb{D}_{\mathcal{K}}}^*$:*

(1) *L is recognized by a $G_{\mathcal{K}}$-DFA*
(2) *L is recognized by a Fraïssé $\mathcal{K}$-automaton*
(3) *the syntactic quotient ${\mathbb{D}_{\mathcal{K}}}^*/\equiv_L$ is orbit finite.*

Example 12.4. For the equality symmetry, Fraïssé $\mathcal{K}$-automata are very similar to finite memory automata studied in Section 6, with two differences:

- the number of registers varies from state to state (thus no need for undefined register values),
- symmetries are imposed on registers.

An even more similar model is that of history-dependent automata [23], where symmetries on local names were first introduced. For the equality symmetry, our Fraïssé $\mathcal{K}$-automata are essentially a deterministic version of history-dependent automata. A connection of the latter with finite memory automata has been tentatively made in [10].

For the total order symmetry, a $\mathcal{K}$-automaton has a totally ordered set of registers in each state, and valuations are monotonic. These automata are capable of comparing data values with respect to data ordering. It is easy to verify that Fraïssé $\mathcal{K}$-automata (and hence also $G_{\mathcal{K}}$-DFA, by Thm 12.2) in this case are expressively equivalent to deterministic finite memory automata of [3, 12] over totally ordered data, in the special case of a singleton alphabet.

For the graph symmetry, a $\mathcal{K}$-automaton keeps a graph of registers in each state, and valuations are graph embeddings into the random graph. An automaton can test a newly read letter for edge connections with nodes stored in current registers. To our best knowledge, this kind of automaton has not been studied in the literature.

References

[1] J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*. Cambridge Univ. Press, 1994.

[2] J. Adamek and V. Trnkova. *Automata and Algebras in Categories*. Kluwer Academic Publishers, 1990.

[3] M. Benedikt, C. Ley, and G. Puppis. What you must remember when processing data words. In *AMW*, 2010.

[4] H. Björklund and T. Schwentick. On notions of regularity for data languages. *TCS*, 411(4-5):702–715, 2010.

[5] M. Bojańczyk. Data monoids. In *STACS*, 2011.

[6] M. Bojańczyk, A. Muscholl, T. Schwentick, L. Segoufin, and C. David. Two-variable logic on words with data. In *LICS*, pages 7–16, 2006.

[7] Mikołaj Bojańczyk, Laurent Braud, Bartek Klin, and Sławomir Lasota. Towards nominal computation. In *Proc. POPL'12*, 2012. To appear.

[8] Mikołaj Bojańczyk, Bartek Klin, and Sławomir Lasota. Automata with group actions. In *Proc. LICS'11*, pages 355–364, 2011.

[9] V. Ciancia. *Accessible functors and final coalgebras for named sets*. PhD thesis, University of Pisa, 2008.

[10] V. Ciancia and E. Tuosto. A novel class of automata for languages on infinite alphabets. Technical Report CS-09-003, University of Leicester, 2009.

[11] Stéphane Demri and Ranko Lazic. Ltl with the freeze quantifier and register automata. *ACM Trans. Comput. Log.*, 10(3), 2009.

[12] D. Figueira, P. Hofman, and S. Lasota. Relating timed and register automata. In *Proc. EXPRESS'10*, volume 41 of *EPTCS*, pages 61–75, 2010.

[13] N. Francez and M. Kaminski. Finite-memory automata. *TCS*, 134(2):329–363, 1994.

[14] N. Francez and M. Kaminski. An algebraic characterization of deterministic regular languages over infinite alphabets. *TCS*, 306(1-3):155–175, 2003.

[15] M. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Asp. Comput.*, 13(3-5):341–363, 2002.

[16] F. Gadducci, M. Miculan, and U. Montanari. About permutation algebras, (pre)sheaves and named sets. *Higher-Order and Symbolic Computation*, 19(2-3):283–304, 2006.

[17] W. Hodges. *A shorter model theory*. Cambridge Univ. Press, 1997.

[18] J. Hubička and J. Nešetřil. Universal partial order represented by means of oriented trees and other simple graphs. *Eur. J. Comb.*, 26:765–778, 2005.

[19] S. Mac Lane and I. Moerdijk. *Sheaves in geometry and logic: a first introduction to topos theory*. Springer, 1992.

[20] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

[21] U. Montanari and M. Pistore. History-dependent automata: An introduction. In *SFM*, pages 1–28, 2005.

[22] F. Neven, T. Schwentick, and V. Vianu. Towards regular languages over infinite alphabets. In *MFCS*, pages 560–572, 2001.

[23] M. Pistore. *History Dependent Automata*. PhD thesis, University of Pisa, 1999.

[24] R. Rado. Universal graphs and universal functions. *Acta Arith.*, 9:331–340, 1964.

[25] L. Segoufin. Automata and logics for words and trees over an infinite alphabet. In *CSL*, pages 41–57, 2006.

[26] S. Staton. *Name-passing process calculi: operational models and structural operational semantics*. PhD thesis, University of Cambridge, 2007.