Bachelor thesis
Computer Science

Radboud University

Automata extended to nominal sets

Author:
Joep Veldhoven
s4456556

First supervisor/assessor:
Jurriaan Rot
jrot@cs.ru.nl

Second and third supervisor:
Herman Geuvers
Joshua Moerman
herman@cs.ru.nl
mail@joshuamoerman.nl

Second assessor:
Wieb Bosma
bosma@math.ru.nl

August 17, 2018

Abstract

In this thesis we will talk about the notion of languages and automata over nominal G-sets. First, we will give the definition of nominal G-sets and automata and languages over them. Then we will prove the Myhill-Nerode theorem for these automata and use it to show that determinization fails. We will introduce the definition of finite and infinite non-determinism which are possible different forms of non-determinism and pose interesting questions for further research. Lastly, we introduce a version of the pumping lemma for nominal automata which can be used to find some languages which are not accepted by automata in this theory.

Contents

Chapter 1

Introduction

Automata theory is the theory of languages and automata. A language is a collection of words over a certain alphabet and an automaton is a machine that takes a word as the input and then either accepts it or it rejects it. The collection of words that it accepts is a language. These automata and languages have several properties and applications based on these properties.

In classical automata theory a restriction is that the alphabet must be finite. It is interesting to look at languages and automata over infinite alphabets and see if they have stronger or new applications that can be used. In this thesis we will look at automata over nominal G-sets. This theory has been introduced in more papers, for example in Automata theory in nominal sets by Mikołaj Bojańczyk, Bartek Klin and Sławomir Lasota [2]. The idea of nominal sets is that they can be infinite, but they are finite up to some equivalence. This way we can have an infinite automaton over an infinite alphabet, but they are both finite up to some equivalence.

The aim of this thesis is to compare this theory of automata and languages over nominal G-sets to the classic theory and see what the differences and similarities are. We will try to go deeper into some more specific topics, like the pumping lemma and non-determinism. By doing so, we might find ways to extend known applications that work in the classical case to the nominal case and discover new applications which work specifically in the nominal case.

Chapter 2

Preliminaries

First we will give the definition of groups and group actions which are necessary for the following theory.

Definition 1. *A* group $(G, *)$ *consists of a non-empty set G and a binary function* $* : G \times G \to G$ *which satisfies the following group axioms.*

- *Closure: for all* a, b *in* G, $a * b$ *is also in* G.
- *Associativity: for all* a, b, c *in* G $(a * b) * c = a * (b * c)$.
- *Identity element: there is an element* e *in* G*, such that* $e*a = a = a*e$ *for all* a *in* G.
- *Inverse element: for all* a *in* G *there exists an* a^{-1} *in* G*, such that* $a * a^{-1} = e = a^{-1} * a$.

Sometimes we write ab instead of $a * b$.

Example 2. *The integers* $\mathbb{Z}$ *with the operation* $+$ *is a group with the identity element* 0*. The natural numbers* $\mathbb{N}$ *with the operation* $+$ *is not a group, because there are no inverse elements.*

Definition 3. *A* (right) action *of a group* G *on a set* X *is a function* $\cdot : X \times G \to X$*, which obeys the following rules:*

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

for $x \in X$ *and* $\pi, \sigma \in G$*, where* e *is the neutral element of* G*. Such a set* X *is called a* G-set.

Example 4. *Any set* X *is a* G*-set with the action* $x \cdot \pi = x$*. For any* G*-sets* X, Y *the Cartesian product* $X \times Y$ *is a* G*-set with the actions* $(x, y) \cdot \pi = (x \cdot \pi, y \cdot \pi)$*. We can see that this is indeed a* G*-set, because*

$$(x, y) \cdot e = (x \cdot e, y \cdot e) = (x, y)$$

and

$$\begin{aligned}(x, y) \cdot (\pi\sigma) &= (x \cdot (\pi\sigma), y \cdot (\pi\sigma)) \\ &= ((x \cdot \pi) \cdot \sigma, (y \cdot \pi) \cdot \sigma) \\ &= (x \cdot \pi, y \cdot \pi) \cdot \sigma \\ &= ((x, y) \cdot \pi) \cdot \sigma.\end{aligned}$$

Chapter 3

G-sets and automata

The theory and examples in this chapter are based on the paper Automata theory in nominal sets by Mikołaj Bojańczyk, Bartek Klin and Sławomir Lasota [2].

The following notion is used to create some simple G-sets, which can be used for interesting examples in the future.

Definition 5. *A* data symmetry $(\mathbb{D}, G)$ *is a collection of data values* $\mathbb{D}$ *with a subgroup* $G \subseteq$ *Sym*$(\mathbb{D})$*, the group of bijections from* $\mathbb{D}$ *to* $\mathbb{D}$.

Example 6. *The following are examples of data symmetries:*

- *Trivial symmetry:* $\mathbb{D} = \emptyset$ *and* G *is the trivial group with one element.*
- *Equality symmetry:* $\mathbb{D}$ *is a countably infinite set and* $G =$ *Sym*$(\mathbb{D})$*.*
- *Total order symmetry:* $\mathbb{D} = \mathbb{Q}$ *and* G *is the group of all monotone bijections.*
- *Integer symmetry:* $\mathbb{D} = \mathbb{Z}$ *and* G *is the group of all translations of the form* $i \mapsto i + c$.

Example 7. *For any data symmetry* $(\mathbb{D}, G)$, $\mathbb{D}$ *is a* G*-set with the action* $x \cdot \pi = \pi(c)$. *This action can also be extended to define the following* G*-sets:*

- $\mathbb{D}^n$*:* $(d_1, d_2 \ldots d_n) \cdot \pi = (\pi(d_1), \pi(d_2), \ldots, \pi(d_n))$.
- $\mathbb{D}^*$*:* $(d_1, d_2 \ldots) \cdot \pi = (\pi(d_1), \pi(d_2), \ldots)$.
- $\mathbb{D}^\omega$*:* $(d_1, d_2 \ldots) \cdot \pi = (\pi(d_1), \pi(d_2), \ldots)$
- $\mathcal{P}(\mathbb{D})$*:* $X \cdot \pi = \{\pi(x) \mid x \in X\}$

Definition 8. *The* orbit *of an element* x *of a* G*-set* X*is the set*

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\}.$$

It is possible that two elements have the same orbit. Because of this we can define the number of orbits of a G-set by the number of different orbits of all the elements of the G-set. A G-set is called orbit finite *if it has a finite number of orbits.*

The notion of orbit finiteness in G-sets can be compared to the notion of finiteness in the classical sets.

Example 9. *In the equality symmetry the orbit of any element x of a G-set X is the whole set X, because all the bijections can map x to any other element in X.*
In the total order symmetry the same holds, since all monotone bijections can still map any element to any other element. Since both of them have only one orbit, they are orbit finite.
The orbit of an element x of $\mathcal{P}(\mathbb{N})$ is the set of all elements with the same cardinality as x. So $\mathcal{P}(\mathbb{N})$ is not orbit finite.

It is useful to determine whether elements are in the same orbit. To get more insight into this we prove the following lemma.

Lemma 10. *Two elements x and y in a G-set X have the same orbit $\iff$ $\exists \sigma \in G$ such that $y = x \cdot \sigma$.*

Proof. $(\Rightarrow)$
Assume x and y have the same orbit. Now we have $y = y \cdot e \in y \cdot G = x \cdot G$, so there is a σ such that $x \cdot \sigma = y$.

$(\Leftarrow)$
Assume we have a σ, such that $y = x \cdot \sigma$. By definition for all the elements $a \in x \cdot G$ there exists aπ such that $a = x \cdot \pi$. Now we have

$$\begin{gathered} a = x \cdot \pi = \\ (y \cdot \sigma) \cdot \pi = \\ y \cdot (\sigma\pi). \end{gathered}$$

So $a \in y \cdot G$ and $x \cdot G \subseteq y \cdot G$. The same way we can see that $y \cdot G \subseteq x \cdot G$. Putting these two together we have the conclusion $x \cdot G = y \cdot G$. □

Corollary 11. *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.*

Proof. Because of lemma 10 two elements X, Y of the powerset have the same orbit iff there is a π such that $Y = X \cdot \pi$ iff $\pi(X) = Y$ iff there is a bijection π between X and Y iff X and Y have the same cardinality. □

From Corollary 11 we can see that in the equality symmetry the powerset of an infinite set is not orbit finite, because it has an infinite number of elements with different cardinalities. So in general the powerset of an orbit finite set is not orbit finite.

Definition 12. *A subset $Y \subseteq X$ is called* equivariant *if for any $\pi \in G$ the following holds:*

$$Y \cdot \pi = Y.$$

This definition can be extended to a relation $R \subseteq X \times Y$ or a relation of greater arity. A relation is called equivariant *if*

$$R \cdot \pi = R.$$

Extending this to functions we define that f is an equivariant function *if for all $x \in X$ and $\pi \in G$*

$$f(x) \cdot \pi = f(x \cdot \pi).$$

If Y is equivariant then Y is a union of orbits in X. The identity function on any set is an equivariant function. If f and g are equivariant then $g(f(x \cdot \pi)) = g(f(x) \cdot \pi) = g(f(x)) \cdot \pi$, so $g \circ f$ is also equivariant.

Example 13. *In the equality symmetry any subset of a G-set X can be mapped to any other subset of the same cardinality, so the only equivariant subsets of X are X and $\emptyset$. In $X \times X$ the three equivariant subsets are $X, \emptyset$ and $\{(x, x \mid x \in X\}$. Because of this the only equivariant function from X to X is the identity.*

$$id(x \cdot \pi) = x \cdot \pi = id(x) \cdot \pi.$$

Example 14. *The following are examples of equivariant functions:*

- *In the equality symmetry the only equivariant function from $\mathbb{D}$ to $\mathbb{D}$ is the identity. If we have another function f, with $f(d) = e$, then look at the π, which keeps d in place and swaps e with f. Then $f(d) \cdot \pi = e \cdot \pi = f \neq e = f(d) = f(d \cdot \pi)$.*

- *There are exactly two equivariant functions from $\mathbb{D}^2$ to $\mathbb{D}$, the projections. Any other function is not equivariant, for example the function f that maps (d, e) to a fixed letter a is not equivariant, because the π that only swaps b with a gives $f((d, e) \cdot \pi) = f(d, e) = a \neq b = a \cdot \pi = f((d, e)) \cdot \pi$.*

- *There is one equivariant function from $\mathbb{D}$ to $\mathbb{D}^3$, the function that maps d to (d, d, d). Any other function is not equivariant. For example the function f that maps a to (a, b, c) is not equivariant. Take the π that swaps a and b, $f(a \cdot \pi) = f(b) = (b, b, c) \neq (b, a, c) = (a, b, c) \cdot \pi = f(a) \cdot \pi$.*

- *There are more than three equivariant functions from $\mathbb{D}^3$ to $\mathbb{D}$. The projections are equivariant, but the function f, where $f((a, b, c)) =$*

$$\begin{cases} a & if\, a = b \\ c & if\, a \neq b \end{cases}$$

  is also equivariant. Because π is a bijection the image of a and b under π will be the same when a and b are the same and different when a and b are. In total there are 12 equivariant functions from $\mathbb{D}^3$ to $\mathbb{D}$. These are the three projections and nine functions similar to the one metioned above. For every two elements we have three of those functions, and we can compare two different elements three times.

Now we can finally define alphabets and languages over G-sets

Definition 15. *An* alphabet *is any orbit finite G-set A. The G-set A^* is the set of all words over the alphabet A. A G-*language *is any equivariant subset $L \subseteq A^*$.*

Lemma 16. *Concatenation over A^* is equivariant*

Proof. This means that for any two words w and v

$$(wv) \cdot \pi = (w \cdot \pi)(v \cdot \pi).$$

We have

$$\begin{aligned} (wv) \cdot \pi &= w_1 \cdot \pi \dots w_n \cdot \pi v_1 \cdot \pi \dots v_n \cdot \pi \\ &= (w \cdot \pi)(v \cdot \pi) \end{aligned}$$

□

Example 17. *The data symmetries $\mathbb{D}$ we have mentioned so far are alphabets, as well as any finite set Σ or the product $\Sigma \times \mathbb{D}$.*

Example 18. *Take a data symmetry $\mathbb{D}$ as the alphabet.*
In the equality symmetry the following are languages.

$$L_1 = \bigcup_{d,e \in \mathbb{D}} ded \qquad L_2 = \bigcup_{d \in \mathbb{D}} dd\mathbb{D}^* dd$$
$$L_3 = \{d_1 \dots d_n \mid n \geq 0, d_i \neq d_i + 1\}$$

Palindromes over $\mathbb{D}$ are a G-language as well. In the total order symmetry

$$\{d_1 \dots d_n \mid n \geq 0, d_1 > \dots > d_n\}$$

is a G-language. This is not a G-language in the equality symmetry, because a function π might not preserve the order of the elements, so the language is not equivariant. The language of all words that start with the specific letter a is not a G-language, because a function π might change the first letter, so the set of these words is not equivariant.

3.1 G-automata

We can now extend the standard notion of automata to the theory of G-sets. The notion of finiteness will be replaced by the notion of orbit finiteness and the components of the automata, like the initial and accepting states and the transition function, are required to be equivariant.

Definition 19. *A G-*automata *consists of the following elements.*

- *An input alphabet A.*
- *A G-set of states Q.*
- *Equivariant subsets I and F of Q, the initial and accepting states.*
- *An equivariant transition relation δ.*

It is called orbit finite *if Q is orbit finite.*

The single-step transition relation δ can be extended to a multi-step transition relation $\delta^* \subseteq Q \times A^* \times Q$ in the following way:

- $(q, \lambda, q) \in \delta^*$ for every $q \in Q$.
- $(q_1, wa, q_2) \in \delta^*$ if there exists a state $q_3 \in Q$ with $(q_1, w, q_3) \in \delta^*$ and $(q_3, a, q_2) \in \delta$.

A word w is accepted by such an automaton if $(q_I, w, q_F) \in \delta^*$ for some starting state $q_I \in I$ and accepting state $q_F \in F$. Because I, F and δ are equivariant so is the of words accepted by the G-automaton. Because of this, the set of words accepted by a G-automaton is a G-language.

Definition 20. *A G-automaton is called* deterministic *if there is only one initial state q_I and the transition relation is functional. That means*

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

is a well-defined function. The automaton is called reachable *if for every state $q \in Q$ there is a word $w \in A^*$, such that $\delta * (q_I, w) = q$.*

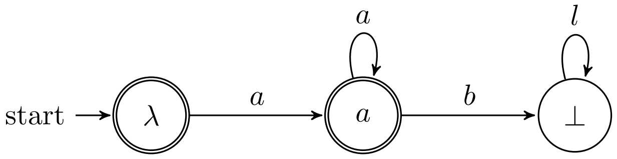
Example 21. *The language L_2 in Example 18 is accepted by the following deterministic G-automaton.*

- *A is a data symmetry $\mathbb{D}$ in the equality symmetry.*
- *$Q = \{\lambda, \bot\} \bigcup \mathbb{D}$.*
- *$I = \{\lambda\}, F = \mathbb{D} \cup \{\lambda\}$.*
- *δ is defined by the following function:*

$$\delta(\lambda, a) = a$$
$$\delta(a, a) = a$$
$$\delta(a, b) = \perp if\; a \neq b$$
$$\delta(\perp, l) = \perp.$$

This automaton has three orbits, $\lambda, \perp$ *and* $\mathbb{D}$*. The orbits* λ *and* $\perp$ *are singleton orbits with* $\lambda \cdot \pi = \lambda$ *and* $\perp \cdot \pi = \perp$ *for all* π*.* $\mathbb{D}$ *is an orbit of infinite size. This* δ *is equivariant, so this is a reachable deterministic orbit finite* G*-automaton.*

The following picture represents this automaton.



The a-state in the picture is not just one state, but represents the whole orbit $\mathbb{Q}$. The idea is that which state in the orbit the automata goes to is based on the letter it reads.

In the future we will call deterministic orbit finite G-automaton G-DFA's and non-deterministic orbit finite G-automaton will be called G-NFA's.

Chapter 4

Nominal G-sets and automata

The theory and examples in this chapter are based on the paper Automata theory in nominal sets by Mikołaj Bojańczyk, Bartek Klin and Sławomir Lasota [2].

A problem with orbit finiteness is that in general the Cartesian product does not preserve it.

Example 22. *In the equality symmetry $X \subseteq \mathcal{P}(\mathbb{Q})$, the set of all sets that are neither finite or cofinite, has only one orbit, and thus is orbit finite. This is the case because they all elements have the same cardinality, so there is a bijection between them. The product $X \times X$ is not orbit finite however. For any $n \in \mathbb{N}$ we can take an element $(C_n, D_n) \in X \times X$, where there are exactly n elements in C_n which are smaller than all elements in D_n. All these elements (C_n, D_n) are in a different orbit, so X^2 is not orbit finite.*

Since the Cartesian product is one of the most simple actions on sets this might be problematic and it is good to look at another, better structure on G-sets. We will introduce the notion of nominal G-sets. This notion focuses on data symmetries $(\mathbb{D}, G)$ and uses the action of G on $\mathbb{D}$ as well as the action of G on the G-set X. Note that these can actually be the same, if you take a data symmetry as a G-set itself, but in general they are different. From now on we will only use G-sets derived from data symmetries $(\mathbb{D}, G)$.

From now on we will only use G-sets derived from data symmetries $(\mathbb{D}, G)$.

Definition 23. *A set $C \subseteq \mathbb{D}$* supports *an element x of a G-set X if for every $\pi \in G$ that works as the identity on C, $x \cdot \pi = x$ holds. A G-set X is called* nominal *if every $x \in X$ has a finite support.*

When using nominal sets, we usually leave the set of data values $\mathbb{D}$ implicit and just talk about nominal G-sets.

Example 24. *For any data symmetry, $\mathbb{D}$ is a nominal G-set, because $\{d\} \subseteq \mathbb{D}$ support the element $d \in \mathbb{D}$. Similarly $\{d_1, \ldots, d_k\}$ supports $(d_1, \ldots, d_k)$, so $\mathbb{D}^k$ is a nominal G-set. The same holds for $\mathbb{D}^*$, but not for $\mathbb{D}^\omega$ or $\mathcal{P}(\mathbb{D})$, if $\mathbb{D}$ is infinite.*

Assume X and Y are nominal. If C supports an element $x \in X$ and D supports an element, then $C \cup D$ support $(x, y) \in X \times Y$, C supports $(x, 0) \in X + Y$ and D supports $(y, 1)$. So, then both $X \times Y$ and $X + Y$ are nominal sets as well.

Example 25. *If we take a data symmetry $\mathbb{D}$ as a G-set itself then every element $x \in \mathbb{D}$ is support by $\{x\}$. So $\mathbb{D}$ is nominal. The same way any element $(d_1, d_2, \ldots, d_n) \in \mathbb{D}^n$ is supported by $\{d_1, d_2, \ldots, d_n\}$, so $\mathbb{D}^n$ is a nominal G-set as well. This is also the case for $\mathbb{D}^*$, but not for $\mathcal{P}(\mathbb{D})$.*

Example 26. *Take $\mathcal{P}(\mathbb{N})$ derived from $(\mathbb{N}$,Sym$(\mathbb{N}))$ in the equality symmetry. The sets $\{1, 2\}$ and $\mathbb{N} \backslash 1, 2$ are supported by $\{1, 2\}$. If a set is finite, it is supported by that same set and if a set is cofinite it is supported by its complement. If it is neither finite nor cofinite, it does not have a finite support, so we see that in the power set of $\mathbb{N}$ the elements with finite support are the finite and cofinite sets.*

Example 27. *Now look at $\mathcal{P}(\mathbb{Q})$ derived from the total order symmetry. If a function π acts as the identity on a and b, with $a < b$ then for any c between a and b $\pi(a) < \pi(c) < \pi(b)$, so $a < \pi(c) < b$. This means that $[a, b] \cdot \pi = [a, b]$, so $\{a, b\}$ supports $[a, b]$. This also holds for open or half-open intervals. A finite boolean combination of these intervals also has a finite support, but any other set does not, so the sets with finite support are exactly the sets which are a finite boolean combination of intervals.*

Example 28. *Take a random G-set derived from the integer symmetry. If a function π works as the identity on an element $x \in \mathbb{Z}$, then $\pi(x) = x + z = x$ for some z, so $z = 0$ and π is the identity. This means that every singleton set supports every element of the G-set, so it is nominal.*

Now that we have introduced nominal sets, we show another useful property of equivariant functions, namely that they preserve supports.

Lemma 29. *If f is an equivariant function and $C \subseteq \mathbb{D}$ supports X, then C supports $f(X)$ as well.*

Proof. Assume π acts as the identity on C. Then we have that $x \cdot \pi = x$,because C support X. Using this fact and the equivariance of f we get $f(x) \cdot \pi = f(x \cdot \pi) = f(x)$, so C supports $f(X)$. □

4.1 Nominal G-automata

We can now use the nominal sets to extend the notion of G-automata to the nominal case.

Definition 30. *We call a G-automaton* nominal *if both the alphabet A as well as the set of states Q are nominal.*

We know already that if A is nominal, then the set of words A^* is so as well, because the word is supported by the union of all letters. Since a G-language is a subset of all the words, it is nominal as well.
We can now extend the Myhill-Nerode theorem to the nominal case. It uses the following notion of the Myhill-Nerode equivalence class.

4.2 Myhill-Nerode Theorem

Now we will extend the Myhill-Nerode theorem to the notion of automata over nominal G-sets to prove whether there are deterministic automata which accept certain languages.

Definition 31. *The Myhill-Nerode equivalence class can be defined for any alphabet A, so nominal ones as well. Consider two words $w, w' \in A^*$, they are equivalent over a G-language L if*

$$wv \in L \iff w'v \in L.$$

This equivalence is denoted by $w \equiv_L w'$. The equivalence class of a word is denoted by $w_{\equiv_L}$.

Lemma 32. *If L is equivariant, then $\equiv_L$ is equivariant as well.*

Proof. We have to prove that

$$w \equiv_L w' \text{ implies } w \cdot \pi \equiv_L w' \cdot \pi \text{ for any } \pi.$$

This means we have to proof that for any $v \in A^*$

$$(w \cdot \pi)v \in L \iff (w' \cdot \pi)v \in L.$$

Applying π^{-1} to both sides gives

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

Using the equivalence of L we get

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

Using Lemma 16 we get that

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

By using the equivalence of w and w' the lemma is now proven. □

Lemma 33. *Let X be a G-set and $R \subseteq X \times X$ is an equivariant equivalence relation. Then X/R is a G-set as well under the action*

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

We also have that the abstraction mapping

$$f : X \to X/R \qquad f(x) = [x]_R$$

is equivariant.

Proof. To show X/R is a G-set we show

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

and

$$\begin{aligned} [x]_R \cdot (\pi\sigma) &= [x \cdot (\pi\sigma)]_R \\ &= [(x \cdot \pi) \cdot \sigma]_R \\ &= [x \cdot \pi] \cdot \sigma \\ &= ([x] \cdot \pi) \cdot \sigma. \end{aligned}$$

So X/R is indeed a G-set.
For f we have

$$f(x) \cdot \pi = [x]_R \cdot \pi = [x \cdot \pi]_R = f(x \cdot \pi)$$

□

We have that $w \equiv_L w' \implies wa \equiv_L w'a$ for any $a \in A$. Using this we can define the following function on equivalence classes

$$\begin{gathered} \delta_L : A^*/\equiv_L \times A \to A^*/\equiv_L \\ \delta_L([w]_{\equiv_L}, a) = [wa]_{\equiv_L}. \end{gathered}$$

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

Definition 34. *Assume A is an orbit finite G-set and $L \subseteq A^*$ is a G-language. The* syntactic automaton *of L is defined as follows:*

- *The input alphabet is A.*
- *The set of states are the Myhill-Nerode equivalence classes of A^* under the L, $\equiv_L$.*
- *The initial state is the equivalence of the empty word $[\lambda]_{\equiv_L}$ and the accepting states are the equivalence classes of the words in L.*
- *The transition function is δ_L as defined above.*

Lemma 35. *The syntactic automaton of a G-language L is a reachable deterministic G-automaton.*

Proof. Since L is equivariant, $\equiv_L$ is so as well by Lemma 32. We can now apply Lemma 33 to get the following action of G on the equivalence classes $\equiv_L$.

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

So we see that the set of states is a G-set. We see that

$$[\lambda]_{\equiv_L} \cdot \pi = [\lambda \cdot \pi]_{\equiv_L} = [\lambda]_{\equiv_L},$$

so there is only one initial state. We also see that

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

So the set of accepting states is equivariant. Lastly we see that

$$\delta_L([w]_{\equiv_L}, a) \cdot \pi = [wa]_{\equiv_L} \cdot \pi = [(w \cdot \pi)(a \cdot \pi)]_{\equiv_L} = \delta_L([w]_{\equiv_L} \cdot \pi, a \cdot \pi,$$

which means that δ_L is equivariant, giving us all needed properties of a reachable deterministic G-automaton. □

Lemma 36. *Consider a reachable deterministic G-automaton in a data symmetry $(\mathbb{D}, G)$ over a nominal alphabet A. The set of states G of that automaton is nominal.*

Proof. Because the automaton is reachable, the function that maps $w \mapsto \delta * (q_I, w)$ is equivariant from A^* to Q. Since A^* is nominal we can apply Lemma 29 to see that Q is so as well. □

We will now define homomorphisms of automata. We use this to see the equivalence of nominal G-automata.

Definition 37. *Consider two deterministic G-automata*

$$H = (Q, A, q_I, F, \delta) \qquad H' = (Q', A, q'_I, F', \delta').$$

An equivariant function $f : Q \to Q'$ is called an automaton homomorphism *if $f(q_I) = q'_I$, it maps F to F':*

$$q \in F \textit{ iff } f(q) \in F' \textit{ for all } q \in Q$$

and the following holds

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

Assume f is an automata homomorphism between H and H'. If a word w is accepted by H, that means $\delta * (q_I, w) = q_F$ for some $q_F \in F$, $\delta'^*(f(q_I), w) = f(\delta'^*(q_I, w)) = f(q_F) \in F'$. This means that H and H' accept the same language. If there is a surjective homomorphism from H to H', H' is called the *homomorphic image* of H.

Lemma 38. *Consider a G-language L, the syntactic automaton of L is homomorphic image of any reachable deterministic G-automaton that recognizes L.*

Proof. Consider $H = (Q, A, q_I, F, \delta)$, a reachable deterministic G-automaton that recognizes L. Consider the mapping

$$\delta^*(q_I, w) \mapsto [w]_{\equiv_L}, w \in A^*.$$

We will show that this mapping f is an automaton homomorphism. It is toatl , because the automaton is reachable. It is well-defined, because $\delta^*(q_I, w) = \delta^*(q_I, v) \Rightarrow w \equiv_L v$. Because L is equivariant we can use Lemma 32 to see that $\equiv_L$ is equivariant and by using Lemma 33 we see that f itself is equivariant. We see that $q_I = \delta * (q_I, \lambda) = [\lambda]_{\equiv_L}$, so the initial state q_I is mapped to the initial state $[\lambda]_{\equiv_L}$. We see that $w \in L$ iff $\delta * (q_I, w)$ is accepting and $[w]_{\equiv_L}$ is accepting, so the accepting states are mapped to accepting states. Lastly we have to show f commutes with δ_L of the syntactic automaton. To show this we have:

$$f(\delta_L(q, a)) = [wa]_{\equiv_L}, \text{ where } \delta_L^*([\lambda]_{\equiv_L}, w) = q$$

and

$$\delta_L(f(q), a) = \delta_L(w, a) = [wa]_{\equiv_L}, \text{ where } \delta_L^*([\lambda]_{\equiv_L}, w) = q.$$

We see that these are equal, and thus f commutes with δ_L, so f is a homomorphism and the syntactic automaton is a homomorphic image of H. □

We can now finally state and prove the Myhill-Nerode theorem.

Theorem 39. *(Myhill-Nerode theorem for G-sets). Let A be an orbit finite G-set, and let $L \subseteq A^*$ be a G-language. Then the following two conditions are equivalent:*

1. *the set of equivalence classes of Myhill-Nerode equivalence $\equiv_L$ is orbit finite;*

2. *L is recognized by a nominal G-DFA.*

Proof. The implication 1 ⇒ 2 follows by combining Lemma 35 and Lemma 36. We see this, because L is recognized by its syntactic automaton and according to Lemma 35 it is deterministic and according to 36 it is nominal. For the implication 2 ⇒ 1 we have to see that if L is recognized by a G-automaton we can make it reachable by removing all unreachable states. Using this we can apply Lemma 38 to prove the implication. □

Chapter 5

Determinism

In the theory of automata over finite alphabets we have the powerset construction which transforms a non-determinstic automaton to a deterministic one. This also proves that in that case deterministic and non-determinstic automata accept the same languages. But we already saw that the powerset of an orbit finite set is not necessarily orbit finite. This means that applying the powerset construction to nominal G-NFA's gives a nominal deterministic automaton, but the state space is not necessarily orbit finite. This is only an intuitive idea to show that the powerset construction does not work, but there might be another method which does work.

In fact, it turns out that determinization fails and nominal G-NFA's are more powerful than nominal G-DFA's. To prove that nominal G-DFA's and nominal G-NFA's do not accept the same languages we will look at the language $L = \{wd \mid w \in A^*, d \in A, d \notin w\}$. First we will give a nominal G-NFA which accepts this language. Define $Q = A \cup \{\top, \bot, \lambda\}, I = \{\lambda\}$, and $F = \{\top\}$ with the transition function δ given by

$$\delta(\lambda, a) = \top$$

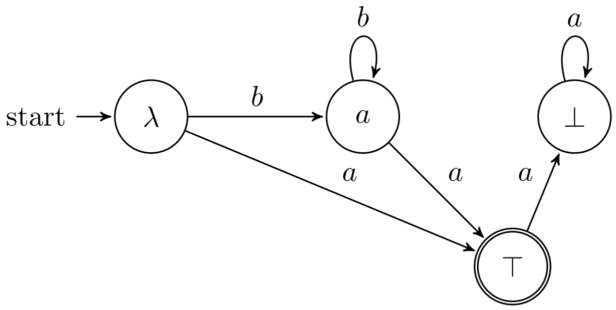
$$\delta(\lambda, b) = a, a \neq b$$

$$\delta(a, a) = \top$$

$$\delta(a, b) = a$$

$$\delta(\top, a) = \bot$$

$$\delta(\bot, a) = \bot$$



This machine first guesses the letter d and then checks if it is only seen as the last letter. It is clearly non-deterministic, the transition function is equivariant and Q has 4 orbits $\{\top\}, \{\bot\}, \{\lambda\}, \{a \mid a \in A\}$. So this automaton is orbit finite as well, and the language is accepted by a nominal G-NFA. To show it is not accepted by a nominal G-DFA we will use Theorem 39, the Myhill-Nerode Theorem for nominal G-sets. By proving the following lemma we can show L, as stated before, is not recognized by a nominal G-DFA.

Lemma 40. *For every $X \subseteq \mathcal{P}(A)$ there is a distinct equivalence class of $\equiv_L$, with L as above.*

Proof. For every X take a word w_X with all the elements of X contained exactly one time in the word and no other letters. The claim is that all these words are in different equivalence classes. Take two sets $X, Y \subseteq \mathcal{P}(A)$, if $X \neq Y$, then w.l.o.g there is an element $a \in X, a \notin Y$. Then $w_X a \notin L$, but $w_Y a \in L$. So $w_X \not\equiv w_Y$. □

Using this lemma we see that the language L has at least $|\mathcal{P}(A)|$ equivalence classes. So we see that if $\mathcal{P}(A)$ is not orbit finite, then the set of equivalence classes is not orbit finite either. By using the Myhill-Nerode theorem we can now see that L is not recognized by a nominal G-DFA.
Normally if we have a non-deterministic automaton for a language over a finite alphabet, and we want to check whether a word is in this language we can just calculate all the paths and check whether at least one of them ends in a final state. In a standard non-deterministic finite automaton the word is finite and since the amount of states is also finite so are the amount of transitions, there are also only a finite amount of paths. But when looking at nominal G-NFA's the amount of states can be infinite, so the number of transitions can be as well.

Definition 41. *We will call a nominal G-NFA* infinite non-deterministic *if there are a state q and a letter a such that $\delta(q, a)$, the set of all possible states we can get to, from q, by reading an a, is infinite. Else we will call the nominal G-NFA* finite non-deterministic.

Example 42. *The nominal G-NFA for the language $\mathcal{L} = \{wd \mid w \in A^*, d \notin w\}$ as seen before is infinite non-deterministic, because $\delta(\lambda, a) = \{b \in A, a \neq b\} \cup \top$, so if A is infinite, so is $\delta(\lambda, a)$ for every letter a.*

Fact 43. *For any nominal G-NFA $\delta(q, a)$ has a finite support.*

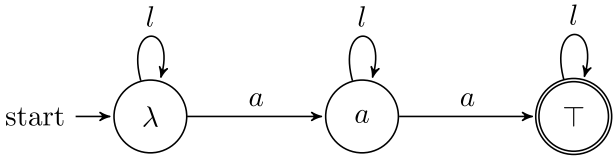
Proof. The function $\delta : Q \times A \to \mathcal{P}(Q)$ is equivariant and Q and A are nominal so $Q \times A$ is nominal as well. Now we apply Lemma 29, and we see that since (q, a) has a finite support, $\delta(q, a)$ does so as well. □

Example 44. *Take the language $L = \{w \mid \exists a$, which is in w at least 2 times$\}$ with the following nominal G-NFA: $Q = A \cup \{\lambda, \top\}, I = \{\lambda\}, F = \{\top\}$ and the transition function given by:*

$$\delta(\lambda, l) = \lambda$$

$$\delta(\lambda, a) = a$$
$$\delta(a, l) = a$$
$$\delta(a, a) = \top$$
$$\delta(\top, l) = \top$$

We can see that this nominal G-NFA is finite non-deterministic.



Remark 45. *The language in Example 44 is not accepted by a nominal G-DFA. This can be seen by using Theorem 39*

It is unclear whether these two sorts of non-determinism are actually different. If they actually are equivalent it would be interesting to find a method to transform one into the other.

Chapter 6

Pumping Lemma

In the classical theory of automata we have the pumping lemma to show that some languages are not regular. This lemma is based on the finiteness of the automata. In the nominal case the automata itself is not finite, but the amount of orbits is finite. With this we can introduce a pumping lemma for the nominal case.

Lemma 46. *(Pumping Lemma) If a language L is accepted by a nominal G-NFA, then there exists a k, such that for every $w \in L$, with $|w| \geq k$, w can be written as $w = xyz$, with $|xy| \leq k$, $|y| \geq 1$ such that $\exists \pi \in G, \forall n \geq 0$ the word $xy^n_\pi(z \cdot \pi^{n-1}) \in L$, where $y^0_\pi = \lambda$ and $y^i_\pi = y(y \cdot \pi)(y \cdot \pi^2) \cdots (y \cdot \pi^{i-1})$, for $i > 0$*

Proof. If there is a nominal G-NFA for L it is orbit finite, because of this we can take k as the number of orbits of the nominal G-NFA. If a word w of at least length k is in L, there is a path in the nominal G-NFA from an initial state to an accepting state. In this path there are at least k transitions, because the length of the word is at least k. Because of this, the path consists of at least $k+1$ states. Because the nominal G-NFA has k orbits there are two states in the first $k+1$ states of this path, which are in the same orbit. We call these states s and $s \cdot \pi$. Now we divide w into three parts $w = xyz$, and we choose x and y such that $s \in \delta^*(\lambda, x)$ and $s \cdot \pi \in \delta^*(\lambda, xy)$, where λ is the initial state of the path. This means $s \cdot \pi \in \delta^*(s, y)$. Because δ^* is equivariant we have that

$$\begin{aligned} \delta^*(s \cdot \pi^n, y \cdot \pi^n) &= \delta^*((s, y) \cdot \pi^n) \\ &= \delta^*(s, y) \cdot \pi^n \end{aligned}$$

Now combining this with the fact that $s \cdot \pi \in \delta^*(s, y)$ we get $s \cdot \pi \cdot \pi^n = s \cdot \pi^{n+1} \in \delta^*(s \cdot \pi^n, y \cdot \pi^n) \forall n \geq 0$. We also have

$$\begin{aligned} \delta^*(s \cdot \pi^{n+1}, z \cdot \pi^n) &= \delta^*((s \cdot \pi, z) \cdot \pi^n) \\ &= \delta^*(s \cdot \pi, z) \cdot \pi^n \end{aligned}$$

and

$$\begin{aligned}\delta^*(s, z\cdot\pi^{-1} &= \delta^*(s\cdot\pi\cdot\pi^{-1}, z\cdot\pi^{-1})\\ &= \delta^*((s\cdot\pi, z)\cdot\pi^{-1})\\ &= \delta^*(s\cdot\pi, z)\cdot\pi^{-1}.\end{aligned}$$

Since there is a final state $f \in \delta^*(s\cdot\pi, z)$ and the set of final states is equivariant, there is also a final state $f' \in \delta^*(s\cdot\pi^n, z\cdot\pi^{n-1})\forall n \geq 0$. So we now see that for all $n\,\delta^*(\lambda, xy_\pi^n(z\cdot\pi^{n+1})) = f'$ for a final state f'. We now have that for all $n \geq 0$ $xy_\pi^n \in L$ □

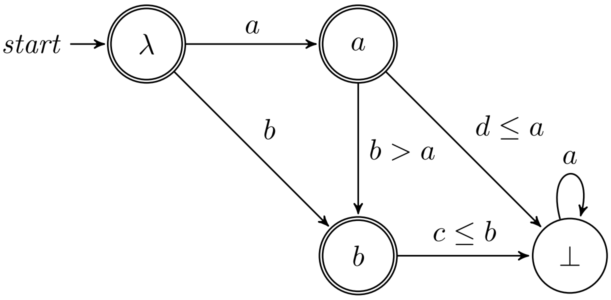
We can use this lemma to prove that some languages are not accepted by a nominal G-NFA.

Example 47. *$L = \{a^n b^n \mid a \neq b, n \geq 0\}$ is not accepted by a nominal G-NFA.*

Proof. If there would be such a G-NFA, then there would be a k for which the pumping lemma holds. Now look at the word $w = a^{k+1}b^{k+1}$ for an arbitrary a and b. If we write this w as xyz, regarding the rules in the pumping lemma, we have that $xy = a^n$ for some $n < k+1$ and $y = a^m$ for some $m \geq 1$ and $z = a^l b^{k+1}$. Now look at $xy_\pi^0(z\cdot\pi^{-1}) = a^{n-m}(a^l b^{k+1})\cdot\pi^{-1}$. The last $k+1$ letters of this word will be $b\cdot\pi^{-1}$, but the word is shorter than $2k+2$letters, so it is not possible that the word is of the form $a^n b^n$, so it is not in L, so there is no G-NFA for this language. □

Example 48. *In the total order symmetry the language $L = \{a_1, a_2, \ldots, a_n \mid a_i < a_{i+1}\forall i\}$, the collection of increasing sequences, is accepted by the following nominal G-DFA:*

- *$Q = \mathbb{Q} \cup \{\lambda, \perp\}$.*
- *$I = \lambda$.*
- *$F = \mathbb{Q} \cup \lambda$.*
- *δ is defined as follows:*
  $$\begin{aligned}\delta(\lambda, a) &= a\\ \delta(a, b) &= b \quad \text{if } a < b\\ \delta(a, b) &= \perp \quad \text{if } a \geq b\\ \delta(\perp, a) &= \perp\end{aligned}$$



The a- and b-state in the picture are just two representatives of all the states in the orbit $\mathbb{Q}$*. We see that* δ *is equivariant and the automata has 3 orbits, so it is orbit finite, we can see it is deterministic, so this is a nominal G-DFA. Now we can apply Lemma 46, the pumping lemma, to pump any word with sufficient length. As seen in the proof of the lemma a good choice of this length is the amount of orbits, so in this case 3. Take a word* w *of at least length 3. For example* $w = 1, 2, 5, 9, 14 \in L$*. Now we can divide* $w = xyz$*, with* $x = 1$*,* $y = 2, 5$ *and* $z = 9, 14$*.* $\delta(\lambda, 1) = 2, \delta(\lambda, xy) = 5$*. Now take* π *with* $x \mapsto x + 4$ *and* $n = 3$*. This is a good choice for* π *because* $1 \cdot \pi = 5$*. Now* $xy^3_\pi(z \cdot \pi^2) = 1, 2, 5, 2+4, 5+4, 2+4+4, 5+4+4, 9+4+4, 14+4+4 = 1, 2, 5, 6, 9, 10, 13, 17, 22$*, and we see this word is also in* L*.*

Lemma 49. *The language* $L = \{a_1, a_2 \cdots, a_n \mid a_i \neq a_j \text{ for } i \neq j\}$ *in the equality symmetry can be pumped as in the pumping Lemma.*

Proof. Assume we have a word $w = a_1, a_2 \cdots, a_n$ with at least length k where $a_i \neq a_j$ for $i \neq j$. Decompose $w = xyz$, we can now find a π with the property that for any letter a_i in y or z $a_i \cdot \pi^n$ not in w for any n and since π is a bijection and all a_i are different so will all the $a_i \cdot \pi^n$ be. Now $xy^n_\pi z \cdot \pi^{n-1}$ is a word with all different letters, so it is in L and the language can indeed be pummped. □

This is an example of a language that can be pumped, but is not accepted by a nominal G-NFA. From this we can deduce some interesting questions: Is there a pumping lemma which would exclude L in the above example? Is there a pumping lemma that would hold for nominal G-DFA's, but excludes standard examples of nominal G-NFA's that can not be determinized?

Chapter 7

Related Work

The notion of nominal G-automata has been the subject of some papers before, for example [2] on which a big part of the theory in this thesis is based. This paper has also proven the Myhill-Nerode theorem and shown that determinization fails. The notion of finite and infinite non-determinism is a new one and this might be an interesting topic for further research. The pumping lemma has been investigated before. Some of the authors of [2] have given a lecture on the subject and asked the following question "Show the following pumping lemma. Let A be non-deterministic register automaton. One can compute a constant M such that if the automaton accepts a word of length at least M, then it also accepts words of arbitrarily large lengths" [1]. This is a weaker version of the pumping lemma and follows immediately from Lemma 46. In the paper Regular and context-free nominal traces by Pierpaolo Degano, Gian-Luigi Ferrari and Gianluca Mezzetti [3] the authors talk about a de-pumping lemma instead of a pumping lemma. The idea of this lemma is similar to the one in this thesis, but it is not used a stand-alone lemma for finding properties of nominal G-NFA's, but rather as a supportive lemma for another theorem. The following blog post [4] by Bartek Klin, one of the writers of [2], considers a pumping lemma exactly as the one given in this thesis. It also asks some of the same questions about this subject as the thesis. This post was only discovered after this thesis was almost finished.

Chapter 8

Conclusions

In this thesis we have shown that determinization fails for nominal G-NFA's and introduced possibly different forms of non-determinism, infinite and finite non-determinism. For further research the difference in the finite and infinite non-determinism can be an interesting subject. The question is whether they are actually different. It is interesting whether a proof can be found for them being different or equal. In the case they are equal it is also interesting to find a construction which transforms an infinite non-deterministic automaton into a finite non-deterministic one.

We also introduced a pumping lemma for nominal G-NFA's. It might be interesting to look at improving the lemma. The lemma mentions an arbitrary π and it might be possible to put more restrictions on this arbitrary π. Maybe this way a lemma can be found that only accepts languages for which there are a nominal G-NFA or a version that works for deterministic automata but not for non-deterministic ones.

Bibliography

[1] Mikołaj Bojańczyk. Exercises on nominal sets: `https://www.mimuw.edu.pl/~bojan/zajecia/phdopen/phdopen.pdf`.

[2] Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Logical Methods in Computer Science*, 10(3), 2014.

[3] Pierpaolo Degano, Gian Luigi Ferrari, and Gianluca Mezzetti. Regular and context-free nominal traces. *Acta Inf.*, 54(4):399–433, 2017.

[4] Bartek Klin. A pumping lemma for automata with atoms: `http://atoms.mimuw.edu.pl/?p=43`.