Bachelor thesis
Computing Science

Radboud University

Exploring the difference between 2DFA and DFA for G-automata

Author:
Alex van der Hulst
s1041239

First supervisor/assessor:
dr. J.C. Rot
j.rot@cs.ru.nl

Second assessor:
prof. dr. J.H. Geuvers
h.geuvers@cs.ru.nl

April 2, 2022

Abstract

In this thesis, we introduce the notions of the G-automaton and two-way deterministic finite automata (2DFA) as in [1] and [6]. We show that 2DFA and DFA accept the same languages and that this is not the case for G-2DFA and G-DFA. A condition is given for which the construction from 2DFA to DFA also works for the construction from G-2DFA to G-DFA. Lastly, we zoom in on an example that does not satisfy this condition and for which the construction does not work.

Contents

Chapter 1

Introduction

Automata theory is a branch of computing science that studies abstract machines and their capabilities. Automata theory was implemented in areas such as compiler construction, circuit design, string matching, communication protocols and program verification [5]. In this paper, we will look at an automaton model that has an infinite alphabet, and that uses a group action to preserve the structure of finite states and transitions that DFA and NFA have. A non-deterministic finite automaton (NFA) is a finite state machine that accepts words based on the computation in the automaton. For deterministic finite automata (DFA) and two way deterministic finite automata (2DFA), this computation is unique. The additional property that 2DFA have, is that they can go back and forth in the word during the computation, in contrast with DFA and NFA, which go through the word from left to right.

There already exist some models of automata that can use infinite alphabets, such as, finite memory automata and data automata [4][2]. In this paper, we look at a model called G-automata that was introduced in [1].

This automaton model uses group actions to represent an infinite alphabet and infinite amount of states in a finite way. This allows us to use known theorems for DFAs, slightly adjust them and use them for G-automata, such as the Myhill-Nerode theorem.

Another piece of inspiration for this paper is a construction introduced in [6] that shows that 2DFA and DFA are expressively equivalent, which means that they accept the same languages. We will give a proof of this statement with the construction and look at when this construction works for the G-automaton variant: G-2DFA and G-DFA. The construction can not work all the time as shown in [1].

This thesis introduces definitions and examples about G-automata and 2DFA that should be more accessible for students than the original sources.

After this, we give a condition for which we can make a G-DFA that accepts the same language as a given G-2DFA.

We now give a quick overview of the content. In chapter 3, we will look at G-sets, the notion of equivariance and the G-automaton. Chapter 4 will introduce 2DFA, with an example for which we work out the construction to obtain a DFA that accepts the same language. Lastly, chapter 5 introduces the notion of a G-2DFA and shows with an extra condition that the construction used in chapter 4 works for G-2DFA to obtain a G-DFA.

Chapter 2

Preliminaries

In this chapter, we will provide some information that is required to read this paper that the reader might not know yet. We first introduce definition about group theory, followed by definitions from set theory and automata theory.

2.1 Group theory

Definition 2.1. A group is a set G with a binary operation $\cdot : G \times G \to G$, that satisfies the following requirements.

(G1) For all $a, b, c \in G$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.

(G2) There exists an $e \in G$ with for all $a \in G$, $e \cdot a = a = a \cdot e$.

Such an e is called the neutral element of G.

(G3) For each $a \in G$, there exists a $b \in G$ that satisfies $a \cdot b = e = b \cdot a$.

Note that we can say *the* neutral element since it is unique by the following. Suppose e and e' both satisfy (G2), then we have $e = e \cdot e' = e'$. Thus we can conclude $e = e'$, which means that e is unique.

In this thesis, we will mainly use a group that consists of bijections on a set X called $\mathrm{Sym}(X)$. The binary operation is defined by $\pi \cdot \sigma \coloneqq \pi \circ \sigma$. The symbol $\circ$ means that we use function composition, so $(\pi \circ \sigma)(x) = \pi(\sigma(x))$ for $x \in X$.

Theorem 2.2. *Sym(X) is a group, where* $e = id_X$*.*

Proof. (G1) Take $\pi, \sigma, \tau \in Sym(X)$ and $x \in X$, then we have:

$$\begin{aligned} &((\pi \circ \sigma) \circ \tau)(x) \\ &= \pi(\sigma(\tau(x))) \\ &= (\pi \circ (\sigma \circ \tau))(x). \end{aligned}$$

Thus we have $(\pi \circ \sigma) \circ \tau = \pi \circ (\sigma \circ \tau)$

(G2) Take $\pi \in G$ and $x \in X$, then we have for the identity function on X, id_X:

$$\begin{aligned}
&(\pi \circ id_X)(x) \\
&= \pi(id_X(x)) \\
&= \pi(x) \\
&= id_X(\pi(x)) \\
&= (id_X \circ \pi)(x).
\end{aligned}$$

We can note that id_X is also a bijection and thus there exists $id_X \in G$ with for all $\pi \in G$, $\pi \cdot id_X = \pi = id_X \cdot \pi$.

(G3) Take $\pi \in G$. Since π is a bijection, there exists an inverse of π, that we will call π^{-1}. This inverse function π^{-1} is also a bijection, since π is a bijection and thus $\pi^{-1} \in G$. Now we have:

$$\begin{aligned}
&\pi \circ \pi^{-1} \\
&= id_X \\
&= \pi^{-1} \circ \pi.
\end{aligned}$$

So, we have for $\pi \in G$, that there exists π^{-1} with $\pi \circ \pi^{-1} = id_X = \pi^{-1} \circ \pi$. We have seen that $\mathrm{Sym}(X)$ satisfies (G1), (G2) and (G3) and is thus a group. □

Definition 2.3. Let H be a subset of a group G. Then H is called a subgroup of G if it satisfies the following requirements.

(H1) H is not empty

(H2) for all $a, b \in H$ we have $a \cdot b \in H$

(H3) for all $a \in H$ we have $a^{-1} \in H$

Note that these requirements imply that $e \in H$. A subgroup of $\mathrm{Sym}(X)$ is, for instance, the set of all bijections $\pi \in G$ where x_0 is a fixed point for some $x_0 \in X$, that is : $\pi(x_0) = x_0$.

Definition 2.4. A bijection $f \colon X \to X$, where $f(x) = y$, $f(y) = x$ and $f(z) = z$ for some $x, y \in X$ and all $z \in X$, where $x \neq y$ and $x \neq z \neq y$ is denoted as $(x\ y)$.

This concludes the preknowledge of group theory. We will now introduce notions about the size of sets, equivalence relations and the Cartesian products of sets.

Definition 2.5. Sets A and B have the same cardinality, denoted $|A| = |B|$, if there exists a bijection from A to B.

With this definition, we have that $|\{a, b, c\}| = |\{a, d, e\}|$ if all elements are different, since there is a bijection $f: \{a, b, c\} \rightarrow \{a, d, e\}$ defined by $f(a) = a$, $f(b) = d$ and $f(c) = e$.

Note that this definition is more useful than counting element of a set since $\{a, b\}$ would then have size 2 and $\{a, a, a, b, b\}$ would have size 5, even though it is the same set.

Definition 2.6. The Cartesian product $X \times Y$ of sets X and Y is defined as:

$$X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\}$$

Definition 2.7. A relation $R \subseteq X \times X$ is an equivalence relation if it satisfies the following properties

- $(x, x) \in R$ for all $x \in X$.
- If $(x, y) \in R$, then $(y, x) \in R$.
- If $(x, y), (y, z) \in R$, then $(x, z) \in R$.

An example of an equivalence relation $R \subseteq X \times X$ is

$$R = \{(x, x) \mid x \in X\}.$$

2.2 Automata theory

Definition 2.8. An alphabet A is a finite set. The elements in A are referred to as letters.

Definition 2.9. If A is an alphabet, the set of words of finite length A^* is defined by:

λ, the empty word or word of length zero is an element of A^*.
If $w \in A^*$, then $aw \in A^*$ for all $a \in A$.

The set A^+ is equal to $A^* \setminus \{\lambda\}$

Definition 2.10. For any letter x in an alphabet A and any word $w \in A^*$, $|w|_x$ denotes the number of times the letter x occurs in A^*.

Definition 2.11. A DFA is a tuple $M = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite non-empty set of states,
- Σ is a finite non-empty alphabet,
- $\delta: Q \times \Sigma \rightarrow Q$ is a transition function,

- $q_0 \in Q$ is the initial state and
- $F \subseteq Q$ is the set of final states.

The function $\delta^*\colon Q \times \Sigma^* \to Q$ is used for defining the acceptance of DFA and is defined as:

- $\delta^*(q, \lambda) = q$ for all $q \in Q$.
- $\delta^*(q, xw) = \delta(\delta^*(q, x), w)$ for all $q \in Q$, $x \in \Sigma$ and $w \in \Sigma^*$.

The language accepted by DFA M is defined as the set

$$\mathcal{L}(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}.$$

To be able to represent configurations of an automaton, we define $\vdash_M$, which allows for compact notation.

Definition 2.12. The function $\vdash_M\colon Q \times \Sigma^* \to Q \times \Sigma^*$ is given by

$$[q_i, xw] \vdash_M [\delta(q_i, x), w]$$

for $x \in \Sigma, w \in \Sigma^*$ and δ the transition function of DFA M.
The notation $[q_j, w] \vdash_M^* [q_i, w']$ indicates that $[q_j, w] \vdash_M ... \vdash_M [q_i, w']$.

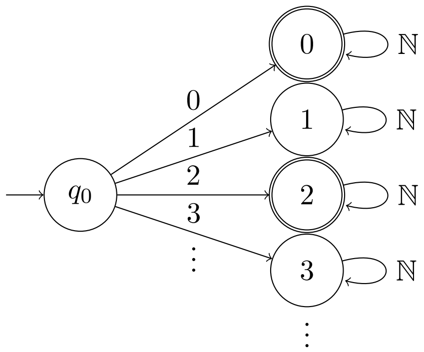
Chapter 3

The definition of G-automata

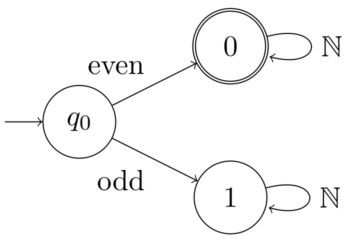
In this chapter, we will define G-automata that were also defined in [1]. First the notions of G-sets and equivariance are introduced, which are crucial for the definition of G-automata.

3.1 G-sets

Let us first look at how we can look at an infinite set in a finite way. Consider an automaton that accepts all the words over $\mathbb{N}^*$ where the first letter is even.

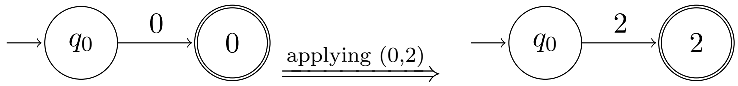


This automaton has an infinite amount of states and transitions, but we could represent this automaton as follows.



Although the set of states is infinite, we can interpret the automaton in a finite way.

We use group actions on sets to formalize the set of states and the alphabet of the G-automaton. A group action on an automaton can be for instance that we permute the states and letters in the automaton. If we were to have the following transition, we could permute the transition and state into the second figure, by taking the permutation $(0, 2)$.



Definition 3.1. A group action is a function $G \times X \to X$, where X is a set and G a group with the following requirements: for all $x \in X$ and for all $\pi, \sigma \in G$

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

The symbol e denotes the neutral element of G. Such a set X is called a G-set.

Definition 3.2. A data symmetry $(\mathbb{D}, G)$ is an infinite set of data $\mathbb{D}$ and a subgroup G of the group of all bijections on $\mathbb{D}$, $\mathrm{Sym}(\mathbb{D})$.

An example of a G-set is $\mathbb{D}$, with G equal to $\mathrm{Sym}(\mathbb{D})$. The group action $\pi \in G$ on $d_i \in \mathbb{D}$ is defined as $\pi \cdot d_i = \pi(d_i)$. Since π is a bijection on $\mathbb{D}$, $\pi(d_i)$ is again an element of $\mathbb{D}$, so we indeed have a function $G \times \mathbb{D} \to \mathbb{D}$. The neutral element of G is $id_{\mathbb{D}}$, since $id_{\mathbb{D}} \cdot d_i = id_{\mathbb{D}}(d_i) = d_i$ for all d_i in $\mathbb{D}$. The second requirement is also met since $\pi \cdot (\sigma \cdot d_i) = \pi(\sigma(d_i)) = (\pi \circ \sigma) \cdot d_i = (\pi \cdot \sigma) \cdot d_i$.

Example 3.3. We will now list some data symmetries:

- The equality symmetry. The set $\mathbb{D}$ is a countably infinite set, for instance the natural numbers and G is the group of all bijections on $\mathbb{D}$.
- The total order symmetry. The set of data values $\mathbb{D}$ is the set of rational numbers: $\mathbb{Q}$. G is the group of all monotone bijections, which means that for every bijection f, we have $q \leq p \implies f(q) \geq f(p)$ for all $q, p \in \mathbb{Q}$.

We saw earlier that $\mathbb{D}$ is a G-set, with the corresponding group $\mathrm{Sym}(\mathbb{D})$. This now also allows for the following G-sets with the group action defined as the point-wise action.

Suppose G has a group action on $\mathbb{D}$ and $\pi \in G$.

- $\mathbb{D}^n$ and $\mathbb{D}^\omega$. The set of n-tuples of data values and infinite sequences of data values. The point-wise action is then defined by: $\pi \cdot (d_1, d_2, \ldots, d_n) = (\pi \cdot d_1, \pi \cdot d_2, \ldots, \pi \cdot d_n)$ and $\pi \cdot (d_1, d_2, d_3, \ldots) = (\pi \cdot d_1, \pi \cdot d_2, \pi \cdot d_3, \ldots)$.

- $\mathbb{D}^*$. The set of all words of data values, where the point-wise action is $\pi \cdot (d_1 d_2 \dots d_n) = \pi \cdot d_1 \pi \cdot d_2 \dots \pi \cdot d_n$
- $\mathcal{P}(\mathbb{D})$. The power set of $\mathbb{D}$, where the element-wise action is $\pi \cdot \{d_1, d_2, d_3, \dots\} = \{\pi \cdot d_1, \pi \cdot d_2, \pi \cdot d_3, \dots\}$
- $\binom{\mathbb{D}}{n} = \{C \subseteq \mathbb{D} \mid |C| = n\}$. The set of subsets of $\mathbb{D}$ of size n, where the element-wise action is $\pi \cdot \{d_1, d_2, \dots, d_n\} = \{\pi \cdot d_1, \pi \cdot d_2, \dots, \pi \cdot d_n\}$.

Let us look at group actions from $\mathrm{Sym}(\mathbb{D})$ on elements in $\binom{\mathbb{D}}{n}$ and $\mathbb{D}^*$ in a more concrete example. Take the permutation that cycles only d_1, d_4 and d_6: $\pi \coloneqq (d_1\ d_4\ d_6) \in \mathrm{Sym}(\mathbb{D})$ and the set $\{d_1, d_2, d_3, d_5\} \subseteq \binom{\mathbb{D}}{n}$. Then

$$\begin{aligned}
&\pi \cdot \{d_1, d_2, d_3, d_5\} \\
&= \{\pi \cdot d_1, \pi \cdot d_2, \pi \cdot d_3, \pi \cdot d_5\} \\
&= \{\pi(d_1), \pi(d_2), \pi(d_3), \pi(d_5)\} \\
&= \{d_4, d_2, d_3, d_5\}.
\end{aligned}$$

Using the same permutation π and applying it to the element $d_4 d_6 d_1 \in \mathbb{D}^*$ gives

$$\begin{aligned}
&\pi \cdot d_4 d_6 d_1 \\
&= \pi \cdot d_4 \pi \cdot d_6 \pi \cdot d_1 \\
&= \pi(d_4)\pi(d_6)\pi(d_1) \\
&= d_6 d_1 d_4.
\end{aligned}$$

One might see that not all elements in, for instance, $\mathbb{D}^*$ can be permuted to all other elements even if they have the same length. Suppose we are in the equality symmetry and suppose we have the element $d_1 d_2 \in \mathbb{D}^*$, where $d_1 \neq d_2$. Then we cannot find $\pi \in G$ such that $\pi \cdot d_1 d_1 = d_1 d_2$ because π is a bijection. To distinguish elements that cannot be sent to each other when using the group action, we will now introduce the notion of orbits.

Definition 3.4. For $x \in X$ and X a G-set, the set $G \cdot x = \{g \cdot x \mid g \in G\}$ is called the orbit of x.

For the construction of the G-automaton, we will only look at orbit-finite sets, which means that there is a finite amount of different orbits. Orbits might have some elements in common and we can actually see that orbits need to always be exactly the same or disjoint.

Theorem 3.5. *Every G-set is a disjoint union of orbits*

Proof. Let X be a G-set and G the group action on X. Take $x_1, x_2 \in X$.

Suppose we have that $G \cdot x_1 \cap G \cdot x_2 \neq \emptyset$, that is, there exist $g_1, g_2 \in G$ with $g_1 x_1 = g_2 x_2$. This implies $g_2^{-1} g_1 x_1 = x_2$ and $x_1 = g_1^{-1} g_2 x_2$, which means that $G \cdot x_1 = G \cdot g_1^{-1} g_2 x_2 = G \cdot x_2$. We can now conclude that two orbits must be the same or disjoint. Since $x = e \cdot x \in G \cdot x$, we have that X is contained in the union of orbits. An orbit is by definition of the group action also contained in X. Thus X is equal to the union of orbits. We can now conclude that X is a disjoint union of orbits □

Example 3.6. The above outcome was also expected when looking at the earlier example of $\mathbb{D}^*$. In fact, we can now distinguish the orbits of $\{w \in \mathbb{D}^* \mid |w| = 2\}$ in the equality symmetry. The orbits are $\{d_1 d_2 \mid d_1 \neq d_2\}$ and $\{d_1 d_2 \mid d_1 = d_2\}$. And the orbits in the total order symmetry are: $\{d_1 d_2 \mid d_1 < d_2\}$, $\{d_1 d_2 \mid d_1 > d_2\}$ and $\{d_1 d_2 \mid d_1 = d_2\}$.

We can also look at orbits of more general cases, for instance $\mathbb{D}^n$ and $\mathcal{P}(\mathbb{D})$.

Example 3.7. Let us first look at $\mathcal{P}(\mathbb{D})$ in the equality symmetry. Since the group action acts as a bijection on $\mathbb{D}$, the number of elements will stay the same when applying the group action. Thus sets of different cardinalities are in different orbits. We can also see that sets of the same cardinality are in the same orbit. Take for instance $\{d_1, d_2, \ldots, d_n\}, \{c_1, c_2, \ldots, c_n\} \in \binom{\mathbb{D}}{n}$. If we remove elements that are in the intersection, we get $\{d_i, \ldots, d_j\}$ and $\{c_k, \ldots, c_l\}$. Note that these sets have the same cardinality, since they were the same cardinality and the number of elements removed was the same. If we now take the permutation $\pi := (d_i\ c_k) \cdot \cdots \cdot (d_j\ c_l)$, we get:

$$\pi \cdot \{d_1, d_2, \ldots, d_n\} = \{c_1, c_2, \ldots, c_n\}$$

This means that sets of the same cardinality are in the same orbit. Thus sets of the same cardinality are exactly the orbits of $\mathcal{P}(\mathbb{D})$. Since $\mathcal{P}(\mathbb{D})$ has an infinite amount of sets of different cardinalities, $\mathcal{P}(\mathbb{D})$ has an infinite amount of orbits.

The set $\mathbb{D}^n$ has a finite amount of orbits. To count the number of orbits, we will use the numbers that are used as a solution to a counting problem, named Bell's numbers.

Theorem 3.8. *The G-set $\mathbb{D}^n$ has B_n orbits in the equality symmetry, where B_n denote Bell's numbers.*

Proof. The orbits of $\mathbb{D}^n$ can be characterized by which indexes of the n-tuple contain the same elements. The orbit $\{(d_1, d_2, d_3) \mid d_1 = d_2, d_1 \neq d_3\} \subseteq \mathbb{D}^3$, for example, can be described by the following partition of the set $\{1, 2, 3\}$: $\{\{1, 2\}, \{3\}\}$. We can thus describe an orbit with the corresponding partition of indexes. Since we can make a bijection this way between the partitions of the set $\{1, 2, \ldots, n\}$ and the orbits of $\mathbb{D}^n$, $\mathbb{D}^n$ has exactly B_n orbits. □

3.2 Equivariance

In this section, we will introduce the notion of equivariance, which will be relevant for our transition function and the sets of initial and final states in the G-automaton.

Definition 3.9. A subset Y of a G-set X is called equivariant if $\pi \cdot Y = Y$ for all $\pi \in G$.

Note that this implies that Y is a union of orbits since:

$$Y = \bigcup_{\pi \in G} \pi \cdot Y = \bigcup_{\pi \in G} \bigcup_{x \in Y} \{\pi \cdot x\} = \bigcup_{x \in Y} G \cdot x.$$

The group action $\pi \cdot Y$ above, is the point-wise action illustrated after Example 3.3. An example of an equivariant subset of G-set X is X itself, which we also saw in Theorem 3.5.

We have so far obtained a way to use group actions on sets. We have also seen after Example 3.3 how a point-wise action can be used to create G-sets. This also holds for the product of two G-sets:
If we have a group G with a group action on X and Y, we can make a new group action on $X \times Y$ as follows.

$$\text{For } \pi \in G \text{ and } (x, y) \in X \times Y : \pi \cdot (x, y) = (\pi \cdot x, \pi \cdot y).$$

A relation $R \subseteq X \times Y$ can be used to describe functions, which means that this also allows us to make a definition about equivariant functions.

Definition 3.10. A function $f \colon X \to Y$, for G-sets X and Y, is equivariant if for all $x \in X$ and for all $\pi \in G$ we have $f(\pi \cdot x) = \pi \cdot f(x)$.

This definition corresponds to Definition 3.9 in the following way. Suppose we have an equivariant function f. We would now like that the relation given by our function f, $\{(x, f(x)) \in X \times Y\}$, would also be equivariant. But f is equivariant, so $\pi \cdot x$ is sent to $\pi \cdot f(x)$ for all $\pi \in G$. Thus $(\pi \cdot x, \pi \cdot f(x)) \in \{(x, f(x)) \in X \times Y\}$, which implies that the set $\{(x, f(x)) \in X \times Y\}$ is equivariant.

We will illustrate this definition with examples involving earlier introduced G-sets with their corresponding group action.

Example 3.11. The function $f : \mathbb{D}^3 \to \mathbb{D}^2$ defined by $(d_1, d_2, d_3) \mapsto (d_1, d_2)$ is equivariant. We can see this by writing out the definitions.

$$\begin{aligned}
&f(\pi \cdot (d_1, d_2, d_3)) \\
&= f((\pi \cdot d_1, \pi \cdot d_2, \pi \cdot d_3)) && \text{(Group action in } \mathbb{D}^3\text{.)} \\
&= (\pi \cdot d_1, \pi \cdot d_2) \\
&= \pi \cdot (d_1, d_2) && \text{(Group action in } \mathbb{D}^2\text{.)} \\
&= \pi \cdot f(d_1, d_2, d_3)
\end{aligned}$$

Thus $f(\pi \cdot (d_1, d_2, d_3)) = \pi \cdot f(d_1, d_2, d_3)$, which implies that f is equivariant.

Example 3.12. A function $g\colon \binom{\mathbb{D}}{2} \to \mathbb{D}^2$ cannot be equivariant.
Suppose g is equivariant and that for $d_1 \neq d_2$, we have $g(\{d_1, d_2\}) = (c_1, c_2)$. If $c_1 \notin \{d_1, d_2\}$, then we can take a permutation π that only swaps c_1 and c_3 and leaves the other data values: (c_1, c_3), where $d_1 \neq c_3 \neq d_2$. Then

$$\begin{aligned}
&g(\pi \cdot \{d_1, d_2\}) \\
&= g(\{\pi \cdot d_1, \pi \cdot d_2\}) \\
&= g(\{d_1, d_2\}) \\
&= (c_1, c_2) \\
&\neq (c_3, c_2) \\
&= (\pi \cdot c_1, \pi \cdot c_2) \\
&= \pi \cdot (c_1, c_2) \\
&= \pi \cdot g(\{d_1, d_2\})
\end{aligned}$$

This is a contradiction with the assumption the g is equivariant. We can thus assume the function g should be of the form $\{d, e\} \mapsto (d, e)$, but now we have the following for a permutation σ that does not swap d_1 and d_2.

$$\begin{aligned}
g(\sigma \cdot \{d_1, d_2\}) &= g(\{\sigma \cdot d_1, \sigma \cdot d_2\}) = g(\{\sigma \cdot d_2, \sigma \cdot d_1\}) = (\sigma \cdot d_2, \sigma \cdot d_1) \\
&\neq (\sigma \cdot d_1, \sigma \cdot d_2) = \sigma \cdot (d_1, d_2) = \sigma \cdot g(\{d_1, d_2\})
\end{aligned}$$

So a function $g : \binom{\mathbb{D}}{2} \to \mathbb{D}^2$ cannot be equivariant.

Theorem 3.13. *If $f\colon X \to Y$ and $g\colon Y \to Z$ are equivariant. Then $g \circ f\colon X \to Z$ is equivariant.*

Proof. Suppose $f\colon X \to Y$ and $g\colon Y \to Z$ are equivariant. Then for all $x \in X$ and for all $\pi \in G$ we get $(g \circ f)(\pi \cdot x) = g(f(\pi \cdot x)) = g(\pi \cdot f(x)) = \pi \cdot g(f(x)) = \pi \cdot (g \circ f)(x)$. $\square$

3.3 G-automata

In this section we will define G-automata, but before we do this, we extend the concept of a language to G-sets. First, the definition of an alphabet is extended as follows.

Definition 3.14. An alphabet A is an orbit finite G-set. A G-language is an equivariant set $L \subseteq A^*$, where the group action on A^* is the point-wise action that was defined after Example 3.3

Definition 3.15. A nondeterministic G-automaton, denoted as G-NFA, consists of a tuple (Q, A, δ, I, F), where

- Q is an orbit finite G-set of states.

- A is an orbit finite G-set called the alphabet.
- $\delta \subseteq Q \times A \times Q$ is an equivariant transition relation.
- $I \subseteq Q$ is an equivariant subset of initial states.
- $F \subseteq Q$ is an equivariant subset of final states.

The relation $\delta^* \subseteq Q \times A^* \times Q$ is used for acceptance and is defined inductively in the following way.

$(q, \lambda, q) \in \delta^*$ for all $q \in Q$
$(q, xw, p) \in \delta^*$ if $(q', w, p) \in \delta^*$ and $(q, x, q') \in \delta$, for $q, p, q' \in Q, w \in A^*$ and $x \in A$.

We define acceptance in the same way as regular automata. We say a word $w \in A^*$ is accepted by the automaton (Q, A, δ, I, F) if $(q_i, w, q_f) \in \delta^*$ where $q_i \in I$ and $q_f \in F$.

We wish to prove that the language accepted by a G-automaton is a G-language, but to prove this, we must first prove that the relation δ^* is equivariant. We will do this after the following lemmas.

Lemma 3.16. Given G-set X with the group action G. If Y is a subset of X and for all $\pi \in G$, $\pi \cdot Y \subseteq Y$, then Y is equivariant.

Proof. Since $\pi \cdot Y \subseteq Y$ for all $\pi \in G$, we also have $\pi^{-1} \cdot Y \subseteq Y$. By applying π to both sides, we obtain $Y \subseteq \pi \cdot Y$. Thus for all $\pi \in G$, $\pi \cdot Y \subseteq Y$ and $Y \subseteq \pi \cdot Y$, which means exactly that $Y = \pi \cdot Y$ and that Y is equivariant. □

Lemma 3.17. If $\delta \subseteq Q \times A \times Q$ is the equivariant transition relation of a nondeterministic G-automaton (Q, A, δ, I, F) and $\{(q, w, p) \in \delta^* \mid |w| \leq n\}$ is equivariant, then $\{(q, w, p) \in \delta^* \mid |w| \leq n + 1\}$ is equivariant.

Proof. We first prove that $(q, xw, p) \in \delta^*$ implies $\pi \cdot (q, xw, p) \in \delta^*$ for all $x \in A$, $w \in A^n$ and $\pi \in G$.
So let us assume that $(q, xw, p) \in \delta^*$ for some $xw \in A^{n+1}$. By definition of δ^*, this implies that $(q', w, p) \in \delta^*$ and $(q, x, q') \in \delta$ for some $q' \in Q$. Because $\{(q, w, p) \in \delta^* \mid |w| \leq n\}$ is equivariant by assumption, we can conclude that also $\pi \cdot (q', w, p) = (\pi \cdot q', \pi \cdot w, \pi \cdot p) \in \delta^*$ and $\pi \cdot (q, x, q') = (\pi \cdot q, \pi \cdot x, \pi \cdot q') \in \delta$. Then $\pi \cdot (q, xw, p) = (\pi \cdot q, \pi \cdot (xw), \pi \cdot p) \in \delta^*$, by definition of δ^*.

Thus we now have that $(q, xw, p) \in \delta^*$ implies $\pi \cdot (q, xw, p) \in \delta^*$. Using the assumption that $\{(q, w, p) \in \delta^* \mid |w| \leq n\}$ is equivariant, we have $\pi \cdot \{(q, w, p) \in \delta^* \mid |w| \leq n + 1\} \subseteq \{(q, w, p) \in \delta^* \mid |w| \leq n + 1\}$ for all $\pi \in G$. Use Lemma 3.16 to conclude that $\{(q, w, p) \in \delta^* \mid |w| \leq n + 1\}$ is equivariant. □

Corollary 3.18. If $\delta \subseteq Q \times A \times Q$ is the equivariant transition relation of a nondeterministic G-automaton (Q, A, δ, I, F), then δ^* is equivariant.

Proof. This follows directly from the condition that δ is equivariant and by induction on Lemma 3.17 $\square$

Since we required I, F to be equivariant, we can also show that the set of words accepted by the G-automaton is also equivariant.

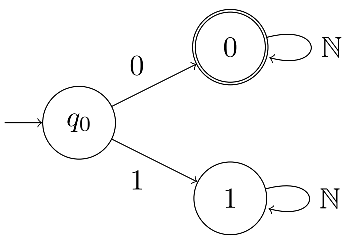
Theorem 3.19. *The language accepted by a G-automaton is a G-language.*

Proof. Suppose w is a word in the language $\mathcal{L}$ that is accepted by the G-automaton, then we have $(q_i, w, q_f) \in \delta^*$ for some $q_i \in I$ and $q_f \in F$. If we apply $\pi \in G$ to this tuple, we obtain $\pi \cdot (q_i, w, q_f) = (\pi \cdot q_i, \pi \cdot w, \pi \cdot q_f)$. Since I and F are equivariant sets, we have that $\pi \cdot q_i \in I$ and $\pi \cdot q_f \in F$ and since δ^* is equivariant, $(\pi \cdot q_i, \pi \cdot w, \pi \cdot q_f) \in \delta^*$. Then by definition of acceptance, $\pi \cdot w$ is also in the language. We can now conclude that $\pi \cdot \mathcal{L} \subseteq \mathcal{L}$. Because we took arbitrary $\pi \in G$, by Lemma 3.16, $\mathcal{L} \subseteq A^*$ is equivariant. Thus the language accepted by a G-automaton is a G-language. $\square$

Definition 3.20. A deterministic G-automaton, also called G-DFA, is a non-deterministic G-automaton, where the definition of δ and I is changed to:

- $\delta : Q \times A \to Q$ is an equivariant transition function.
- the set of initial states is the equivariant singleton $\{q_0\}$.

Example 3.21. Let us return to the example automaton we introduced at the beginning of this chapter. Let our group be the subgroup of $\mathrm{Sym}(\mathbb{N})$, where for all bijections, even numbers are sent to even numbers. A G-DFA $(Q, \mathbb{N}, \delta, \{q_0\}, \{n \in \mathbb{N} \mid n \text{ is even }\})$ that accepts the language $\mathcal{L} = \{nw \in \mathbb{N}^+ \mid n \in \mathbb{N} \text{ where } n \text{ is even}\}$ is then given by:



Note that we can represent our automaton as above. If $\delta(q_0, 0) = 0$, then $\delta(q_0, n) = n$ for all even n, by equivariance of δ. And since 0 is a final state, n is also a final state, for even n, because F is equivariant.

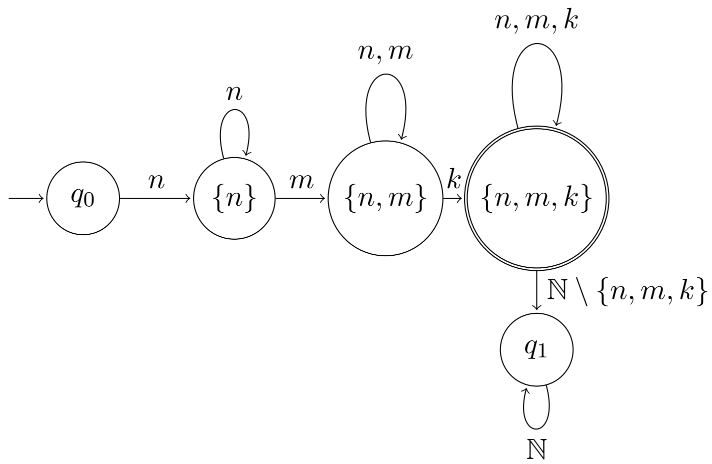
Example 3.22. We now give a G-DFA $(Q, \mathbb{N}, \delta, \{q_0\}, F)$ that recognises the language

$$\mathcal{L} = \{w \in \mathbb{N}^* \mid \text{there exist different } i, j, k \in \mathbb{N} \text{ where } |w|_i > 0, |w|_j > 0, |w|_k > 0 \\ \text{and for all } n \in \mathbb{N} \text{ different from } i, j, k : |w|_n = 0\}$$

in the equality symmetry. Where the set of states Q is:

$$\begin{aligned} Q = &\{q_0, q_1\} \\ &\cup \{\{n\} \mid n \in \mathbb{N}\} \\ &\cup \{\{n, m\} \mid n \in \mathbb{N}, m \in \mathbb{N}\} \\ &\cup \{\{n, m, k\} \mid n \in \mathbb{N}, m \in \mathbb{N}, k \in \mathbb{N}\}, \end{aligned}$$

,the set of final states is: $F = \{\{n, m, k\} \mid n \in \mathbb{N}, m \in \mathbb{N}\backslash\{n\}, k \in \mathbb{N}\backslash\{n, m\}\}$ and the group action on q_0, q_1 is: $\pi \cdot q_0 = q_0$ and $\pi \cdot q_1 = q_1$. The group action on the other states, is the element-wise action. We make sure that we have read three different letters before we reached a final state. Note that n, m, k can really be any number in $\mathbb{N}$ as long as they are different. We could thus have chosen 1,2 and 3 to represent n, m and k in the G-DFA below.



Chapter 4

2DFA

In this chapter, we define 2DFA and show that they are expressively equivalent to DFA using a construction as in [6]. We also prove the Myhill-Nerode theorem to give the reader some intuition for this construction.

4.1 Defining 2DFA

A regular DFA can only go through the input word in one direction, namely from left to right. But there is also the concept of a 2DFA where every transition can be to the left or right. We will first introduce this automaton and then after, show that it accepts the same languages as a regular type of DFA.

Definition 4.1. A 2DFA is a tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states,
- Σ is a finite alphabet,
- q_0 is the initial state,
- $F \subseteq Q$ is a set of final states,
- and $\delta : Q \times \Sigma \to Q \times \{L, R\}$ is a transition function that can also state if we move to the left after it has read a letter in the alphabet

The symbol R refers to a move to the right and L to a move to the left.

We will represent the current configuration of a 2DFA as an element of $\Sigma^* Q \Sigma^*$, where wqw' means that the input was ww' and we are currently in state q reading the first letter of w'. We use this notation to be able to represent moves of a 2DFA. We will write $wqxw' \vdash wxpw'$ to indicate that the second configuration follows from the first.

Assume $w, w' \in \Sigma^*$, $q \in Q$ and $x, y \in \Sigma$.

wqw' is a final configuration if $w' = \lambda$.
$wqxw'$ is a final configuration if $w = \lambda$ and $\delta(q, x) = (p, L)$.
$wqxw' \vdash wxpw'$ if $\delta(q, x) = (p, R)$.
$wyqxw' \vdash wpyxw'$ if $\delta(q, x) = (p, L)$.

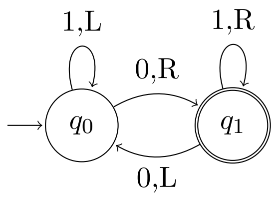
The first final configuration is because we are not reading a letter and the second final configuration is because we fall off the left side of the tape. To simplify the definition of acceptance, we will introduce some notation. We will write $c_1 \vdash^* c_n$ for configurations c_1 and c_n if we have $c_1 \vdash c_2 \vdash \cdots \vdash c_n$.

Definition 4.2. A word w is accepted by the 2DFA $(Q, \Sigma, \delta, q_0, F)$ if we have $q_0 w \vdash^* w q_f$ for some $q_f \in F$. So the language that the 2DFA accepts is defined by the set $\mathcal{L} = \{w \in \Sigma^* \mid q_0 w \vdash^* w q_f\}$.

To show how a 2DFA works, we will give an example. The example 2DFA is formally given by $M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$ where δ is defined by: $\delta(q_0, 0) = (q_1, R)$, $\delta(q_0, 1) = (q_0, L)$, $\delta(q_1, 1) = (q_1, R)$ and $\delta(q_1, 0) = (q_0, L)$. Note that this 2DFA shows the three behaviours. It will fall off the tape for the word 1, not terminate for the word 00 and accept the word 01:

$$q_0 1$$
$$q_0 00 \vdash 0 q_1 0 \vdash q_0 00 \vdash 0 q_1 0 \ldots$$
$$q_0 01 \vdash 0 q_1 1 \vdash 01 q_1.$$

The language this 2DFA accepts is $\mathcal{L} = \{01^n \mid n \in \mathbb{N}\}$.



4.2 Myhill-Nerode

We can simulate an arbitrary DFA by a 2DFA by converting all transitions to right transitions, which means that languages that DFAs accept can also be accepted by 2DFAs. The converse is also true: Given a 2DFA, we can construct a DFA that accepts the same language. To get some intuition before we begin with the construction, we will first look at how we can partition words with respect to an DFA, using a concept from Myhill-Nerode.

Take a language L over Σ. Then the Myhill-Nerode equivalence relation $x =_L y$ if defined by $xz \in L$ if and only if $yz \in L$ for all $z \in \Sigma^*$. Note that this is indeed an equivalence relation:

$$xz \in L \text{ if and only if } xz \in L \tag{4.1}$$

$$(xz \in L \text{ if and only if } yz \in L) \implies (yz \in L \text{ if and only if } xz \in L) \tag{4.2}$$

$$\begin{aligned}&(xz \in L \text{ if and only if } yz \in L) \text{ and } (yz \in L \text{ if and only if } az \in L)\\ &\quad\implies xz \in L \text{ if and only if } az \in L\end{aligned} \tag{4.3}$$

Theorem 4.3. *(Myhill-Nerode) A language L over Σ can be accepted by a DFA if and only if $=_L$ has finitely many equivalence classes.*

Proof. " $\Longleftarrow$ "
We need to show that L can be accepted by a DFA. To achieve this, we create the following DFA $M = (Q, \Sigma, \delta, q_0, F)$.

- $Q = \{[w]_{=_L} \mid w \in \Sigma^*\}$
- δ is defined by $\delta([w]_{=_L}, x) = [wx]_{=_L}$
- $q_0 = [\lambda]_{=_L}$
- $F = \{[w]_{=_L} \mid w \in L\}$

The things left to prove are now that δ is well-defined and that M accepts exactly L. Let us first prove that δ is well-defined. So we have to show that $w =_L w'$ implies $wx =_L w'x$ for all $x \in \Sigma$. Suppose $w =_L w'$, then $wv \in L \iff w'v \in L$ for all $v \in \Sigma^*$. Now take $x \in \Sigma$. Then $(wx)v \in L \iff (w'x)v \in L$, since $w(xv) \in L \iff w'(xv) \in L$. Thus we can conclude that $w =_L w'$ implies $wx =_L w'x$ for all $x \in \Sigma$, which implies that δ is well-defined.
We now prove that M accepts exactly L. Note that $[[\lambda]_{=_L}, w] \vdash_M^* [w]_{=_L}$. Thus if $w \in L$, then $[w]_{=_L} \in F$, which implies that w is accepted by M. And because for all $w' \in [w]_{=_L}$, we have $wv \in L \iff w'v \in L$ for all $v \in \Sigma^*$, we can take $v = \lambda$, which gives us: $w\lambda \in L \iff w'\lambda \in L$. This means that $w' \in L$ and that M accepts precisely L.

" $\Longrightarrow$ "
Suppose L is accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$. We can now make an equivalence relation $=_M$ defined as $w =_M w'$ if w and w' halt in the same state of M. We first note that L is the union of all words that halt in a final state and that L is thus a finite union of some equivalence classes.

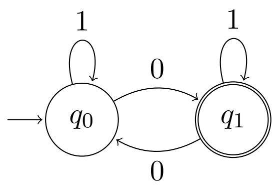
Now suppose $w =_M w'$. Then $[q_0, w] \vdash_M^* [q_i, \lambda]$ and $[q_0, w'] \vdash_M^* [q_i, \lambda]$ for some $q_i \in Q$. This also implies $[q_0, wv] \vdash_M^* [q_i, v] \vdash_M^* [q_j, \lambda]$ and $[q_0, w'v] \vdash^* [q_i, v] \vdash_M^* [q_j, \lambda]$ for all $v \in \Sigma^*$ and some $q_j \in Q$ dependant on v. Thus if $w =_M w'$ then also $wv =_M w'v$ for all $v \in \Sigma^*$. Since L was a union of these equivalence classes, we also have that $wv \in L \iff w'v \in L$ for all $v \in \Sigma^*$

because wv halts in the same state as $w'v$. Thus our w and w' are also equivalent with respect to $=_L$. We can now conclude that $[w]_{=_M} \subseteq [w]_{=_L}$.

So the results obtained so far are that L is a finite union of some equivalence classes and that that $[w]_{=_M} \subseteq [w]_{=_L}$ for all $w \in \Sigma^*$. We can now conclude that every equivalence class $[w]_{=_L}$ is not empty, since $[w]_{=_M} \subseteq [w]_{=_L}$ for all $w \in \Sigma^*$. But because there were a finite amount of equivalence classes with respect to $=_M$, there must also be a finite amount of equivalence classes $=_L$, again since $[w]_{=_M} \subseteq [w]_{=_L}$. Thus we can end our proof by concluding that $=_L$ has finitely many equivalence classes. $\square$

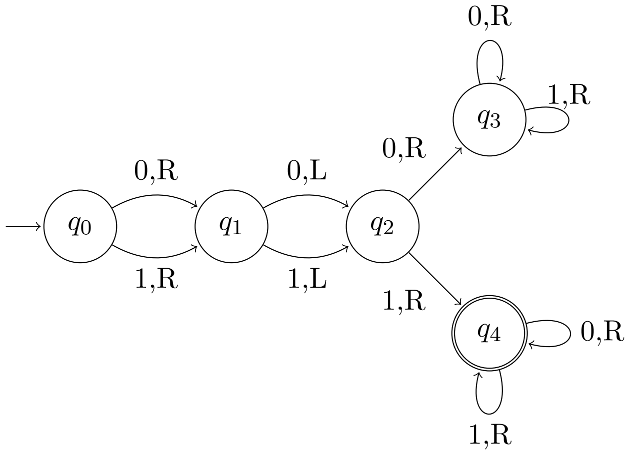
Before we look at how this theorem can help with our construction, let us first look at a step from the theorem in an example. Consider the following DFA. If we end up at the same state, when computing two different words, for instance the words 101 and 00111 in the automaton below. Then this implies that they are also in the same equivalence class, since:

$$[q_0, 101w] \vdash_M^* [q_1, w] \vdash_M^* [q_1, \lambda] \iff [q_0, 00111w] \vdash_M^* [q_1, w] \vdash_M^* [q_1, \lambda].$$



The proof of the Myhill-Nerode theorem uses the property that $[w] = [w']$ implies $[wx] = [w'x]$ to use the equivalence classes as states. We wish to achieve a similar concept where we would have a finite amount of equivalence classes that satisfy $[w] = [w'] \implies [wx] = [w'x]$, because this would allow for a simple construction of the DFA that will accept the same language as a given 2DFA.

For moves to the right (x, R), we already have that $[w] = [w'] \implies [wx] = [w'x]$. We might encounter a problem with a left transition however, because we may end up in a different state where the words from the same equivalence class show different behaviour. Consider the automaton below.



Although we would link q_1 with the equivalence class $\{0, 1\}$ when using the relation $=_M$, we see that it is not true that $0 =_L 1$ because 0 and 1 have different behaviour later on in the automaton: we are able to move to the left and look at the 0 and 1 again, as seen in state q_2. This means that the earlier implication $w =_M w' \implies wx =_M w'x$ does not hold anymore for $x \in \Sigma$. It would be useful for the upcoming construction to come up with an equivalence relation, for which $w =_M w' \implies wx =_M w'x$ holds again. To achieve this, consider the following.

4.3 2DFA equivalence

We construct, given a 2DFA $(Q, \Sigma, \delta, q_0, F)$, functions $\tau_w : Q \cup \{\bar{q}\} \to Q \cup \{0\}$, where $\bar{q}$ and 0 are fresh symbols. This function tells us a lot about the behaviour of the given 2DFA. The function tells us if we ever move one letter to the right and if so, in which state we end up. The output of $\bar{q}$ tells us if we will have ever moved through the word and if so, in which state we end up.

For all $w \in \Sigma^*$ and $x \in \Sigma$, τ_{wx} is defined by:

$$\tau_{wx}(q) = \begin{cases} p & \text{if } wqx \vdash^* wxp \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_{wx}(\bar{q}) = \begin{cases} p & \text{if } q_0wx \vdash^* wxp \\ 0 & \text{otherwise} \end{cases}$$

Note that we can determine if $wqx \vdash^* wxp$, since we can determine if we have fallen of the tape or have entered a loop.

If we consider w and w' to be equivalent if $\tau_w = \tau_{w'}$, we have an equivalence relation over words that end up in the same state when reading the first letter from the right. We now wish to prove that $\tau_w = \tau_{w'}$ implies $\tau_{wx} = \tau_{w'x}$ for $x \in \Sigma$, to prove that a transition function is well-defined if we use τ_w as input. So let us now prove the implication.

Lemma 4.4. For 2DFA $(Q, \Sigma, \delta, q_0, F)$, $w \in \Sigma^+$ and $x \in \Sigma$, $\tau_w = \tau_{w'}$ implies $\tau_{wx} = \tau_{w'x}$

Proof. We will first prove that $\tau_{wx}(q) = \tau_{w'x}(q)$ for $q \in Q$ if $\tau_w = \tau_{w'}$ and secondly that $\tau_{wx}(\bar{q}) = \tau_{w'x}(\bar{q})$ if $\tau_w = \tau_{w'}$.

Suppose $\tau_w(q) = \tau_{w'}(q)$ for all $q \in Q$. If state q contains a transition (x, R), we have $wqx \vdash wxp$ and $w'qx \vdash w'xp$ which imply $\tau_{wx}(q) = p = \tau_{w'x}(q)$. If state q contains a transition (x, L), we have that since $\tau_w = \tau_{w'}$, the values $\tau_{wx}(q)$ and $\tau_{w'x}(q)$ will be the same.

Now also suppose $\tau_w(\bar{q}) = \tau_{w'}(\bar{q})$. We can now use the just proven statement $\forall q \in Q[\tau_w(q) = \tau_{w'}(q)] \implies \forall q \in Q[\tau_{wx}(q) = \tau_{w'x}(q)]$ to prove $\tau_{wx}(\bar{q}) = \tau_{w'x}(\bar{q})$. If $\tau_w(\bar{q}) = 0$, then $\tau_{wx}(\bar{q})$ is also 0, since we will never even look at x. If $\tau_w(\bar{q}) = p$, then $\tau_{wx}(\bar{q}) = \tau_{wx}(p) = \tau_{w'x}(p) = \tau_{w'x}(\bar{q})$. We can now conclude the desired $\tau_w = \tau_{w'} \implies \tau_{wx} = \tau_{w'x}$. □

The result of this Lemma shows that we can indeed use this equivalence relation in a construction similar to that of Myhill-Nerode theorem. We will now give the construction and prove that 2DFA and DFA accept the same languages.

Theorem 4.5. *2DFA's and DFA's accept the same languages.*

Proof. As said before is the conversion from DFA to 2DFA quite simple because we can transform the transitions into right transitions. This would formally look as follows.

Take an arbitrary DFA $M = (Q, \Sigma, \delta, q_0, F)$. Now make 2DFA $M' = (Q, \Sigma, \delta', q_0, F)$, where $\delta'(q, x) = (\delta(q, x), R)$. Because a transition in a DFA is always a transition to the right, M' accepts the same language as M.

We now prove the other inclusion. Take an arbitrary 2DFA $M = (Q, \Sigma, \delta, q_0, F)$ We use the functions τ_w, that were introduced above, as states with the same idea as the Myhill-Nerode theorem.

We now construct our DFA $M' = (Q', \Sigma, \delta', q_0', F')$, where

- $Q' = \{q_0'\} \cup \{\tau_w \mid w \in \Sigma^+\}$
- $\delta'(q_0', x) = \tau_x$ and $\delta'(\tau_w, x) = \tau_{wx}$, with $x \in \Sigma, w \in \Sigma^+$
- q_0' is a new starting state that did not occur in Q and
- $F' = \{\tau_w \mid \tau_w(\bar{q}) = q_f$ for some $q_f \in F\}$.

Note that because τ_w are functions from $Q \cup \{\bar{q}\}$ to $Q \cup \{0\}$, we have $|Q'| \leq 1 + (|Q|+1)^{|Q|+1}$. So the set of states is finite even though the set of words is infinite. Also note that the transition function is well defined on these states by Lemma 4.4. We can see that by induction we have $\delta'^*(q_0', w) = \tau_w$. Thus we have now that:

$$
\begin{aligned}
& w \text{ is accepted by } M \\
\iff & q_0 w \vdash^* w q_f \text{ for some } q_f \in F && \text{(By the definition of acceptance.)} \\
\iff & \tau_w(\bar{q}) = q_f \text{ for some } q_f \in F && \text{(By the construction of } \tau_w.\text{)} \\
\iff & \tau_w \in F' && \text{(By construction of } F'.\text{)} \\
\iff & \delta'^*(q_0', w) \in F' && \text{(Because } \delta'^*(q_0', w) = \tau_w.\text{)} \\
\iff & w \text{ is accepted by } M' && \text{(By the definition of acceptance.)}
\end{aligned}
$$

We have now obtained a DFA that accepts exactly the same language as the given 2DFA, thus we can conclude that a DFA's and 2DFA's accept the same languages. □

Example 4.6. We now show the construction on the example 2DFA after Definition 4.2. There are 3^3 functions from $\{q_0, q_1, \bar{q}\}$ to $\{q_0, q_1, 0\}$, but we won't be using every function because we remark the following. We cannot have $\tau_w(q_0) = 0 = \tau_w(q_1)$ for some $w \in \Sigma^+$ because we move to the right in q_0 if we are reading a 0 and we move to the right in q_1 if we are reading a 1. We can also not have the state q_0 as output since it does not have any incoming right transitions. This all leaves us with the following possible functions.

	$f(q_0)$	$f(q_1)$	$f(\bar{q})$
1	0	q_1	0
2	0	q_1	q_1
3	q_1	0	0
4	q_1	0	q_1
5	q_1	q_1	0
6	q_1	q_1	q_1

In our construction we only use the above functions if it is equal to τ_w for some $w \in \Sigma^+$. We now look at which words correspond to the above functions.

1. $\tau_w(q_0) = 0$ indicates that the 2DFA with input w either does not terminate or walk of the left side of the tape. We can only walk off the tape if we have only 1's. This criterion also corresponds to the other values, so $w \in 1^+$

2. $\tau_w(q_0) = 0$ indicates again that w contains only 1's. Now $\tau_w(\bar{q}) = q_1$ indicates that the words is accepted by the 2DFA. This is however not possible since the word contains only 1's and we have seen that the language accepted by the 2DFA is $\{01^n \mid n \in \mathbb{N}\}$. So this function is not part of our construction.

3. $\tau_w(q_0) = q_1$ and $\tau_w(q_1) = 0$ indicate that w ends on a 0. $\tau_w(\bar{q}) = 0$ indicates that the word is not accepted, which must mean that there are two or more 0's in w or that there are one or more 1's. Thus we must have $w \in \{0,1\}^+0$.

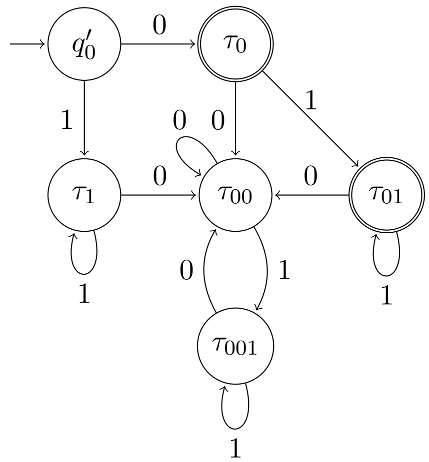
4. We have again that w must end on a 0. The word is accepted in q_1, so the only option is that $w = 0$.

5. $\tau_w(q_0) = q_1$ indicates that the word contains a 0 and $\tau_w(q_1) = q_1$ indicates that it ends on a 1. The word is not accepted, so there again must be another 0 or 1 left the the 0. This all means that $w \in \{0,1\}^+01^+$.

6. We again have that the word ends on a 1 and contains a 0. Since the word is accepted, we must have $w \in 01^+$.

Note that we have indeed obtained a partition of Σ^*.

We can now easily make the state diagram of the resulting DFA that accepts the same language as the 2DFA after Definition 4.2. We can also see that the automaton below is not minimal, so our construction does not necessarily give a minimal automaton.



Chapter 5

G-2DFA

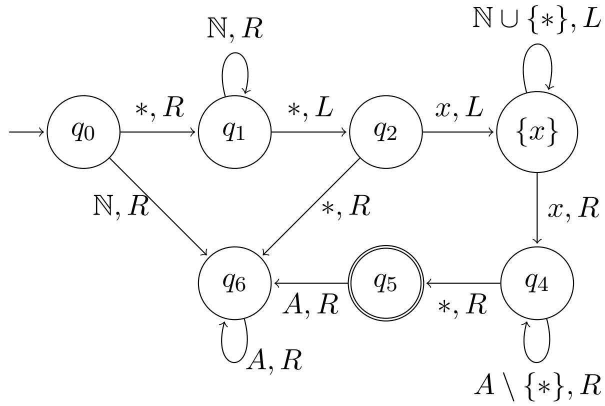
In this chapter, we combine the notions of 2DFA and G-DFA to define a G-2DFA. We explore if these G-2DFAs accept the same languages as G-DFAs using the functions τ_w that we introduced in Section 4.3.

Definition 5.1. A G-2DFA is a tuple $(Q, A, \delta, \{q_0\}, F)$, where

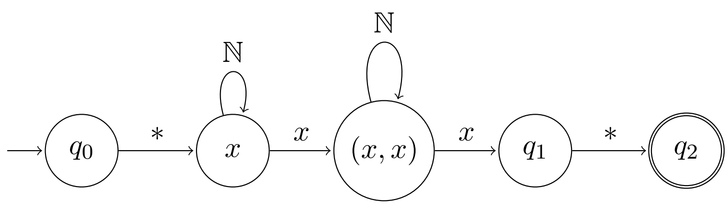
- Q is an orbit finite G-set of states.
- A is an orbit finite G-set called the alphabet.
- $\delta : Q \times A \to Q \times \{L, R\}$ is an equivariant transition relation, where the group action on $\{R, L\}$ is the identity.
- $\{q_0\} \subseteq Q$ is an equivariant subset of initial states.
- $F \subseteq Q$ is an equivariant subset of final states.

Acceptance is defined the same as in Definition 4.2. A word w is accepted by the G-2DFA $(Q, \Sigma, \delta, \{q_0\}, F)$ if we have $q_0 w \vdash^* w q_f$ for some $q_f \in F$.

Example 5.2. Let us now look at an example G-2DFA $(Q, \mathbb{N} \cup \{*\}, \delta, \{q_0\}, \{q_5\})$ that accepts the language $\mathcal{L} = \{*wx* \mid w \in \mathbb{N}^*, x \in \mathbb{N} \text{ and } |w|_x > 0\}$. The set of states in the automaton below is $Q = \{q_0, q_1, q_2, q_4, q_5, q_6\} \cup \{\{x\} \mid x \in \mathbb{N}\}$ and $G = \mathrm{Sym}(\mathbb{N})$.



If we wanted to make a G-DFA that accepts the same language as above, we could try to remember all the letters we have encountered as sets in our state. This would mean that the word $*123432$ ends up in state $\{1, 2, 3, 4\}$. Since words can have all different letters, for all lengths, we will not have an orbit finite amount of states, by an argument very similar to Example 3.7. This means that this specific construction will not work. The language above cannot be accepted by a G-DFA by a version of the Myhill-Nerode theorem for G-DFA's. We could, however, construct the following G-NFA that accepts $\mathcal{L}$.



The set of states in the G-NFA above is $Q = \{q_0, q_1, q_2\} \cup \{x \in \mathbb{N}\} \cup \{(x, x) \mid x \in \mathbb{N}\}$, where the group action is $\pi \cdot q_i = q_i$ for $0 \leq i \leq 2$ and the point-wise action of $\mathrm{Sym}(\mathbb{N})$ for the other states.

The first state already assumes the number that will occur multiple times in the word. Suppose this number is n. Then state q_1 will only be reached if the letter n has occurred twice already and if the last letter was an n.

We extend the Myhill-Nerode equivalence relation from Theorem 4.3 to infinite alphabets. In this thesis we will not give a proof. The proof can be found in [1].

Theorem 5.3. Myhill-Nerode theorem for G-sets. *Assume A is an orbit finite G-set and $L \subseteq A^*$ a G-language, then the following statements are equivalent.*

- $\{[w]_{=_L} \mid w \in A^*\}$ *is orbit finite.*

- *L is recognised by a G-DFA.*

Let us return to the G-2DFA and give an infinite amount of orbits with respect to the Myhill-Nerode equivalence relation.

The set $\{*1, *12, *123, *1234, *12345, \dots\}$ contains elements with disjoint orbits with respect to the Myhill-Nerode equivalence relation. To see this, take different x and y with $x, y \in \{*1, *12, *123, *1234, *12345, \dots\}, \pi, \pi' \in \mathrm{Sym}(X)$ and consider $\pi \cdot x$ and $\pi' \cdot y$. Without loss of generality assume that $|\pi' \cdot y| > |\pi \cdot x|$. Then $\pi' \cdot y$ contains a letter that is not in $\pi \cdot x$. Call this letter k. This means that $(\pi \cdot x)k* \notin \mathcal{L} = \{*wx* \mid w \in \mathbb{N}^*, x \in \mathbb{N} \text{ and } |w|_x > 0\}$ and $(\pi' \cdot y)k* \in \mathcal{L}$.

Because $\{*1, *12, *123, *1234, *12345, \dots\}$ has an infinite amount of elements in different orbits, $\{*wx* \mid w \in \mathbb{N}^*, x \in \mathbb{N} \text{ and } |w|_x > 0\}$ cannot be accepted by a G-DFA. We can thus conclude that G-DFAs and G-NFAs do not accept the same languages and that G-NFAs can accept more languages that G-DFAs.

5.1 From G-2DFA to G-DFA

Let us now return to the construction used for DFA and 2DFA equivalence and look at when the construction works and why it fails for G-DFA, when using the following group action.

$$\pi \cdot \tau_w = \tau_{\pi \cdot w}$$

To use this group action, we have to prove that it is well defined, thus that if $\tau_w = \tau_{w'}$ for $w, w' \in A^+$, we have for all $\pi \in G$ that $\pi \cdot \tau_w = \pi \cdot \tau_{w'}$.

To show this, we will show that the group action determines the function as follows:

$$\pi \cdot \tau_w = x \mapsto \pi(\tau_w(\pi^{-1} \cdot x)).$$

Lemma 5.4. Given G-2DFA $(Q, A, \delta, \{q_0\}, F)$, if $\tau_{wx}(q) = p$ for $w \in A^*, x \in A$ and $p, q \in Q$, then $\pi \cdot \tau_{wx}(\pi \cdot q) = \pi \cdot p$.

Proof. Suppose $\tau_{wx}(q) = p$ for $w \in A^*, x \in A$ and $p, q \in Q$, then this means by construction of τ_{wx} that $wqx \vdash^* wxp$. Thus there are a number of transitions taken to get from wqx to wxp, where each transition is of the form $\delta(p_i, y) = (p_j, M)$ with y a letter in wx and $M \in \{L, R\}$. Then by equivariance of δ, we also have

$$\begin{aligned}
&\delta(\pi \cdot p_i, \pi \cdot y) \\
&= \pi \cdot \delta(p_i, y) && \text{(Equivariance of } \delta.) \\
&= \pi \cdot (p_j, M) && \text{(Assumed value.)} \\
&= (\pi \cdot p_j, M). && \text{(Group action on tuple.)}
\end{aligned}$$

If we use induction on the number of transitions, this implies that $\pi \cdot \tau_{wx}(\pi \cdot q) = \pi \cdot p$ $\square$

Note that this corresponds to the function $x \mapsto \pi \cdot (\tau_{wx}(\pi^{-1} \cdot x))$ before the Lemma, since $\pi \cdot q \mapsto \pi \cdot (\tau_{wx}(\pi^{-1} \cdot \pi \cdot q)) = \pi \cdot \tau_{wx}(q) = \pi \cdot p$.

Theorem 5.5. *The group action $\pi \cdot \tau_w = \tau_{\pi \cdot w}$ is well-defined, where $w \in A^+$. Thus $\tau_w = \tau_{w'}$ implies $\tau_{\pi \cdot w} = \tau_{\pi \cdot w'}$*

Proof. We first note that applying the group action to a transition does not change the direction and that the length of a word does not change when applying the group action.

Suppose $w \in A^*$, $x \in A$ and $\tau_{wx}(q) = 0$. Then there are two cases:

- When starting with the configuration wqx, we walk of the left side of the tape. Since the length of the word does not change nor does the direction of each transition, when applying the group action, we must also walk of the left side of the tape when starting with the configuration $\pi \cdot w \pi \cdot q \pi \cdot x$.

- When starting with the configuration wqx, we do not walk of the left side of the tape and at each point of the computation, the number of right transitions taken is always less than or equal to the number of left transitions taken. This means that we will never reach a configuration wxp for some $p \in Q$. Since the length of the word does not change nor does the direction of each transition, when applying the group action, we do still not walk of the left side of the tape and at each point of the computation, the number of right transitions taken is always less than or equal to the number of left transitions taken.

We can thus conclude that $\tau_{wx}(q) = 0$ implies $\tau_{\pi \cdot (wx)}(\pi \cdot q) = 0$.

Now suppose that $\tau_{wx}(\bar{q}) = p$ for some $p \in Q$, which by construction means that $q_0 wx \vdash^* wxp$. Because each $\vdash$ corresponds to a transition and δ is equivariant, $q_0 wx \vdash^* wxp$ implies $\pi \cdot q_0 \pi \cdot (wx) \vdash^* \pi \cdot (wx) \pi \cdot p$. Where $\pi \cdot q_0 = q_0$ by definition. Thus $\tau_{wx}(\bar{q}) = p$ implies $\tau_{\pi \cdot (wx)}(\bar{q}) = \pi \cdot p$.

If we combine our results with Lemma 5.4, have see that the function τ_{wx} completely determines the function $\tau_{\pi \cdot (wx)}$. Thus if $\tau_w = \tau_{w'}$, then $\pi \cdot \tau_w = \pi \cdot \tau_{w'}$ for all $\pi \in G$. We can now conclude that our group action is well-defined. □

Theorem 5.6. *For an arbitrary G-2DFA $M = (Q, A, \delta, \{q_0\}, F)$, there exists a G-DFA that accepts the same language if the set $\{\tau_w \mid w \in A^+\}$ is orbit finite, where the group action on τ_w is $\pi \cdot \tau_w = \tau_{\pi \cdot w}$.*

Proof. Take G-2DFA $M = (Q, A, \delta, \{q_0\}, F)$. We define the deterministic G-automaton $M' = (Q', A, \delta', \{q_0'\}, F')$ the following way, where τ_w are the same τ_w as in theorem 4.2.

- $Q' = \{\tau_w \mid w \in A^+\}$

- $\delta'(\tau_w, x) = \tau_{wx}$
- q'_0 is a new state.
- $F' = \{\tau_w \mid \tau_w(\bar{q}) \in F\}$

We define the group action on q_0 as $\pi \cdot q'_0 = q'_0$, since $\{q'_0\}$ needs to be equivariant. Lemma 4.4 still holds, so $\tau_w = \tau_{w'}$ implies $\tau_{wx} = \tau_{w'x}$. This means again that our δ' is well defined. We have given a well-defined group action on Q', so Q' is a G-set. The alphabet A is the same, so still is an orbit finite G-set. We also see that δ' is equivariant:

$$\begin{aligned}
&\pi \cdot \delta'(\tau_w, x) \\
&= \pi \cdot \tau_{wx} && \text{(definition of } \delta'\text{)} \\
&= \tau_{\pi\cdot(wx)} && \text{(group action on } \tau_w\text{)} \\
&= \tau_{\pi\cdot w \pi\cdot x} && \text{(point-wise action on words)} \\
&= \delta'(\pi \cdot \tau_w, \pi \cdot x).
\end{aligned}$$

The initial states $\{q'_0\}$ is an equivariant set because $\pi \cdot q'_0 = q'_0$ by construction. The set of final states F is also equivariant because the language accepted by M is equivariant:

$$\begin{aligned}
&\tau_w \in F' \\
\iff &\tau_w(\bar{q}) \in F && \text{(By construction of } F'.\text{)} \\
\iff &q_0 w \vdash^* w q_f \text{ for some } q_f \in F && \text{(By construction of } \tau_w.\text{)} \\
\iff &w \in L && \text{(By acceptance of M.)} \\
\iff &\pi \cdot w \in L && \text{(Since } L \text{ is a } G\text{-language.)} \\
\iff &q_0(\pi \cdot w) \vdash^* (\pi \cdot w) q_f \text{ for some } q_f \in F && \text{(By acceptance of M.)} \\
\iff &\tau_{\pi\cdot w}(\bar{q}) \in F && \text{(By construction of } \tau_w.\text{)} \\
\iff &\tau_{\pi\cdot w} \in F' && \text{(By construction of } F'.\text{)}
\end{aligned}$$

Thus we have obtained that Q' is a G-set and, by our assumption that the set $\{\tau_w \mid w \in A^+\}$ is orbit finite, Q' is also orbit finite. We have also obtained that $\delta', \{q'_0\}$ and F are equivariant. The only thing left to prove is now that M' accepts the same language as M.

$$\begin{aligned}
&w \text{ accepted by } M \\
\iff &q_0 w \vdash^* w q_f \text{ for some } q_f \in F \\
\iff &\tau_w(\bar{q}) \in F \\
\iff &\tau_w \in F' \\
\iff &\delta'^*(q'_0, w) \in F' \\
\iff &w \text{ is accepted by } M'.
\end{aligned}$$

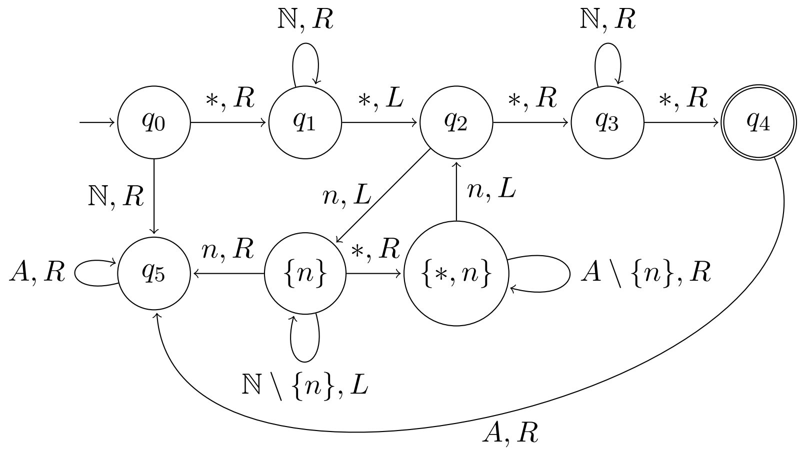
Thus there exists a G-DFA that accepts the same language as a G-2DFA $M = (Q, A, \delta, \{q_0\}, F)$ if the set $\{\tau_w \mid w \in A^+\}$ is orbit finite. $\square$

5.2 Orbit infinite examples

The last theorem means that the construction does work if the set $\{\tau_w \mid w \in A^+\}$ is orbit finite, but this is not always the case, as we will see in the following G-2DFA.

Example 5.7. The G-2DFA below accepts the language $\mathcal{L} = \{*d_0d_1 \dots d_n* \mid d_k \in \mathbb{N}, d_i \neq d_j \text{ for } k \in \mathbb{N} \text{ and } i \neq j\}$ in the equality symmetry. We define the group action on q_i as $\pi \cdot q_i = q_i$ for all $\pi \in G$ and $0 \leq i \leq 5$ and the group action on $*$ is also $\pi \cdot * = *$ for all $\pi \in G$. The alphabet A is $\mathbb{N} \cup \{*\}$.

The G-2DFA walks through the word until the letter $*$. It then checks if the last unchecked letter occurs another time in the word. If this is the case, we end up in q_5 and do not accept the word. If there are no letters occurring twice in between the two $*$, we eventually take the transition from q_2 to q_3 and reach a final configuration in q_4.

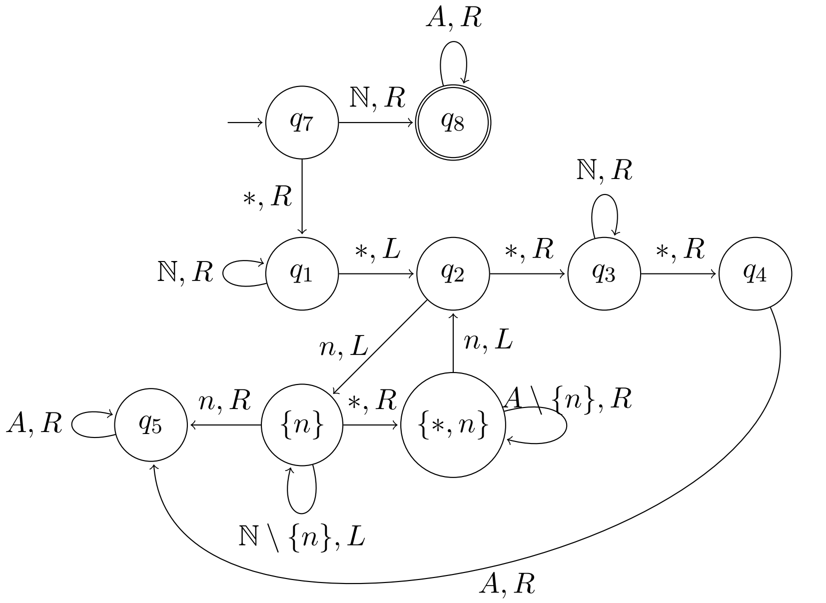


Let us now look at a subset that shows that $\{\tau_w \mid w \in A^+\}$ is not orbit finite for this G-2DFA. Take the subset $\{1, 12, 123, 1234, 12345, \dots\} \subseteq A^*$ and take two words $w, w' \in \{1, 12, 123, 1234, 12345, \dots\}$, where $w \neq w$. Now assume, without loss of generality, that the length of w is greater than the length of w'. Note that this implies that w contains a letter that does not occur in w'. Even if we take a permutation $\pi \in G$ and apply it to w', w still must contain a letter that does not occur in $\pi \cdot w'$. Let us cal this letter k. Then $\tau_{w'}(\{k\}) = 0$ and $\tau_w(\{k\}) = q_5$

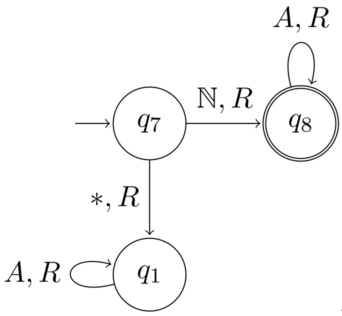
Because we can for each permutation $\pi \in G$, find a $\{k\} \in Q$ such that $\tau_w(\{k\}) \neq \tau_{\pi \cdot w'}(\{k\})$, we can conclude that $\tau_{w'}$ and τ_w are in different orbits. Since this holds for every two words in $\{1, 12, 123, 1234, 12345, \dots\}$, we can conclude that the construction does not work, since $\{\tau_1, \tau_{12}, \tau_{123}, \tau_{1234}, \tau_{12345}\} \subseteq Q = \{q_0'\} \cup \{\tau_w \mid w \in A^+\}$ is not orbit finite.

Note that if we take this set $\{1, 12, 123, 1234, 12345, \dots\}$ and add a $*$ before each word: $\{*1, *12, *123, *1234, *12345, \dots\}$, we also obtain a set that can be used to show that the language $\mathcal{L} = \{*d_0 d_1 \dots d_n * \mid d_k \in \mathbb{N}, d_i \neq d_j \text{ for } k \in \mathbb{N} \text{ and } i \neq j\}$ cannot be accepted by the Myhill-Nerode theorem for G-sets.

We can now also use this counterexample to show that it is not true that the set $\{\tau_w \mid w \in A^+\}$ is orbit finite if a G-2DFA that accepts the language $\mathcal{L}$ can be accepted by a G-DFA.



The G-2DFA above uses the G-2DFA of Example 5.7 for words that are already not going to be in the language. This allows us to create a G-2DFA that accepts an easy language, but where the set $\{\tau_w \mid w \in A^+\}$ is not orbit finite. We can see that the language that the G-automaton above accepts is $\mathcal{L} = \{nw \mid n \in \mathbb{N} \text{ and } w \in A^*\}$, which can be accepted by the G-DFA below. We can thus conclude that the statement "The set $\{\tau_w \mid w \in A^+\}$ is orbit finite if a G-2DFA that accepts language $\mathcal{L}$ can be accepted by a G-DFA" is not true.



.

A possible assumption is to require an automaton to be reachable. This notion refers to automata, where each state can be visited with some computation.

Definition 5.8. A G-2DFA $(Q, A, \delta, \{q_0\}, F)$ is called reachable if for every state $q \in Q$, there exist $w, w' \in A^*$ such that $q_0ww' \vdash wqw'$.

This assumption would not help improve the theorem, since the G-DFA given above is reachable.

Suppose we have a G-2DFA $(Q, A, \delta, \{q_0\}, F)$ that accepts language $\mathcal{L}$. We have seen with these examples that the implication

$$\text{there exists a } G\text{-DFA that accepts } \mathcal{L} \implies \{\tau_w \mid w \in A^+\} \text{ is orbit finite}$$

does not hold, even if the G-2DFA is reachable.

Chapter 6

Related Work

The main source of this thesis is [1]. This paper contains the notion of G-automata, nominal sets and the Myhill-Nerode theorem for G-sets and nominal G-sets. The language used to show that we cannot accept all languages with a G-DFA that a G-2DFA accepts also originates from this paper.

In this thesis, we combined the proof of expressive equivalence of 2DFA and DFA used in [6] with the notion of G-2DFA from [1] to show that using an extra condition, the construction from the proof still works for G-2DFA and G-DFA. The language from [1] was then used as an counterexample to show that the construction does not always work if the language can be accepted by a G-DFA.

Chapter 7

Conclusions

In this thesis, we have seen what G-automata and 2DFAs are and how a DFA can accept the same language as an 2DFA using a construction. Theorem 5.6 was our main result and showed that this construction for G-2DFA to G-DFA works if the set $\{\tau_w \mid w \in A^*\}$ is orbit finite. Lastly, we showed that the converse was not true and that the assumption that the G-2DFA was reachable did also not cause the converse to be true.

7.1 Future work

A good extension to this thesis would be to improve Theorem 5.6, by proving that there exists a G-DFA that accept the same language if and only if the set $\{\alpha_w \mid w \in A^*\}$ is orbit finite. This construction, with α, would ensure that all function values have a purpose, since it occurs in a computation of a word in the language, while this was not the case for τ_w.

$$\alpha_{wx}(q) = \begin{cases} p & \text{if } wqx \vdash^* wxp \text{ and } \exists w' \in A^* \text{ with } q_0wxw' \vdash^* wqxw' \vdash^* wxw'q_f \text{ for some } q_f \in F \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_w(\bar{q}) = \begin{cases} p & \text{if } q_0w \vdash^* wp \\ 0 & \text{otherwise} \end{cases}$$

Bibliography

[1] Mikolaj Bojanczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014.

[2] Mikolaj Bojanczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. Two-variable logic on words with data. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 7–16. IEEE Computer Society, 2006.

[3] Frans Oort Hendrik Lenstra Jr. and Ben Moonen. Groepentheorie, 2014. lecture notes. Radboud University.

[4] Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science*, 134(2):329–363, 1994.

[5] Sheng Yu Andrei Paun. Implementation and application of automata. *Berlin, Heidelberg.*

[6] Jeffrey Shallit. *A Second Course in Formal Languages and Automata Theory.* Cambridge University Press, 2009.

[7] Thomas A. Sudkamp. *An Introduction to the Theory of Computer Science Languages and Machines.* Pearson, 3 edition, 2006.