Michal Kunc | Alexander Okhotin

On deterministic two-way finite automata over a unary alphabet

Turku Centre for Computer Science

TUCS Technical Report
No 950, February 2011

On deterministic two-way finite automata over a unary alphabet

Michal Kunc
Department of Mathematics, Masaryk University,
Brno, Czech Republic
kunc@math.muni.cz
Alexander Okhotin
Academy of Finland, *and*
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FIN–20014, Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 950, February 2011

Abstract

A framework for the study of two-way deterministic finite automata (2DFA) over a one-letter alphabet is developed, generalizing the concept of transformation semigroups to the case of bi-directional motion. It allows analyzing the behaviour of automata globally, on all inputs at once, rather than locally, following a particular computation, as per the mainstream approach to two-way computations. The method is used to show that transforming an n-state unary 2DFA to an equivalent sweeping 2DFA requires exactly $n+1$ states, and that exactly $\max_{0 \leqslant \ell \leqslant n} g(n-\ell)+\ell+1$ states, where $g(k)$ is the maximum order of a permutation of k elements, are needed for a similar transformation of a unary 2DFA to a one-way automaton.

Keywords: Finite automata, two-way automata, transformation semigroups, unary languages, descriptional complexity.

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

Two-way deterministic finite automata (2DFA) were introduced in the famous paper by Rabin and Scott [15] alongside the one-way nondeterministic automata (1NFA). Both kinds of automata recognize the same language family as the one-way deterministic finite automata (1DFA). However, they are substantially different in terms of succinctness of description, and the number of states needed to represent a language by one type of finite automata is sometimes much greater than for another type.

While the methods for determining the number of states in one-way automata, both deterministic and nondeterministic, are well-known, and the main descriptional complexity questions [6] have been researched to extinction, the succinctness issues of two-way automata have proved to be truly challenging. The question of whether 2DFAs can simulate their nondeterministic counterpart (2NFA) with only a polynomial blowup has attracted a lot of attention due to its close connection to the *L vs. NL* problem in the complexity theory [2], yet no definite answers could be found. Even such a basic question as the precise number of states in a 1DFA needed to simulate an n-state 2DFA could not be determined precisely for almost half a century: the $(n+1)^{n+1}$ upper bound by Shepherdson [17] was approached by a relatively close $(\frac{n-5}{2})^{\frac{n-5}{2}}$ lower bound by Moore [13], but only a few years ago the exact value $n(n^n - (n-1)^n)$ was finally determined by Kapoutsis [8]. Simulations of 2NFAs by simpler automata, first studied by Vardi [19], were also determined precisely by Kapoutsis [8]. The complexity of operations on 2DFAs has recently been investigated by Jirásková and Okhotin [7].

The state complexity of 2DFAs in the seemingly trivial case of a one-letter alphabet turned out to be challenging as well. The first study of unary 2DFAs was undertaken by Chrobak [4], who has sketched an argument that an n-state 2DFA over a unary alphabet can be simulated by a $\Theta(g(n))$-state 1DFA, where $g(n) = e^{(1+o(1))\sqrt{n \ln n}}$ is the maximal order of a permutation on n elements, known as *Landau's function* [9]. Further work in this direction was done by Mereghetti and Pighizzini [10] and by Geffert, Mereghetti and Pighizzini [5], who similarly gave good asymptotic estimations of the 2NFA–1DFA tradeoff. Their approach lies with considering only the periodic part of the language and using a general upper bound on the starting point of its periodicity, and thus leads only to asymptotic succinctness tradeoffs. No optimal simulations between unary two-way and one-way automata are known up to date. The first such results are obtained in this paper for the deterministic case.

The goal of this paper is to develop a general framework for reasoning about unary 2DFAs, which would cover both their periodic and non-periodic behaviour in a unified way, and subsequently allow determining the precise number of states needed to represent particular languages. Recalling the algebraic representation of a 1DFA by a monoid of *partial transformations* of its set of states (Section 3), the paper proceeds with generalizing this concept to semigroups of *two-way transformations* representing bi-directional motion (Section 4). Each 2DFA over a unary alphabet is represented by a *monogenic subsemigroup* of the semigroup of two-way transformations, and the properties of such subsemigroups are gradually worked out in Section 5. The final result is the precise characterization of monogenic semigroups of two-way transformations on n states: their index ℓ (that is, the starting point of the periodicity) and period $\operatorname{lcm}(p_1, \ldots, p_k)$ satisfy the inequality $p_1 + \cdots + p_k + \ell \leqslant n + 1$.

Based on this analysis, the use of the states by a unary 2DFA is explained as follows. It is in fact found that 2DFAs can do just two things which lower the required number of states, as compared to 1DFAs:

1. count divisibility separately for powers of distinct primes;

2. when counting up to a finite bound ℓ, they can count one step less than one would expect, and then use one of the cycles to distinguish between strings of length ℓ and $\ell + 1$.

This understanding is used in the rest of the paper to establish precise state complexity results for several classes of unary 2DFAs.

The first to be considered is the subclass of *sweeping 2DFAs*, studied in Section 6. For an arbitrary alphabet, as independently proved by Berman [1] and by Micali [12], the succinctness blowup from general 2DFAs to sweeping 2DFAs is exponential. For a unary alphabet, Mereghetti and Pighizzini [11] established a transformation of an n-state 2NFA to a sweeping 2NFA with $O(n^2)$ states. For 2DFAs, Chrobak [4] has claimed without a proof that every unary 2DFA can be made sweeping without increasing the number of states. The claim was not substantiated, and the best result known in the literature is the $O(n^2)$ bound for unary 2NFAs from the work of Mereghetti and Pighizzini [11]. This paper determines the exact number of states needed to make an n-state 2DFA sweeping, which turns out to be $n + 1$.

Section 7 considers the standard question of converting an n-state 2DFA to an equivalent 1DFA. Chrobak's [4] asymptotic estimation $\Theta(g(n))$ is hereby improved to the precise expression, which is $\max_{0 \leqslant \ell \leqslant n} g(n - \ell) + \ell + 1$. The same function applies to the 2DFA to 1NFA transformation.

2 Two-way deterministic automata

Given an input string w, a 2DFA operates on a tape containing the string $\vdash w \dashv$, where $\vdash$ and $\dashv$ are special symbols known as the left-end marker and the right-end marker, respectively. According to the standard definition, a 2DFA begins its computation at the left-end marker and accepts at the right-end marker. In this paper, the definition is extended to allow acceptance on both sides: this leads to symmetric constructions and allows avoiding some awkward exceptions in the results. Furthermore, to simplify mathematical treatment of these automata, accepting states are replaced with moving beyond the markers.

Definition 1. *A* 2DFA (with two-sided acceptance) *is a quadruple* $\mathcal{A} = (\Sigma, Q, q_1, \delta)$*, in which* Σ *is a finite alphabet with* $\vdash, \dashv \notin \Sigma$*,* Q *is a finite set of states,* $q_1 \in Q$ *is the initial state and* $\delta \colon Q \times (\Sigma \cup \{\vdash, \dashv\}) \to Q \times \{-1, +1\}$ *is a partially defined transition function.*

A computation *of* $\mathcal{A}$ *on a string* $w = a_1 \ldots a_\ell \in (\Sigma \cup \{\vdash, \dashv\})^+$ *beginning with a configuration* (p_0, i_0) *is the longest sequence* $(p_0, i_0), (p_1, i_1), \ldots$*, finite or infinite, in which*

- $p_t \in Q$ *and* $1 \leqslant i_t \leqslant \ell$ *for each* t*-th step, except maybe for the last element of a finite computation, which may have* $i_t \in \{0, \ell+1\}$*;*

- *every next element* (p_t, i_t)*, if it is defined, satisfies* $\delta(p_{t-1}, a_{i_{t-1}}) = (p_t, d_t)$ *and* $i_t = i_{t-1} + d_t$*.*

The computation beginning with a given configuration (p_0, i_0) *is always uniquely defined. It is* accepting *if it is finite and its last configuration* (p_f, i_f) *satisfies* $i_f \in \{0, \ell+1\}$*. In this case, this last configuration is denoted by* $\delta^*_w(p_0, i_0) = (p_m, i_m)$*.*

Then $\mathcal{A}$ *is said to* accept *an input string* $w \in \Sigma^*$ *if* $\delta^*_{\vdash w \dashv}(q_1, 1)$ *is defined. Define* $L(\mathcal{A}) = \{w \mid \mathcal{A} \text{ accepts } w\}$*.*

Besides accepting, a 2DFA may explicitly *reject* by encountering an undefined transition, or it may *loop*, in the sense that the sequence defined above continues indefinitely within the margins of the tape. In both cases the string is not in the language.

3 Ordinary transformation semigroups

For any set N, the set of partial functions from N to N (called partial transformations of N) is known to form a monoid with respect to the function composition $\circ$. This monoid is denoted by $\mathcal{PT}_N$. For every $h \in \mathcal{PT}_N$, its mth power $\underbrace{h \circ \cdots \circ h}_{m}$ will be denoted by h^m.

It is well-known that the behaviour of a (partial) 1DFA over Σ, with a set of states Q, is characterized by its *transitions monoid*, which is the submonoid of $\mathcal{PT}_Q$ consisting of all actions of strings δ_w on this automaton, where $w \in \Sigma^*$. This submonoid is generated by the actions of letters, and the function composition takes the form $\delta_u \circ \delta_v = \delta_{vu}$. A more detailed introduction was given by Perrin [14].

A semigroup generated by one of its elements, that is, with all elements being powers of this element, is called *monogenic*. If a monogenic semigroup S generated by s is finite, then there exist positive integers $i \neq j$ satisfying

$s^i = s^j$. In this case, the least positive integer i, for which there exists $j > i$ with $s^i = s^j$, is called the *index* of S. Then the *period* of S is the least number $p \geqslant 1$ with $s^i = s^{i+p}$. The index and period determine a monogenic semigroup up to isomorphism.

In the case of the monoid $\mathcal{PT}_N$, the index and period of the subsemigroup generated by a partial transformation h can be easily calculated from the structure of h when viewed as an oriented graph of out-degree 1, with the set of nodes N and with $h(\alpha) = \beta$ represented by an arc $\alpha \to \beta$. If $\alpha \in N$ is such, that $h^i(\alpha) = \alpha$ for some $i \geqslant 1$, then the *h-cycle* containing α is the set $\{h^j(\alpha) \mid 0 \leqslant j \leqslant \ell - 1\}$, where $\ell \geqslant 1$ is the smallest number satisfying $h^\ell(\alpha) = \alpha$; this ℓ is called the *length* of the h-cycle. An h-tail is any maximal subset of N consisting of elements not belonging to any h-cycle, in which for any two elements α and β there exists k satisfying either $h^k(\alpha) = \beta$ or $h^k(\beta) = \alpha$. There is a one-to-one correspondence between h-tails and elements $\alpha \in N$ with $h(\beta) \neq \alpha$ for every $\beta \in N$; the h-tail given by such α consists of all elements $h^k(\alpha)$ reachable from α which do not belong to any cycle. Note that every tail leads either into a cycle, or into a *dead element* where h is not defined; in the latter case it shall be called an *orphan tail.* The number of elements of N which belong to a given tail is called the *length* of the tail. Every element of N belongs to a cycle or to a tail, but not to both. Furthermore, note that the cycles are necessarily disjoint, while tails are not.

Lemma 1. *For any $h \in \mathcal{PT}_N$, the index of the subsemigroup generated by h is equal to the length of the longest h-tail (it equals* 1 *if there is no tail) and its period is equal to the least common multiple of the lengths of all h-cycles.*

4 Two-way transformation semigroups

In this section, the above graph-theoretic outlook on partial transformations is generalized to the case of bi-directional motion. This notation allows extending Lemma 1 to the two-way case, which shall be used as a basis for state complexity results for unary 2DFAs.

Consider the behaviour of an n-state deterministic two-way automaton on any nonempty string w. It enters the string in a certain state either from its first or from its last symbol. Then the automaton may either loop inside the string, or eventually leave the string by going to the left of its first symbol or to the right of its last symbol in a certain new state.

The symbol $\triangleright$ represents entering a string from the left. Entering the leftmost symbol of w in a state i is described by a pair $(\triangleright, i)$. Then, leaving w by going to the right of its last symbol in a state j is denoted by a pair $(\triangleright, j)$, because this means entering the string to the right of w on its first symbol. Symmetrically, the symbol $\triangleleft$ represents entering a string from the right. If the automaton enters the last symbol of w from the right in a state

k, this is represented by a pair $(\triangleleft, k)$, and a pair $(\triangleleft, \ell)$ also represents leaving w by going to the left beyond its first symbol in a state ℓ.

Thus the behaviour of an n-state 2DFA on any string can be represented as a partial transformation of a set $N = \{\triangleright, \triangleleft\} \times \{1, \ldots, n\}$. These transformations shall be called *two-way transformations* on $\{1, \ldots, n\}$ and denoted by the symbols f and g. They shall be depicted by oriented graphs, such as the one in Figure 1. The following notation for each half of the set N shall be employed: $N_\triangleright = \{\triangleright\} \times \{1, \ldots, n\}$ and $N_\triangleleft = \{\triangleleft\} \times \{1, \ldots, n\}$.

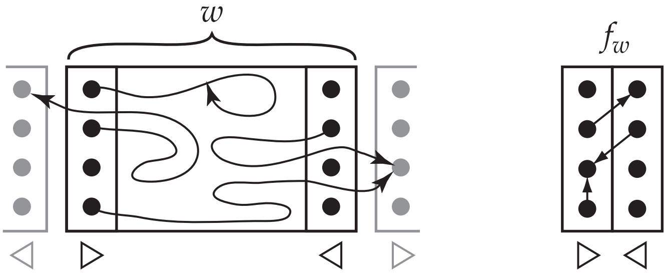

Figure 1: Representing behaviour of 2DFA on w as $f_w^{\mathcal{A}} \colon N \to N$.

Definition 2. *Let $\mathcal{A} = (\Sigma, Q, q_1, \delta)$ be a 2DFA with $Q = \{q_1, \ldots, q_n\}$, let $w \in (\Sigma \cup \{\vdash, \dashv\})^+$ be a nonempty string. Then the* behaviour *of $\mathcal{A}$ on w is a two-way transformation $f_w^{\mathcal{A}}$ on $\{1, \ldots, n\}$ defined as follows:*

$$f_w^{\mathcal{A}}(\triangleright, i) = \begin{cases} (\triangleright, j), & \text{if } \delta_w^*(q_i, 1) = (q_j, |w|+1), \\ (\triangleleft, j), & \text{if } \delta_w^*(q_i, 1) = (q_j, 0), \\ \text{undefined}, & \text{if } \delta_w^*(q_i, 1) \text{ is undefined.} \end{cases}$$

$$f_w^{\mathcal{A}}(\triangleleft, i) = \begin{cases} (\triangleright, j), & \text{if } \delta_w^*(q_i, |w|) = (q_j, |w|+1), \\ (\triangleleft, j), & \text{if } \delta_w^*(q_i, |w|) = (q_j, 0), \\ \text{undefined}, & \text{if } \delta_w^*(q_i, |w|) \text{ is undefined.} \end{cases}$$

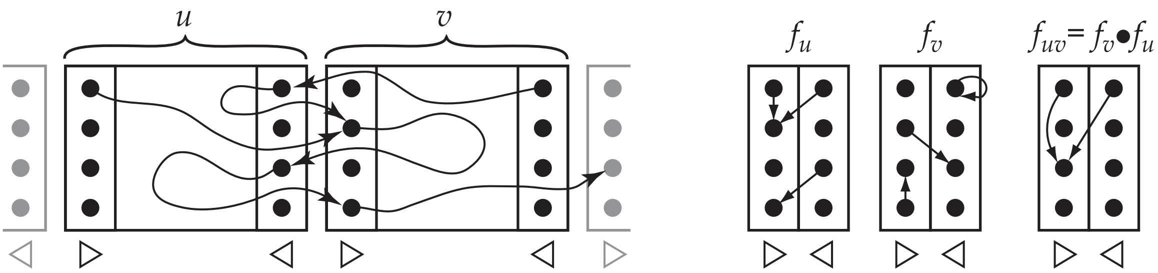

Figure 2: $g \bullet f$.

Once the behaviour of the automaton on some strings $u, v \in \Sigma^+$ is known to be f and g, respectively, its behaviour on their concatenation uv can be obtained as a certain *product* $g \bullet f$ (note that this product is unrelated to the plain function composition $g \circ f$). The value $f_{uv}^{\mathcal{A}}(\alpha)$ can be inferred from $f_u^{\mathcal{A}}$ and $f_v^{\mathcal{A}}$ by the following chain of equivalences. An equality $f_{uv}^{\mathcal{A}}(\triangleright, i) = (\triangleright, j)$ holds, that is, $\delta_{uv}^*(q_i, 1) = (q_j, |uv| + 1)$, if and only if there exists a number $k \in \mathbb{N}$ representing how many times the computation on uv crosses the boundary between u and v, and the intermediate states $i_\ell \in \{1, \dots, n\}$ with $\ell \in \{0, \dots, k-1\}$ and $j_\ell \in \{1, \dots, n\}$ with $\ell \in \{1, \dots, k-1\}$ entered by the automaton on the successive traversals of this boundary, with some computations inside u and v occurring in between: $\delta_u^*(q_i, 1) = (q_{i_0}, |u| + 1)$, $\delta_v^*(q_{i_{k-1}}, 1) = (q_j, |v| + 1)$ and $\delta_u^*(q_{j_\ell}, |u|) = (q_{i_\ell}, |u| + 1)$ and $\delta_v^*(q_{i_{\ell-1}}, 1) = (q_{j_\ell}, 0)$, for $\ell = 1, \dots, k-1$. These conditions can be reformulated purely in terms of the transformations $f_u^{\mathcal{A}}$ and $f_v^{\mathcal{A}}$ as follows: $f_u^{\mathcal{A}} \circ (f_v^{\mathcal{A}} \circ f_u^{\mathcal{A}})^\ell((\triangleright, i)) = (\triangleright, i_\ell)$ for $\ell \in \{0, \dots, k-1\}$, $(f_v^{\mathcal{A}} \circ f_u^{\mathcal{A}})^\ell((\triangleright, i)) = (\triangleleft, j_\ell)$ for $\ell \in \{1, \dots, k-1\}$, and $(f_v^{\mathcal{A}} \circ f_u^{\mathcal{A}})^k(\alpha) = (\triangleright, j)$.

Dealing with all other possibilities in the same way leads to the following definition of $g \bullet f$. For every $\alpha \in N_\triangleright$, the value $(g \bullet f)(\alpha)$ is defined as follows.

- If there exists $k \in \mathbb{N}$ such that $f \circ (g \circ f)^\ell(\alpha) \in N_\triangleright$ for $\ell = 0, \dots, k-1$, $(g \circ f)^\ell(\alpha) \in N_\triangleleft$ for $\ell = 1, \dots, k-1$ and $(g \circ f)^k(\alpha) \in N_\triangleright$, then $(g \bullet f)(\alpha) = (g \circ f)^k(\alpha)$.

- If there exists $k \in \mathbb{N}_0$ such that $f \circ (g \circ f)^{\ell-1}(\alpha) \in N_\triangleright$ and $(g \circ f)^\ell(\alpha) \in N_\triangleleft$ for $\ell = 1, \dots, k$ and $f \circ (g \circ f)^k(\alpha) \in N_\triangleleft$, then $(g \bullet f)(\alpha) = f \circ (g \circ f)^k(\alpha)$.

- It is undefined otherwise.

Symmetrically, the value $(g \bullet f)(\alpha)$ for $\alpha \in N_\triangleleft$ is defined as follows:

- If there exists $k \in \mathbb{N}$ such that $g \circ (f \circ g)^\ell(\alpha) \in N_\triangleleft$ for $\ell = 0, \dots, k-1$, $(f \circ g)^\ell(\alpha) \in N_\triangleright$ for $\ell = 1, \dots, k-1$ and $(f \circ g)^k(\alpha) \in N_\triangleleft$, then $(g \bullet f)(\alpha) = (f \circ g)^k(\alpha)$.

- If there exists $k \in \mathbb{N}_0$ such that $g \circ (f \circ g)^{\ell-1}(\alpha) \in N_\triangleleft$ and $(f \circ g)^\ell(\alpha) \in N_\triangleright$ for $\ell = 1, \dots, k$ and $g \circ (f \circ g)^k(\alpha) \in N_\triangleright$, then $(g \bullet f)(\alpha) = g \circ (f \circ g)^k(\alpha)$.

- It is undefined otherwise.

As a historical reference, one could recall a similar product defined by Birget [3] in the special case of 2DFAs with disjoint "right-moving" and "left-moving" states.

Proposition 1. *Let $u, v \in (\Sigma \cup \{\vdash, \dashv\})^+$ be nonempty strings. Then $f_{uv}^{\mathcal{A}} = f_v^{\mathcal{A}} \bullet f_u^{\mathcal{A}}$.*

A natural question is whether all two-way transformations may occur as behaviours of some automata on some strings, and the answer is negative. Let f be a behaviour of some automaton on a string $w \in \Sigma^+$, let $(\triangleright, i) \in N_\triangleright$

and assume $f\big((\triangleright, i)\big) = (\triangleright, j) \in N_\triangleright$, as illustrated in Figure 3. Then the computation of the automaton going through w to the state j beyond w should pass through the last symbol of w in some state k, from where the automaton went to the right. Accordingly, there should exist a state k with $f\big((\triangleleft, k)\big) = (\triangleright, j)$.

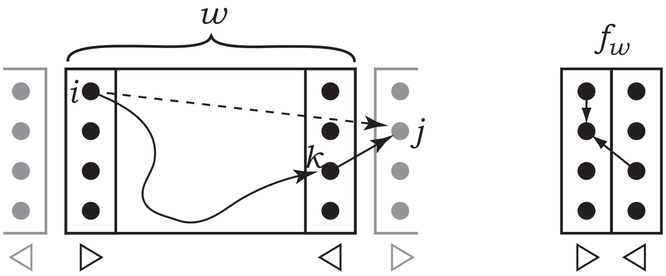

Figure 3: Condition (1).

A symmetric condition applies to elements of $N_\triangleleft$, and the entire necessary condition can be succinctly stated as follows. For all $\alpha, \beta \in N$, if both α and β belong to $N_\triangleright$ or both belong to $N_\triangleleft$, this is denoted by $\alpha \sim \beta$ and represents entering strings in the same direction. Then

$$\forall \alpha \in N \colon f(\alpha) \sim \alpha \implies \exists \beta \in N \colon \beta \nsim \alpha \ \wedge \ f(\beta) = f(\alpha). \tag{1}$$

Denote by $\mathcal{TT}_n$ the set of two-way transformations on $\{1, \ldots, n\}$ satisfying this condition.

Two-way transformations that occur as behaviours of some automata on one-symbol strings $w = a \in \Sigma$ must satisfy a stronger condition:

$$f_a^{\mathcal{A}}(\triangleright, i) = f_a^{\mathcal{A}}(\triangleleft, i) \quad (\text{for all } i \in \{1, \ldots, n\}) \tag{2}$$

And conversely, if a two-way transformation satisfies the condition (2), then it is a behaviour of some letter in some 2DFA. Such two-way transformations shall be called *distinguished*.

Lemma 2. *Every $f \in \mathcal{TT}_n$ is a product of two distinguished two-way transformations.*

Proof. Define these distinguished two-way transformations $f_a^{\mathcal{A}}$ and $f_b^{\mathcal{A}}$ as follows. For every $\alpha \in N$ with $f(\alpha)$ undefined, $f_a^{\mathcal{A}}(\alpha)$ is undefined if $\alpha \in N_\triangleright$, and $f_b^{\mathcal{A}}(\alpha)$ is undefined if $\alpha \in N_\triangleleft$. For every $\alpha \in N$ with $f(\alpha) \nsim \alpha$, define $f_a^{\mathcal{A}}(\alpha) = f(\alpha)$ if $\alpha \in N_\triangleright$, and $f_b^{\mathcal{A}}(\alpha) = f(\alpha)$ if $\alpha \in N_\triangleleft$. Clearly, $(f_b^{\mathcal{A}} \bullet f_a^{\mathcal{A}})(\alpha) = f(\alpha)$ for every such α.

If $\alpha, f(\alpha) \in N_\triangleright$, then, by (1), there exists $(\triangleleft, i) \in N_\triangleleft$ with $f((\triangleleft, i)) = f(\alpha)$. Define $f_a^{\mathcal{A}}(\alpha) = (\triangleright, i)$. Since $f_b^{\mathcal{A}}((\triangleright, i)) = f_b^{\mathcal{A}}((\triangleleft, i))$ has already been defined as $f((\triangleleft, i)) \in N_\triangleright$ at the previous step, $(f_b^{\mathcal{A}} \bullet f_a^{\mathcal{A}})(\alpha) = (f_b^{\mathcal{A}} \circ f_a^{\mathcal{A}})(\alpha) = f_b^{\mathcal{A}}((\triangleright, i)) = f((\triangleleft, i)) = f(\alpha)$, as required.

The last case of $\alpha, f(\alpha) \in N_\triangleleft$ is symmetric. By (1), there is $(\triangleright, i) \in N_\triangleright$ with $f((\triangleright, i)) = f(\alpha)$, and it is sufficient to define $f_b^{\mathcal{A}}(\alpha) = (\triangleleft, i)$. □

Proposition 2. *A two-way transformation belongs to $\mathcal{TT}_n$ if and only if it is a behaviour of some 2DFA on some string.*

Proof. Since condition (1) is satisfied by the behaviour of any automaton on an arbitrary string, such a behaviour belongs to $\mathcal{TT}_n$. Conversely, Lemma 2 shows that every element $f \in \mathcal{TT}_n$ is of the form $f = f_b^{\mathcal{A}} \bullet f_a^{\mathcal{A}}$, where $f_a^{\mathcal{A}}, f_b^{\mathcal{A}}$ are behaviours of a 2DFA $\mathcal{A}$ on some letters a, b. Then, according to Proposition 1, it holds $f = f_{ab}^{\mathcal{A}}$. □

Proposition 3. *The set $\mathcal{TT}_n$ equipped with the product $\bullet$ is a semigroup.*

Though this fact could be proved directly by establishing the associativity of $\bullet$ and demonstrating that it preserves the class $\mathcal{TT}_n$, there is a simpler argument relying on the correspondence of $\mathcal{TT}_n$ to 2DFAs and on the associativity of the concatenation.

Proof. First note that, given finitely many elements of $\mathcal{TT}_n$, not only each of them is a behaviour of some 2DFA on some string by Proposition 2, but it can also be assumed that all of them are behaviours of the same automaton on different strings; this can be achieved simply by taking strings over different alphabets. Then the closure of $\mathcal{TT}_n$ under $\bullet$ follows directly from Proposition 1. Furthermore, the product $\bullet$ is associative on $\mathcal{TT}_n$, because Proposition 1 implies $f_w^{\mathcal{A}} \bullet (f_v^{\mathcal{A}} \bullet f_u^{\mathcal{A}}) = f_{uvw}^{\mathcal{A}} = (f_w^{\mathcal{A}} \bullet f_v^{\mathcal{A}}) \bullet f_u^{\mathcal{A}}$. □

Note that $\mathcal{TT}_n$ is not a monoid for the lack of an identity element. Though there exists an identity two-way transformation defined by $e(\alpha) = \alpha$, it does not satisfy condition (1). For this reason, the semigroup representation of two-way automata considers only their computations on nonempty inputs. The empty string will be reintroduced later, when turning from semigroups back to automata.

The semigroup $\mathcal{TT}_n$ will be called the *full two-way transformation semigroup* on $\{1, \ldots, n\}$. Since elements of $\mathcal{TT}_n$ will often be considered also as ordinary transformations of the set N, a different notation for powers with respect to the operation of $\mathcal{TT}_n$ has to be introduced: $f^{\bullet m}$ stands for $\underbrace{f \bullet \ldots \bullet f}_{m}$.

There is the following formal connection between computations of 2DFAs and semigroups $\mathcal{TT}_n$:

Proposition 4. *Let $\mathcal{A} = (\Sigma, Q, q_1, \delta)$ be a 2DFA, where $Q = \{q_1, \ldots, q_n\}$, and consider the subsemigroup of $\mathcal{TT}_n$ generated by distinguished two-way transformations $f_a^{\mathcal{A}}$, for $a \in \Sigma \cup \{\vdash, \dashv\}$. Then*

$$L(\mathcal{A}) = \big\{\, a_1 \ldots a_\ell \bigm| (f_{\dashv}^{\mathcal{A}} \bullet f_{a_\ell}^{\mathcal{A}} \bullet \ldots \bullet f_{a_1}^{\mathcal{A}} \bullet f_{\vdash}^{\mathcal{A}})\big((\triangleright, 1)\big) \textit{ is defined} \,\big\}.$$

In particular, for a given automaton $\mathcal{A}$, the membership of a string $a_1 \ldots a_\ell \in \Sigma^+$ in $L(\mathcal{A})$ depends only on the element $f_{a_1 \ldots a_\ell}^{\mathcal{A}} = f_{a_\ell}^{\mathcal{A}} \bullet \ldots \bullet f_{a_1}^{\mathcal{A}}$ of the subsemigroup generated by $\{f_a^{\mathcal{A}} \mid a \in \Sigma\}$.

This proposition, in particular, shows that if two strings $u, v \in \Sigma^+$ satisfy $f_u^{\mathcal{A}} = f_v^{\mathcal{A}}$, then they represent the same element of the syntactic monoid of $L(\mathcal{A})$, because $f_{xuy}^{\mathcal{A}} = f_y^{\mathcal{A}} \bullet f_u^{\mathcal{A}} \bullet f_x^{\mathcal{A}} = f_y^{\mathcal{A}} \bullet f_v^{\mathcal{A}} \bullet f_x^{\mathcal{A}} = f_{xvy}^{\mathcal{A}}$ for every $x, y \in \Sigma^*$.

Proof. By the definition of a 2DFA, $w \in L(\mathcal{A})$ if and only if $\delta^*_{\vdash w \dashv}((q_1, 1))$ is defined. According to the definition of $f_{\vdash w \dashv}^{\mathcal{A}}$, this is in turn equivalent to $f_{\vdash w \dashv}^{\mathcal{A}}(\triangleright, 1)$'s being defined. Since, by Proposition 1, $f_{\vdash a_1 \ldots a_\ell \dashv}^{\mathcal{A}} = f_{\dashv}^{\mathcal{A}} \bullet f_{a_\ell}^{\mathcal{A}} \bullet \ldots \bullet f_{a_1}^{\mathcal{A}} \bullet f_{\vdash}^{\mathcal{A}}$, the equality follows. □

Consider Proposition 4 in the case of a unary alphabet $\Sigma = \{a\}$. Then it asserts that

$$L(\mathcal{A}) = \left\{\, a^\ell \;\middle|\; (f_{\dashv}^{\mathcal{A}} \bullet (f_a^{\mathcal{A}})^{\bullet \ell} \bullet f_{\vdash}^{\mathcal{A}})\big((\triangleright, 1)\big) \text{ is defined} \,\right\},$$

and thus the membership of a^ℓ in $L(\mathcal{A})$ is determined by the element $(f_a^{\mathcal{A}})^{\bullet \ell}$ of the (monogenic) subsemigroup of $\mathcal{TT}_n$ generated by $f_a^{\mathcal{A}}$. Thus understanding the structure of monogenic semigroups generated by distinguished two-way transformations is essential for characterizing the power of 2DFAs over a unary alphabet.

5 Monogenic subsemigroups of $\mathcal{TT}_n$

Let $f \in \mathcal{TT}_n$ be any fixed element of the two-way transformation semigroup. This element represents the behaviour of some 2DFA on some string x, and computations of this 2DFA on long sequences x^m of its copies can be explained entirely in terms of a monogenic subsemigroup of $\mathcal{TT}_n$. In particular, this applies to computations on strings over a unary alphabet, where $x = a$ and f is a distinguished two-way transformation. However, in general, f need not be distinguished, and the following results apply to more general computations on x^m for a fixed $x \in \Sigma^+$ and variable m.

5.1 The distance travelled after i steps

This setting leads to the following model. Consider a bi-infinite string of fs, with the copies of f numbered by integers. For any $\alpha \in N_\triangleright$, consider the computation starting from α in copy number 0. At every j-th step, the automaton proceeds to the neighbouring instance of f: to the right if $f^j(\alpha) \in N_\triangleright$ and to the left if $f^j(\alpha) \in N_\triangleleft$. Such a computation is illustrated in Figure 4, where $f(\alpha) = \beta$, $f^2(\alpha) = \gamma$, ..., $f^6(\alpha) = \alpha$, etc. Similarly one can consider computations starting from $\alpha' \in N_\triangleleft$, numbering the instances of f in the reverse direction.

Consider the *distance* travelled in such a computation. In Figure 4, three steps of computation move the head back by one square, while six steps of computation result in moving forward by two squares. This shall be denoted by $\mathrm{d}(\alpha, 3) = -1$ and $\mathrm{d}(\alpha, 6) = 2$, respectively.

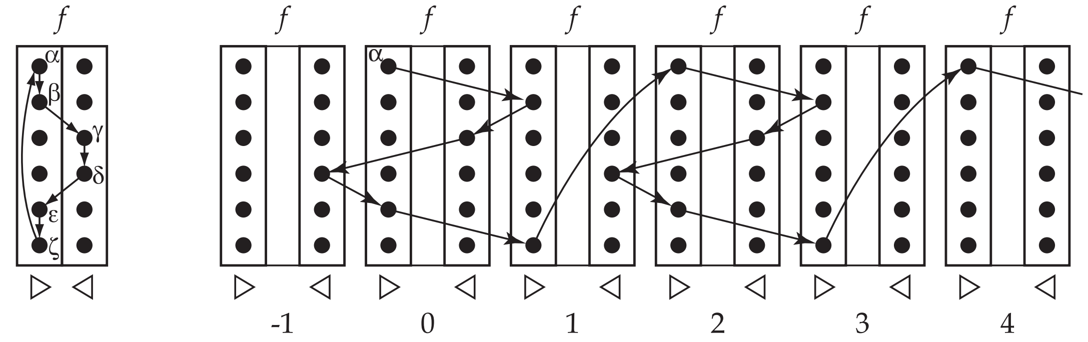

Figure 4: Computation on a bi-infinite string of a's

Definition 3. *For every $\alpha \in N$ and $i \geqslant 0$, such that $f^i(\alpha)$ is defined, let*

$$\mathrm{d}(\alpha, i) = \left|\{j \mid 1 \leqslant j \leqslant i, f^j(\alpha) \sim \alpha\}\right| - \left|\{j \mid 1 \leqslant j \leqslant i, f^j(\alpha) \nsim \alpha\}\right|.$$

In other words, $\mathrm{d}(\alpha, i)$ expresses how far one moves from the original position in the bi-infinite string of fs by means of i steps of the computation represented by the two-way transformation f, where positive numbers mean continuing in the direction of α, while negative numbers mean that the direction was reversed.

Observe that $\mathrm{d}(\alpha, i)$ and $\mathrm{d}(\alpha, i+1)$ always differ exactly by one. Furthermore, for every $\alpha \in N$, the distance travelled after $i + j$ steps can be calculated as follows:

$$\mathrm{d}(\alpha, i+j) = \begin{cases} \mathrm{d}(\alpha, i) + \mathrm{d}(f^i(\alpha), j), & \text{if } f^i(\alpha) \sim \alpha \\ \mathrm{d}(\alpha, i) - \mathrm{d}(f^i(\alpha), j), & \text{if } f^i(\alpha) \nsim \alpha \end{cases} \tag{3}$$

The next observation expresses the fact that when a position with a positive distance m is first visited, it must be entered from the same direction as α (and from the other direction for a position with a negative distance):

Lemma 3. *Let $\alpha \in N$ and $m \in \mathbb{Z}$. If $m > 0$ and $i \geqslant 1$ is the smallest integer with $\mathrm{d}(\alpha, i) = m$, then $f^i(\alpha) \sim \alpha$. Symmetrically, if $m < 0$ and $i \geqslant 1$ is the smallest integer with $\mathrm{d}(\alpha, i) = m$, then $f^i(\alpha) \nsim \alpha$.*

In the following, a two-way transformation $f \in \mathcal{TT}_n$ will be considered as an ordinary transformation of the set N, and it will be investigated how the structure of the corresponding graph determines the behaviour of f as a two-way transformation. This investigation will be based on calculating the values of d for nodes of the graph.

To begin with, the values of d for nodes belonging to f-cycles are given by the structure of their cycles.

Lemma 4. *Let C be an f-cycle of length m and $\alpha, \beta \in C$. Then*

i. $\mathrm{d}(\alpha, m) = |\{\gamma \in C \mid \gamma \sim \alpha\}| - |\{\gamma \in C \mid \gamma \nsim \alpha\}|$.

ii. If $\alpha \sim \beta$ *then* $\mathrm{d}(\beta, m) = \mathrm{d}(\alpha, m)$.

iii. If $\alpha \nsim \beta$ *then* $\mathrm{d}(\beta, m) = -\mathrm{d}(\alpha, m)$.

iv. For every $k \in \mathbb{N}$*:* $\mathrm{d}(\alpha, k \cdot m) = k \cdot \mathrm{d}(\alpha, m)$.

For example, the f-cycle $C = \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta\}$ in Figure 4 is of length 6, and $\mathrm{d}(\alpha, 6) = 4 - 2 = 2$, $\mathrm{d}(\alpha, 12) = 4$ and $\mathrm{d}(\gamma, 6) = -2$.

5.2 $f^{\bullet m}$: computation on a block of m instances of f

For any $m \geqslant 1$ and $\alpha \in N$, the value $f^{\bullet m}(\alpha)$ represents the computation on a block of m instances of f. This computation begins on the first or on the last instance of f in this block, depending on whether $\alpha \in N_{\triangleright}$ or $\alpha \in N_{\triangleleft}$. Consider the case of $\alpha \in N_{\triangleright}$. Then the computation begins on the first instance of f, and at every j-th step the computation proceeds to the neighbouring instance as described above. Unless $f^{\bullet m}(\alpha)$ is undefined, the computation eventually leaves the block to the right or to the left.

The condition of leaving the block can be defined in terms of d as $\mathrm{d}(\alpha, i) \in \{-1, m\}$ for some i. This is formally established in the following lemma, which handles the cases of $\alpha \in N_{\triangleright}$ and $\alpha \in N_{\triangleleft}$ uniformly.

Lemma 5. *For every* $m \in \mathbb{N}$ *and* $\alpha \in N$, $f^{\bullet m}(\alpha) = f^i(\alpha)$, *where* $i \in \mathbb{N}$ *is the smallest number with* $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$, *or, equivalently, with* $\mathrm{d}(\alpha, i) \in \{-1, m\}$. *If such an* i *does not exist, then* $f^{\bullet m}(\alpha)$ *is undefined.*

Furthermore, $\mathrm{d}(\alpha, i) = m$ *if and only if* $f^{\bullet m}(\alpha) \sim \alpha$ *and* $\mathrm{d}(\alpha, i) = -1$ *if and only if* $f^{\bullet m}(\alpha) \nsim \alpha$.

Proof. Assume that $\alpha \in N_{\triangleright}$; for $\alpha \in N_{\triangleleft}$, symmetric arguments can be used due to the associativity of $\bullet$ and the symmetry of its definition. The statement will be proved by induction on m.

Basis: For $m = 1$, the statement turns into $f(\alpha) = f(\alpha)$, which is true.

Induction step: Consider the definition of the operation $\bullet$ in the case of $(f^{\bullet(m-1)} \bullet f)(\alpha)$. In order to verify the statement, it is enough to show the following two claims, since the only remaining possibility is the case when $(f^{\bullet(m-1)} \bullet f)(\alpha)$ is not defined and the number i does not exist.

- The first case of the definition of $(f^{\bullet(m-1)} \bullet f)(\alpha)$ applies if and only if the least i, such that $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$, satisfies $\mathrm{d}(\alpha, i) = m$. In this case, the claim is that $f^{\bullet m}(\alpha) = f^i(\alpha)$ and $f^{\bullet m}(\alpha) \sim \alpha$.

- The second case of the definition of $(f^{\bullet(m-1)} \bullet f)(\alpha)$ applies if and only if the least i, such that $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$, satisfies $\mathrm{d}(\alpha, i) = -1$. The claim in this case is that $f^{\bullet m}(\alpha) = f^i(\alpha)$ too and $f^{\bullet m}(\alpha) \nsim \alpha$.

The former claim shall be verified in the rest of this proof; the argument for the latter claim could be carried out in the same way.

First, assume that in the definition of $(f^{\bullet(m-1)} \bullet f)(\alpha)$ the first case is applicable, that is, there exists $k \in \mathbb{N}$ such that $f \circ (f^{\bullet(m-1)} \circ f)^{\ell}(\alpha) \in N_{\triangleright}$, for $\ell = 0, \ldots, k-1$, $(f^{\bullet(m-1)} \circ f)^{\ell}(\alpha) \in N_{\triangleleft}$, for $\ell = 1, \ldots, k-1$, $(f^{\bullet(m-1)} \circ f)^{k}(\alpha) \in N_{\triangleright}$ and $(f^{\bullet(m-1)} \bullet f)(\alpha) = (f^{\bullet(m-1)} \circ f)^{k}(\alpha)$. By the induction hypothesis, for every $\ell = 1, \ldots, k$,

$$(f^{\bullet(m-1)} \circ f)^{\ell}(\alpha) = f^{i_\ell}(f \circ (f^{\bullet(m-1)} \circ f)^{\ell-1}(\alpha)),$$

where i_ℓ is the least number such that

$$\mathrm{d}(f \circ (f^{\bullet(m-1)} \circ f)^{\ell-1}(\alpha), i_\ell) \notin \{0, \ldots, m-2\}.$$

Additionally, the induction assumption also gives that $\mathrm{d}(f \circ (f^{\bullet(m-1)} \circ f)^{\ell-1}(\alpha), i_\ell) = -1$ for $\ell = 1, \ldots, k-1$ and $\mathrm{d}(f \circ (f^{\bullet(m-1)} \circ f)^{k-1}(\alpha), i_\ell) = m-1$. Denote for all $\ell = 0, \ldots, k$ the sum $1 + i_1 + 1 + \cdots + i_\ell = i_1 + \cdots + i_\ell + \ell$ by s_ℓ. Then (3) gives

$$\mathrm{d}(\alpha, s_{\ell-1} + 1 + j) = \mathrm{d}(\alpha, s_{\ell-1} + 1) + \mathrm{d}(f \circ (f^{\bullet(m-1)} \circ f)^{\ell-1}(\alpha), j),$$

for all $\ell = 1, \ldots, k$ and $j = 0, \ldots, i_\ell$. This in particular means that $\mathrm{d}(\alpha, s_\ell) = 0$ and $\mathrm{d}(\alpha, s_\ell + 1) = 1$, for $\ell = 1, \ldots, k-1$, and $\mathrm{d}(\alpha, s_k) = m$. Therefore $\mathrm{d}(\alpha, s_{\ell-1} + 1 + j) = \mathrm{d}(f \circ (f^{\bullet(m-1)} \circ f)^{\ell-1}(\alpha), j) + 1$, for $\ell = 1, \ldots, k$ and $j = 0, \ldots, i_\ell$, which implies that s_k is the least number such that $\mathrm{d}(\alpha, s_k) \notin \{0, \ldots, m-1\}$. However, since $f^{\bullet m}(\alpha) = (f^{\bullet(m-1)} \circ f)^{k}(\alpha) = f^{s_k}(\alpha)$, the direct implication of the first case is proved.

Conversely, assume that the smallest i, such that $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$, satisfies $\mathrm{d}(\alpha, i) = m$. Let $s_1, \ldots, s_{k-1} \in \mathbb{N}$ be all numbers less than i such that $\mathrm{d}(\alpha, s_\ell) = 0$. Define $i_\ell = s_\ell - s_{\ell-1} - 1$ for $\ell = 1, \ldots k$, where s_0 stands for 0 and s_k stands for i. Then $\mathrm{d}(\alpha, 1 + i_1 + 1 + \cdots + i_\ell) = \mathrm{d}(\alpha, s_\ell) = 0$ and $\mathrm{d}(\alpha, 1 + i_1 + 1 + \cdots + i_\ell + 1) = 1$, for $\ell = 1, \ldots, k-1$, and $\mathrm{d}(\alpha, 1 + i_1 + 1 + \cdots + i_k) = \mathrm{d}(\alpha, i) = m$. This in particular shows that $f^{s_\ell}(\alpha) \in N_{\triangleleft}$ for $\ell = 1, \ldots, k-1$ (since $\mathrm{d}(\alpha, s_\ell - 1)$ cannot be -1) and $f^{s_{\ell-1}+1}(\alpha) \in N_{\triangleright}$ for $\ell = 1, \ldots, k$. Therefore $\mathrm{d}(f^{s_{\ell-1}+1}(\alpha), j) = \mathrm{d}(\alpha, s_{\ell-1} + 1 + j) - 1 < m - 1$ for $j = 0, \ldots, i_\ell - 1$, which proves that i_ℓ is the smallest number such that $\mathrm{d}(f^{s_{\ell-1}+1}(\alpha), i_\ell) \notin \{0, \ldots, m-2\}$. By the induction hypothesis, this implies $f^{\bullet(m-1)}(f^{s_{\ell-1}+1}(\alpha)) = f^{s_\ell}(\alpha)$. Putting these facts together for all ℓ, one obtains $f \circ (f^{\bullet(m-1)} \circ f)^{\ell}(\alpha) = f^{s_\ell+1}(\alpha) \in N_{\triangleright}$, for $\ell = 0, \ldots, k-1$, $(f^{\bullet(m-1)} \circ f)^{\ell}(\alpha) = f^{s_\ell+1}(\alpha) \in N_{\triangleleft}$, for $\ell = 1, \ldots, k-1$, and $(f^{\bullet(m-1)} \circ f)^{k}(\alpha) \in N_{\triangleright}$. This means that in the definition of $(f^{\bullet(m-1)} \bullet f)(\alpha)$ the first case applies. □

5.3 $\tilde{f}$: moving by f until advancing by one position

Consider the example in Figure 4. What the f-cycle of length 6 does is, starting from α, first going one step forward, then two steps back and finally three steps forward. Provided that there is an extra instance of f in position -1, this results in going 2 steps forward. However, if the instance of f in position 0 is the leftmost one, then this computation would not take place.

In order to deal with computations on long blocks of fs, it is useful to assume that there is an unbounded supply of fs on both sides, and consider the computation starting from α that results in advancing by one position (that is, to the right if $\alpha \in N_{\triangleright}$ and to the left if $\alpha \in N_{\triangleleft}$).

Definition 4. *For every $f \in \mathcal{TT}_n$, define a partial transformation $\tilde{f} \in \mathcal{PT}_N$ by the rule $\tilde{f}(\alpha) = f^i(\alpha)$, where $i \in \mathbb{N}$ is the smallest number with* $\mathrm{d}(\alpha, i) = 1$. *If such an i does not exist, $\tilde{f}(\alpha)$ is undefined.*

Returning to Figure 4, $\tilde{f}(\alpha) = f(\alpha) = \beta$, since $\mathrm{d}(\alpha, 1) = 1$. The value of $\tilde{f}(\beta)$ is given by $f^5(\beta) = \alpha$, because $\{\mathrm{d}(\beta, n)\}_{n \geqslant 1} = \{-1, -2, -1, 0, \mathbf{1}, \ldots\}$. For the elements $\gamma, \delta \in N_{\triangleleft}$, $\tilde{f}$ represents going by one step to the left, and accordingly, $\tilde{f}(\gamma) = \delta$ and $\tilde{f}(\delta)$ is undefined.

A few basic properties of $\tilde{f}$ need to be noted. First, since $\tilde{f}(\alpha) = f^i(\alpha)$ for some i, every arc in the graph of $\tilde{f}$ is a shortcut in the graph of f, representing the resultant movement by one step forward:

Remark 1. *If there is an $\tilde{f}$-path from α to β, then there is an f-path from α to β. In particular, if α and β belong to the same $\tilde{f}$-cycle, then they also belong to the same f-cycle.*

Secondly, $\tilde{f}(\alpha) \sim \alpha$, since the last step of the computation defining $\tilde{f}(\alpha)$ is a move in the same direction as α.

Lemma 6. *Let $\alpha \in N$ be such that $\tilde{f}(\alpha)$ is defined. Then there exists such $i \geqslant 1$, that $\alpha \sim f^{i-1}(\alpha) \sim f^i(\alpha) = \tilde{f}(\alpha)$.*

Proof. According to the definition of $\tilde{f}$, let i be the smallest number with $\mathrm{d}(\alpha, i) = 1$. Then the last element of the sequence $\{\mathrm{d}(\alpha, j)\}_{j=0}^{i}$ is 1, and hence the second last element must be $\mathrm{d}(\alpha, i-1) = 0$ (if it were 2, then there would exist $i' < i$ satisfying $\mathrm{d}(\alpha, i') = 1$, which would contradict the choice of i). Then $\mathrm{d}(\alpha, i) = \mathrm{d}(\alpha, i-1) + 1$ implies that $f^i(\alpha) \sim \alpha$.

If $i = 1$, then $f^{i-1}(\alpha) = \alpha \sim \alpha$, and if $i \geqslant 2$, then, continuing the above argument, the third last element of the sequence $\{\mathrm{d}(\alpha, j)\}_{j=0}^{i}$ must be -1. Therefore, $\mathrm{d}(\alpha, i-1) = \mathrm{d}(\alpha, i-2) + 1$ and $f^{i-1}(\alpha) \sim \alpha$. □

The partial transformation $\tilde{f}$ corresponds to advancing by one position. The next lemma establishes that the m-th power of $\tilde{f}$ (that is, the composition of m instances of $\tilde{f}$) represents advancing by m positions.

Lemma 7. *For every $m \in \mathbb{N}$ and $\alpha \in N$ it holds that $\tilde{f}^m(\alpha) = f^i(\alpha)$, where $i \in \mathbb{N}$ is the smallest number with* $\mathrm{d}(\alpha, i) = m$.

Proof. It has to be proved by induction on m that $\tilde{f}^m(\alpha)$ is defined if and only if there exists $i \in \mathbb{N}$ with $\mathrm{d}(\alpha, i) = m$, and if this is the case, then $\tilde{f}^m(\alpha) = f^i(\alpha)$, where $i \in \mathbb{N}$ is the smallest number having this property.

Basis $m = 1$: immediate by the definition of $\tilde{f}$.

Induction step: First assume that $\tilde{f}^m(\alpha)$ is defined. Then $\tilde{f}^{m-1}(\alpha)$ is defined as well and by the induction hypothesis $\tilde{f}^{m-1}(\alpha) = f^i(\alpha)$, where i is the smallest number with $\mathrm{d}(\alpha, i) = m - 1$. Let $\beta = \tilde{f}^{m-1}(\alpha)$. Since $\tilde{f}(\beta)$ is defined, it is equal to $f^j(\beta)$, where j is the smallest number with $\mathrm{d}(\beta, j) = 1$. This gives $\tilde{f}^m(\alpha) = f^{i+j}(\alpha)$.

Since $f^i(\alpha) \sim \alpha$ due to Lemma 3, by (3),

$$\mathrm{d}(\alpha, i + j) = \mathrm{d}(\alpha, i) + \mathrm{d}(f^i(\alpha), j) = (m - 1) + 1 = m.$$

Suppose $i+j$ is not the smallest such number. If $\mathrm{d}(\alpha, k) = m$ for some $k < i$, then i is not the smallest number with $\mathrm{d}(\alpha, i) = m - 1$, which contradicts the assumption. So let $\mathrm{d}(\alpha, i + k) = m$ for some $k \leqslant j$. Then

$$m = \mathrm{d}(\alpha, i + k) = \mathrm{d}(\alpha, i) + \mathrm{d}(f^i(\alpha), k),$$

and therefore $\mathrm{d}(f^i(\alpha), k) = 1$, which, by the choice of j, means $k = j$.

Conversely, assume that there exists $i \in \mathbb{N}$ such that $\mathrm{d}(\alpha, i) = m$. Then there also exists $j < i$ such that $\mathrm{d}(\alpha, j) = m-1$. Therefore, by the induction assumption, $\tilde{f}^{m-1}(\alpha) = f^j(\alpha)$ for the smallest j with this property. Since $\tilde{f}^{m-1}(\alpha) \sim \alpha$ by Lemma 3, $\mathrm{d}(\tilde{f}^{m-1}(\alpha), i - j) = \mathrm{d}(\alpha, i) - \mathrm{d}(\alpha, j) = 1$ due to (3). Consequently, $\tilde{f}(\tilde{f}^{m-1}(\alpha))$ is defined. This completes the proof of the induction step. □

5.4 Isomorphism of subsemigroups generated by f and $\tilde{f}$

In order to describe the subsemigroup generated by f in $\mathcal{TT}_n$, it will be proved that it is isomorphic to the subsemigroup generated by $\tilde{f}$ in $\mathcal{PT}_N$. This amounts to showing that for all positive integers m and k the equality $f^{\bullet m} = f^{\bullet k}$ is equivalent to $\tilde{f}^m = \tilde{f}^k$. The direct implication follows immediately from the following lemma, which states that making m steps forward over f is the same as making one step forward over a block of m instances of f.

Lemma 8. *For every $m \geqslant 1$, $\widetilde{f^{\bullet m}} = \tilde{f}^m$.*

Proof. Let d_m be defined for the element $f^{\bullet m} \in \mathcal{TT}_n$ in the same way as d is defined for f. First, the following claim will be proved.

Claim 1. *For every $k \geq 1$ it holds that $(f^{\bullet m})^k(\alpha) = f^i(\alpha)$, where i is the kth smallest positive integer j satisfying one of the conditions*

1. *$f^j(\alpha) \sim \alpha$ and $m \mid \mathrm{d}(\alpha, j)$,*

2. $f^j(\alpha) \nsim \alpha$ and $m \mid (\mathrm{d}(\alpha, j) + 1)$.

Additionally, if $f^i(\alpha) \sim \alpha$, then $\mathrm{d}_m(\alpha, k) = \mathrm{d}(\alpha, i)/m$, and if $f^i(\alpha) \nsim \alpha$, then $\mathrm{d}_m(\alpha, k) = (\mathrm{d}(\alpha, i) - m + 1)/m$.

The proof of the claim proceeds by induction on k. The basis of the induction for $k = 1$ follows directly from Lemma 5. To prove the induction step, assume that the claim holds for k. Two cases have to be distinguished.

First, let $(f^{\bullet m})^k(\alpha) = f^i(\alpha) \sim \alpha$. This means that the first condition holds for i and so $m \mid \mathrm{d}(\alpha, i)$. Because $\mathrm{d}((f^{\bullet m})^k(\alpha), \ell) = \mathrm{d}(\alpha, i + \ell) - \mathrm{d}(\alpha, i)$ for every $\ell \geq 1$ due to 3, this number reaches -1 or m precisely when $\mathrm{d}(\alpha, i + \ell)$ reaches $\mathrm{d}(\alpha, i) - 1$ or $\mathrm{d}(\alpha, i) + m$, respectively. Note that if some ℓ satisfies both $f^{i+\ell}(\alpha) \nsim \alpha$ and $\mathrm{d}(\alpha, i + \ell) = \mathrm{d}(\alpha, i) + m - 1$, then there exists $\ell' < \ell$ such that $f^{i+\ell'}(\alpha) \sim \alpha$ and $\mathrm{d}(\alpha, i + \ell') = \mathrm{d}(\alpha, i) + m$, and so this ℓ is not the smallest one such that $j = i + \ell$ satisfies one of the above conditions. Similarly, if some ℓ satisfies both $f^{i+\ell}(\alpha) \sim \alpha$ and $\mathrm{d}(\alpha, i + \ell) = \mathrm{d}(\alpha, i)$, then there exists $\ell' < \ell$ such that $f^{i+\ell'}(\alpha) \nsim \alpha$ and $\mathrm{d}(\alpha, i + \ell') = \mathrm{d}(\alpha, i) - 1$, and so this ℓ is also not the smallest one such that $j = i + \ell$ satisfies one of the above conditions. This shows that the smallest ℓ such that $\mathrm{d}((f^{\bullet m})^k(\alpha), \ell) \in \{-1, m\}$ is the same as the smallest ℓ such that one of the above conditions holds for $j = i + \ell$. Accordingly, by Lemma 5, $(f^{\bullet m})^{k+1}(\alpha) = f^j(\alpha)$, where j is the $(k + 1)$th smallest j satisfying one of the above conditions. Finally, if $f^j(\alpha) \sim \alpha$, that is, if $\mathrm{d}((f^i(\alpha), \ell) = m$, then one can use the induction hypothesis to calculate $\mathrm{d}_m(\alpha, k + 1) = \mathrm{d}_m(\alpha, k) + 1 = (\mathrm{d}(\alpha, i) + m)/m = \mathrm{d}(\alpha, j)/m$. Analogously, if $f^j(\alpha) \nsim \alpha$, that is, if $\mathrm{d}((f^i(\alpha), \ell) = -1$, then $\mathrm{d}_m(\alpha, k + 1) = \mathrm{d}_m(\alpha, k) - 1 = (\mathrm{d}(\alpha, i) - m)/m = (\mathrm{d}(\alpha, j) - m + 1)/m$.

The second case of $(f^{\bullet m})^k(\alpha) = f^i(\alpha) \nsim \alpha$ can be dealt with in the same way as follows. This means that the first condition holds for i and so $m \mid \mathrm{d}(\alpha, i)$. Because $\mathrm{d}((f^{\bullet m})^k(\alpha), \ell) = \mathrm{d}(\alpha, i + \ell) - \mathrm{d}(\alpha, i)$ for every $\ell \geq 1$ due to 3, this number reaches -1 or m precisely when $\mathrm{d}(\alpha, i + \ell)$ reaches $\mathrm{d}(\alpha, i) - 1$ or $\mathrm{d}(\alpha, i) + m$, respectively. Note that if some ℓ satisfies both $f^{i+\ell}(\alpha) \nsim \alpha$ and $\mathrm{d}(\alpha, i + \ell) = \mathrm{d}(\alpha, i) + m - 1$, then there exists $\ell' < \ell$ such that $f^{i+\ell'}(\alpha) \sim \alpha$ and $\mathrm{d}(\alpha, i + \ell') = \mathrm{d}(\alpha, i) + m$, and so this ℓ is not the smallest one such that $j = i + \ell$ satisfies one of the above conditions. Similarly, if some ℓ satisfies both $f^{i+\ell}(\alpha) \sim \alpha$ and $\mathrm{d}(\alpha, i + \ell) = \mathrm{d}(\alpha, i)$, then there exists $\ell' < \ell$ such that $f^{i+\ell'}(\alpha) \nsim \alpha$ and $\mathrm{d}(\alpha, i + \ell') = \mathrm{d}(\alpha, i) - 1$, and so this ℓ is also not the smallest one such that $j = i + \ell$ satisfies one of the above conditions. This shows that the smallest ℓ such that $\mathrm{d}((f^{\bullet m})^k(\alpha), \ell) \in \{-1, m\}$ is the same as the smallest ℓ such that one of the above conditions holds for $j = i + \ell$. Accordingly, by Lemma 5, $(f^{\bullet m})^{k+1}(\alpha) = f^j(\alpha)$, where j is the $(k + 1)$th smallest j satisfying one of the above conditions. Finally, if $f^j(\alpha) \sim \alpha$, that is, if $\mathrm{d}((f^i(\alpha), \ell) = m$, then one can use the induction hypothesis to calculate $\mathrm{d}_m(\alpha, k + 1) = \mathrm{d}_m(\alpha, k) + 1 = (\mathrm{d}(\alpha, i) + m)/m = \mathrm{d}(\alpha, j)/m$. Analogously, if $f^j(\alpha) \nsim \alpha$, that is, if $\mathrm{d}((f^i(\alpha), \ell) = -1$, then $\mathrm{d}_m(\alpha, k + 1) = \mathrm{d}_m(\alpha, k) - 1 = (\mathrm{d}(\alpha, i) - m)/m = (\mathrm{d}(\alpha, j) - m + 1)/m$.

The claim implies that $\widetilde{f^{\bullet m}}(\alpha) = f^i(\alpha)$, where $i \in \mathbb{N}$ is the smallest number such that $\mathrm{d}(\alpha, i) = m$. Really, $\mathrm{d}_m(\alpha, k)$ can reach 1 only if the corresponding value $\mathrm{d}(\alpha, i)$ is at least m, and if this ever happens, then the first time $\mathrm{d}(\alpha, i)$ equals m, it also holds that $f^i(\alpha) \sim \alpha$, and therefore $f^i(\alpha) = (f^{\bullet m})^k(\alpha)$ for some $k \geq 1$. The statement now follows from Lemma 7. □

The isomorphism of subsemigroups generated by f and $\tilde{f}$ in $\mathcal{TT}_n$ and $\mathcal{PT}_N$, respectively, can now be proved.

Lemma 9. *For every $m, k \in \mathbb{N}$, $f^{\bullet m} = f^{\bullet k}$ if and only if $\tilde{f}^m = \tilde{f}^k$.*

Proof. For the "only if" part, the equality $f^{\bullet m} = f^{\bullet k}$ and Lemma 8 imply that $\tilde{f}^m = \widetilde{f^{\bullet m}} = \widetilde{f^{\bullet k}} = \tilde{f}^k$.

In order to prove the "if" part, assume that $f^{\bullet m}(\alpha) \neq f^{\bullet k}(\alpha)$ with $m < k$, and let us find some element of N where $\tilde{f}^m$ and $\tilde{f}^k$ differ.

Case I. If $f^{\bullet k}(\alpha)$ is not defined, then, by Lemma 5, there is no i with $\mathrm{d}(\alpha, i) = -1$. Then, since $f^{\bullet m}(\alpha)$ is defined, the only option in Lemma 5 is that for the smallest i with $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$ it holds that $\mathrm{d}(\alpha, i) = m$. Accordingly, by Lemma 7, $\tilde{f}^m(\alpha) = f^i(\alpha)$, that is, $\tilde{f}^m(\alpha)$ is defined. At the same time, $\tilde{f}^k(\alpha)$ must be undefined, because if it is defined, then $\tilde{f}^k(\alpha) = \widetilde{f^{\bullet k}}(\alpha)$ by Lemma 8, and thus $f^{\bullet k}(\alpha)$ is defined as well, which contradicts the assumption.

Case II. If $f^{\bullet k}(\alpha) \sim \alpha$, then, by Lemma 5, for the least j with $\mathrm{d}(\alpha, j) \notin \{0, \ldots, k-1\}$ it holds that $\mathrm{d}(\alpha, j) = k$ and $f^{\bullet k}(\alpha) = f^j(\alpha)$. Since $m < k$, there exists $i < j$ with $\mathrm{d}(\alpha, i) = m$, and it is least among numbers with $\mathrm{d}(\alpha, i) \notin \{0, \ldots, m-1\}$, so $f^{\bullet m}(\alpha) = f^i(\alpha)$.

Applying Lemma 7 to both cases yields $\tilde{f}^k(\alpha) = f^j(\alpha)$ and $\tilde{f}^m(\alpha) = f^i(\alpha)$, and therefore

$$\tilde{f}^m(\alpha) = f^i(\alpha) = f^{\bullet m}(\alpha) \neq f^{\bullet k}(\alpha) = f^j(\alpha) = \tilde{f}^k(\alpha).$$

Case III. It remains to deal with the case $f^{\bullet k}(\alpha) \nsim \alpha$. Since $f^{\bullet m}(\alpha) \nsim \alpha$ would imply that $f^{\bullet k}(\alpha)$ and $f^{\bullet m}(\alpha)$ are equal by Lemma 5, it has to be the case that $f^{\bullet m}(\alpha) \sim \alpha$. If $\tilde{f}^m(\alpha) \neq \tilde{f}^k(\alpha)$, then α itself is the required element. So it can be assumed that $\tilde{f}^m(\alpha) = \tilde{f}^k(\alpha)$. Denote by

- i the smallest number with $\mathrm{d}(\alpha, i) = m$;
- j the smallest number with $\mathrm{d}(\alpha, j) = -1$;
- ℓ the smallest number with $\mathrm{d}(\alpha, \ell) = k$.

This means that $f^{\bullet m}(\alpha) = \tilde{f}^m(\alpha) = f^i(\alpha)$, $f^{\bullet k}(\alpha) = f^j(\alpha)$, $\tilde{f}^k(\alpha) = f^\ell(\alpha)$ and $i < j < \ell$. Note that by our assumption $f^i(\alpha) = f^\ell(\alpha)$ and so $f^i(\alpha)$ already lies in an f-cycle whose length divides $\ell - i$. Now it is clear that there exists $p \in \mathbb{N}$ such that $j < p < \ell$, $\mathrm{d}(\alpha, p) = 0$ and $f^p(\alpha) \sim \alpha$. It will be shown by contradiction that $\tilde{f}^m$ and $\tilde{f}^k$ differ on $\beta = f^p(\alpha)$. So assume $\tilde{f}^m(\beta) = \tilde{f}^k(\beta)$ and note that in fact

$$\tilde{f}^k(\beta) = \tilde{f}^k(\alpha) = \tilde{f}^m(\alpha) = f^i(\alpha).$$

Let q be the smallest number such that $\mathrm{d}(\beta, q) = m$ that is, $\tilde{f}^m(\beta) = f^q(\beta)$. Then the number $s = p - i + q$ is a multiple of the length of the f-cycle containing $f^i(\alpha)$, because

$$f^s(f^i(\alpha)) = f^{p+q}(\alpha) = \tilde{f}^m(\beta) = \tilde{f}^k(\beta) = f^i(\alpha).$$

One can also calculate

$$\mathrm{d}(f^i(\alpha), s) = \mathrm{d}(f^i(\alpha), p - i) + \mathrm{d}(\beta, q) = \mathrm{d}(\alpha, p) - \mathrm{d}(\alpha, i) + \mathrm{d}(\beta, q) = 0 - m + m = 0,$$

using (3) twice. By Lemma 4.iv this means that $\mathrm{d}(f^i(\alpha), t) = 0$ for any $t \in \mathbb{N}$ which is a multiple of the length of the cycle. Since the length of the cycle divides $\ell - i$, using (3) one obtains $\mathrm{d}(\alpha, i) = \mathrm{d}(\alpha, \ell)$, contradicting the choice of i and ℓ. Therefore, $\tilde{f}^m(\beta) \neq \tilde{f}^k(\beta)$, and so $\tilde{f}^m$ differs from $\tilde{f}^k$, as required. □

5.5 The cycles and the longest tail in $\tilde{f}$

According to Remark 1, every $\tilde{f}$-cycle is formed by some of the nodes in some f-cycle. The exact number of these nodes is determined in the next lemma:

Lemma 10. *Let $\alpha \in N$ belong to an $\tilde{f}$-cycle consisting of nodes from an f-cycle C. Then the length of the $\tilde{f}$-cycle of α is $|\{\gamma \in C \mid \gamma \sim \alpha\}| - |\{\gamma \in C \mid \gamma \nsim \alpha\}|$. Additionally, if $\beta \in N$ belongs to C and satisfies $\alpha \nsim \beta$, then β belongs to an $\tilde{f}$-tail.*

Proof. Assume that α belongs to an $\tilde{f}$-cycle, that is, $\tilde{f}^k(\alpha) = \alpha$ for some $k \geqslant 1$. Then $\tilde{f}^k(\alpha) = f^i(\alpha)$, where i is the least number with $\mathrm{d}(\alpha, i) = k$. This means that i is a multiple of the length m of the cycle C. Let $i = \ell \cdot m$. According to Lemma 4(i), it is sufficient to prove that the length of the $\tilde{f}$-cycle of α is $\mathrm{d}(\alpha, m)$. Since $\tilde{f}^j(\alpha)$ cannot be equal to α for $j < \mathrm{d}(\alpha, m)$ due to Lemma 7, it is enough to verify $\tilde{f}^{\mathrm{d}(\alpha,m)}(\alpha) = \alpha$. Suppose this is not the case. Then by Lemma 7 the least j satisfying $\mathrm{d}(\alpha, j) = \mathrm{d}(\alpha, m)$ is smaller than m. Using (3), Lemma 4(ii) and Lemma 4(iv) this gives $\mathrm{d}(\alpha, j + (\ell - 1) \cdot m) = \mathrm{d}(\alpha, j) + \mathrm{d}(f^j(\alpha), (\ell - 1) \cdot m) = \mathrm{d}(\alpha, m) + (\ell - 1) \cdot \mathrm{d}(f^j(\alpha), m) = \mathrm{d}(\alpha, m) + (\ell - 1) \cdot \mathrm{d}(\alpha, m) = \mathrm{d}(\alpha, m) + \mathrm{d}(\alpha, (\ell - 1) \cdot m) = \mathrm{d}(\alpha, \ell \cdot m) = \mathrm{d}(\alpha, i) = k$, which contradicts minimality of i.

Because the length of the $\tilde{f}$-cycle is a positive number, the first statement of the lemma in particular gives

$$|\{\gamma \in C \mid \gamma \sim \alpha\}| > |\{\gamma \in C \mid \gamma \nsim \alpha\}|.$$

Therefore β cannot belong to an $\tilde{f}$-cycle too, since that would imply that the converse inequality is true as well. □

As long as an f-cycle has a different number of nodes from $N_{\triangleright}$ and from $N_{\triangleleft}$, it represents eventual advancement in one or the other direction, and there is an $\tilde{f}$-cycle representing this advancement:

Lemma 11. *Let C be an f-cycle, such that $|C \cap N_{\triangleright}| > |C \cap N_{\triangleleft}|$. Then there exists an $\tilde{f}$-cycle of length $|C \cap N_{\triangleright}| - |C \cap N_{\triangleleft}|$ consisting of some of the nodes from $C \cap N_{\triangleright}$. The symmetric statement holds for $|C \cap N_{\triangleright}| < |C \cap N_{\triangleleft}|$.*

Proof. Let m be the length of the cycle C and consider an arbitrary $\alpha \in C \cap N_{\triangleright}$. By Lemma 4.i and Lemma 4.iv it holds that $\mathrm{d}(\alpha, k \cdot m) = k \cdot (|C \cap N_{\triangleright}| - |C \cap N_{\triangleleft}|)$. This means that $\mathrm{d}(\alpha, i)$ reaches arbitrarily large values with increasing i and so, according to Lemma 7, $\tilde{f}^m(\alpha)$ is defined for every $m \geqslant 0$. In particular, for $m \geqslant |C \cap N_{\triangleright}|$ the node $\tilde{f}^m(\alpha)$, which belongs to $C \cap N_{\triangleright}$ by Remark 1, must lie in an $\tilde{f}$-cycle, whose length is $|C \cap N_{\triangleright}| - |C \cap N_{\triangleleft}|$ due to Lemma 10. □

Assume that for some node in $N_{\triangleleft}$ and for another node in $N_{\triangleright}$, their f-paths eventually converge. Then at most one of these nodes continues advancement in the same direction, represented by an $\tilde{f}$-cycle:

Lemma 12. *Let $\alpha, \beta \in N$ be such that $\alpha \nsim \beta$ and there exist $i, j \geqslant 0$ with $f^i(\alpha) = f^j(\beta)$. Then either α or β belongs to an orphan $\tilde{f}$-tail.*

Proof. If the conclusion of the lemma does not hold, then there exist $\gamma, \delta \in N$ belonging to $\tilde{f}$-cycles that can be reached from α and β, respectively, by a sequence of applications of $\tilde{f}$. Then, by Remark 1, both γ and δ belong to the corresponding f-cycles. Since γ and δ can be reached from α and β also by applying f due to Remark 1, the assumption of the lemma guarantees that γ and δ in fact belong to the same f-cycle. Because $\alpha \sim \gamma$ and $\beta \sim \delta$, it holds that $\gamma \nsim \delta$, contradicting Lemma 10. □

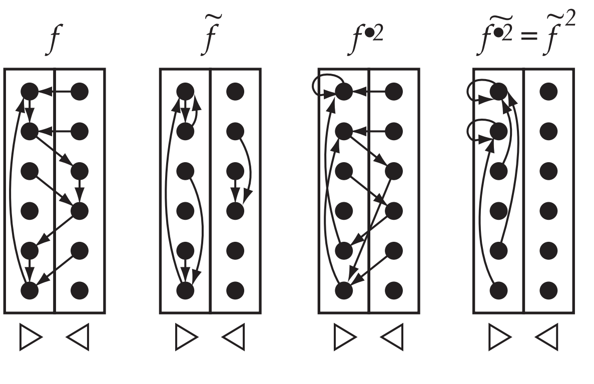

Figure 5: The elements $\tilde{f}$, $f^{\bullet 2}$ and $\widetilde{f^{\bullet 2}} = \tilde{f}^2$ corresponding to f from Figure 4.

Let C be the set of all nodes from N belonging to $\tilde{f}$-cycles, and fix one of the longest $\tilde{f}$-tails. Let T denote this $\tilde{f}$-tail and denote $D = C \cup T$. Denote the restriction of $\tilde{f}$ to D by $\hat{f} \in \mathcal{PT}_D$.

Then the subsemigroups generated by $\tilde{f}$ and $\hat{f}$ have the same index and period by Lemma 1, and hence are isomorphic. Therefore, some information about the index and period of the subsemigroup generated by f can be obtained by finding an upper bound on the size of D with respect to n. In the following, it will be shown that D never contains more than $n+1$ nodes, and that it contains $n+1$ nodes only in one special case. The argument is based on the following simple observation.

Lemma 13. *All nodes in D that do not belong to an f-cycle belong to the same f-tail.*

Proof. Let $\alpha, \beta \in D$ be arbitrary nodes belonging to f-tails. First observe that neither α nor β can belong to an $\tilde{f}$-cycle, since, by Remark 1, this would mean that they belong to an f-cycle as well. Therefore both α and β belong to the unique $\hat{f}$-tail. Consequently, one can reach one of these nodes from the other by several applications of f (again by Remark 1), and so the f-tail of α is the same as the f-tail of β. □

The verification of the inequality $|D| \leqslant n+1$ is rather simple if f is a distinguished transformation, because it can be easily shown that there exists at most one i such that both $(\triangleright, i)$ and $(\triangleleft, i)$ belong to D. Really, assume there is such an $i \in \{1, \ldots, n\}$. Since $f(\triangleright, i) = f(\triangleleft, i)$, at least one of the nodes $(\triangleright, i)$ and $(\triangleleft, i)$ (say the former one) belongs to an f-tail. As they cannot belong to the same f-tail, Lemma 13 ensures that $(\triangleleft, i)$ belongs to an f-cycle. Therefore $(\triangleright, i)$ is the last node of the unique f-tail, which contains some element of D. This implies that such i is uniquely determined. Moreover, one can also see that if such an i exists, then $\hat{f}$ contains a cycle (the node $(\triangleright, i)$ cannot belong to the only $\hat{f}$-tail due to $(\triangleright, i) \nsim (\triangleleft, i)$) and there exists a node in D (namely the last node of the $\hat{f}$-tail of $(\triangleright, i)$) where $\hat{f}$ is not defined.

It turns out that these properties are not specific for distinguished transformations. The following lemma will be used to generalise the above argument to the case of an arbitrary element of $\mathcal{TT}_n$.

Lemma 14. *There exists at most one $\gamma \in N$ such that there exist $\alpha, \delta \in D$ and $i, j \geqslant 1$ satisfying $\alpha \nsim \delta$, $f^j(\alpha) = f^i(\delta) = \gamma$ and $f^{j-1}(\alpha) \neq f^{i-1}(\delta)$.*

Proof. Consider any such γ, α, δ, i and j. The conditions of the lemma imply that γ is a junction node of f where two paths join, which rules out the cases of α and δ belonging to the same f-tail, or both of them belonging to f-cycles. Since $\alpha, \delta \in D$, by Lemma 13, it also cannot be the case that they belong to different f-tails.

Therefore, one of them (let it be α) belongs to an f-tail and δ belongs to an f-cycle. Then, by Lemma 13, α belongs to a certain f-tail that contains all elements of D not belonging to cycles, and accordingly γ must be the node where this f-tail reaches an f-cycle. As such, γ is uniquely determined. □

According to Lemma 14, there exists at most one γ satisfying the conditions. If it exists, denote it by γ_0.

Lemma 15. *It holds that $|N| \geqslant 2\,|D| - 2$. Additionally, if $|N| = 2\,|D| - 2$, then $\hat{f}$ is undefined on some element of D and contains a cycle.*

Proof. Consider the set

$$E = \{\alpha \in N \setminus D \mid f(\alpha) \nsim \alpha \text{ and } f(\alpha) \in \operatorname{Im}(\hat{f})\}.$$

The goal is to find for each $\gamma \in \operatorname{Im}(\hat{f})$ a node $\alpha \in E$ with $f(\alpha) = \gamma$.

Assume $\gamma = \hat{f}(\delta)$, where $\delta \in D$. Since $\tilde{f}(\delta)$ is defined, by Lemma 6 there exists a number $i \geqslant 1$ such that $f^{i-1}(\delta) \sim \tilde{f}(\delta) = f^{i}(\delta) = \gamma$. Denote $\beta = f^{i-1}(\delta)$; then $\beta \sim \gamma$ and $f(\beta) = \gamma$. The condition (1) ensures that there also exists some $\alpha \in N$ such that $\alpha \nsim \gamma$ and $f(\alpha) = \gamma$.

Suppose $\alpha \in D$. Then Lemma 14 is applicable to γ, α, δ, i and $j = 1$; the required condition $\beta = f^{i-1}(\delta) \neq f^{j-1}(\alpha) = \alpha$ holds because $\beta \nsim \alpha$. According to the lemma, this case is only possible for $\gamma = \gamma_0$.

Therefore, for every $\gamma \neq \gamma_0$ from $\operatorname{Im}(\hat{f})$, the corresponding α is not in D. Since $f(\alpha) = \gamma \nsim \alpha$, $\alpha \in E$. For different such γs, the corresponding αs are distinct. There are at least $\bigl|\operatorname{Im}(\hat{f})\bigr| - 1 \geqslant |D| - 2$ such γs different from γ_0, and hence E contains at least $|D| - 2$ elements, which proves $|N| \geqslant 2\,|D| - 2$. Moreover, the equality $|N| = 2\,|D| - 2$ can be satisfied only if $E = |D| - 2$, which could happen only if γ_0 exists. However, in this case Lemma 12 can be applied, which shows that one of α and δ belongs to an $\tilde{f}$-tail containing a node where $\tilde{f}$ is not defined. Because both α and δ belong to D, this $\tilde{f}$-tail must be in fact the only $\hat{f}$-tail. Finally, from $\alpha \nsim \delta$ it follows that the other node cannot belong to this unique $\hat{f}$-tail, and so there exists an $\hat{f}$-cycle. □

5.6 The main theorem and its implications

Now the structure of all monogenic subsemigroups of $\mathcal{TT}_n$ can be described.

Theorem 1. *For every monogenic subsemigroup S of $\mathcal{TT}_n$ there exist $k \geqslant 1$ and numbers $p_1, \dots, p_k \geqslant 1$ and $\ell \geqslant 1$, with $p_1 + \ldots + p_k + \ell \leqslant n+1$, such that S has index ℓ and period $\operatorname{lcm}(p_1, \dots, p_k)$. More precisely, if S is generated by $f \in \mathcal{TT}_n$, then ℓ can be obtained as the length of the longest $\tilde{f}$-tail and numbers $p_1, \dots, p_k$ as the lengths of all $\tilde{f}$-cycles (if there is no cycle in $\tilde{f}$, then one can take $k = 1$ and $p_1 = 1$).*

Conversely, for any $k \geqslant 1$ and arbitrary integers $p_1, \dots, p_k \geqslant 1$ and $\ell \geqslant 1$ satisfying $p_1 + \ldots + p_k + \ell \leqslant n + 1$, the semigroup $\mathcal{TT}_n$ contains a distinguished transformation which generates a subsemigroup with index ℓ and period $\operatorname{lcm}(p_1, \dots, p_k)$.

Proof. The subsemigroup generated by f is isomorphic to the subsemigroup generated by $\tilde{f}$ by Lemma 9, which is in turn isomorphic to the subsemigroup generated by $\hat{f}$ in $\mathcal{PT}_D$. Therefore it follows immediately from Lemma 1 that the index and period of S are ℓ and $\operatorname{lcm}(p_1, \ldots, p_k)$, respectively. It remains to verify that the numbers ℓ, p_1, …, p_k satisfy the required inequalities. First note that ℓ is really at least 1, because otherwise $\tilde{f}$ would be a permutation of N, which is impossible by Lemma 15. If $\tilde{f}$ contains some cycle, then the choice of D guarantees that the sum $p_1 + \ldots + p_k + \ell$ is at most equal to the size of D, which is bounded by $n + 1$ due to Lemma 15. If $\tilde{f}$ contains no cycles, then Lemma 15 shows that D contains at most n nodes, and so $p_1 + \ell = 1 + \ell \leq n + 1$ as well. This verifies the first part of the claim.

In order to verify the converse statement, one can assume, without loss of generality, that $p_1 + \ldots + p_k + \ell = n + 1$. Let g be any permutation of the set $\{\ell, \ldots, n\}$ consisting of cycles of lengths p_1, …, p_k. This allows constructing a distinguished transformation $f \in \mathcal{TT}_n$ as follows: $f(\rhd, i) = f(\lhd, i) = (\rhd, i + 1)$ for $i = 1, \ldots, \ell - 1$ and $f(\rhd, i) = f(\lhd, i) = (\lhd, g(i))$ for $i = \ell, \ldots, n$. In other words, there is a right-going tail on states $\{1, \ldots, \ell\}$ and left-going cycles on states $\{\ell, \ldots, n\}$ according to g. The corresponding partial mapping $\tilde{f}$ coincides with f on $(\rhd, i)$ for $i = 1, \ldots, \ell-1$ and on $(\lhd, i)$ for $i = \ell, \ldots, n$, and it is undefined elsewhere. Since the subsemigroup generated by f is isomorphic to the one generated by $\tilde{f}$ by Lemma 9, Lemma 1 shows that it has index ℓ and period $\operatorname{lcm}(p_1, \ldots, p_k)$, as required. □

Corollary 1. *Let S be a monogenic subsemigroup of $\mathcal{TT}_n$, which has index ℓ and period p. Let $p = p_1 \cdots p_k$, where $p_1, \ldots, p_k$ are powers of distinct primes, be the prime factorization of p. Then, n must be at least $p_1 + \ldots + p_k + \ell - 1$.*

Proof. For S, the theorem states that there are numbers $m \geqslant 1$ and $q_1, \ldots, q_m \geqslant 1$ satisfying $q_1 + \ldots + q_m + \ell \leqslant n + 1$, such that the period of S is $\operatorname{lcm}(q_1, \ldots, q_m)$. Therefore $\operatorname{lcm}(q_1, \ldots, q_m) = p_1 \cdots p_k$, since the period is $p_1 \cdots p_k$ by assumption. This in particular means that each of the prime powers p_i divides at least one of the numbers q_j, and as the primes are pairwise distinct, it implies that $p_1 + \ldots + p_k + \ell \leqslant q_1 + \ldots + q_m + \ell \leqslant n+1$. □

Using Proposition 4, the previous result about monogenic subsemigroups of $\mathcal{TT}_n$ can be applied to languages recognized by 2DFAs.

Corollary 2. *Let $\mathcal{A}$ be an n-state 2DFA over an arbitrary finite alphabet. Then for every monogenic subsemigroup S of the syntactic semigroup of $L(\mathcal{A})$ there exist $k \geqslant 1$ and numbers $p_1, \ldots, p_k \geqslant 1$ and $\ell \geqslant 1$, with $p_1 + \ldots + p_k + \ell \leqslant n + 1$, such that S has index ℓ and period $p = \operatorname{lcm}(p_1, \ldots, p_k)$.*

Corollary 3. *Let L be a regular language over an arbitrary finite alphabet, whose syntactic semigroup contains a monogenic subsemigroup S with index ℓ and period p. Let $p = p_1 \cdots p_k$, where $p_1, \ldots, p_k$ are powers of distinct primes, be the prime factorization of p. Then, every 2DFA recognizing L must have at least $p_1 + \ldots + p_k + \ell - 1$ states.*

Consider computations of a 2DFA $\mathcal{A}$ on inputs ux^iv, in which the behaviour on the infix x^i is described by the two-way transformation $(f_x^{\mathcal{A}})^{\bullet i}$. Due to Proposition 4, the membership of a string ux^iv in the language $L(\mathcal{A})$ depends only on the element $f_{ux^iv}^{\mathcal{A}} = f_v^{\mathcal{A}} \bullet (f_x^{\mathcal{A}})^{\bullet i} \bullet f_u^{\mathcal{A}}$ of $\mathcal{TT}_n$. Therefore, the periodic behaviour of the set $\{i \geqslant 1 \mid ux^iv \in L(\mathcal{A})\}$ depends only on the structure of the subsemigroup generated in $\mathcal{TT}_n$ by $f_x^{\mathcal{A}}$. More precisely, the period of this set divides the period of the subsemigroup, and its index is bounded by the index of the subsemigroup. This leads to the following consequences of Theorem 1 and Corollary 2.

Corollary 4. *Let $\mathcal{A}$ be an n-state 2DFA over an arbitrary finite alphabet Σ, and let $u, v \in \Sigma^*$ and $x \in \Sigma^+$ be any strings. Then, there exist $k \geqslant 1$ and numbers $p_1, \ldots, p_k \geqslant 1$ and $\ell \geqslant 1$, with $p_1 + \ldots + p_k + \ell \leqslant n+1$, such that the set of numbers $\{i \geqslant 1 \mid ux^iv \in L(\mathcal{A})\}$ is periodic from ℓ with period $p = \operatorname{lcm}(p_1, \ldots, p_k)$.*

Corollary 5. *Let L be a regular language over an arbitrary finite alphabet Σ, and let $u, v \in \Sigma^*$ and $x \in \Sigma^+$ be any strings. Let the set of numbers $\{i \geqslant 1 \mid ux^iv \in L(\mathcal{A})\}$ have period p beginning from ℓ. Let $p = p_1 \cdots p_k$, where $p_1, \ldots, p_k$ are powers of distinct primes, be the prime factorization of p. Then, every 2DFA recognizing L must have at least $p_1 + \ldots + p_k + \ell - 1$ states.*

For example, Corollary 5 asserts that every 2DFA for the language $ab\big(((abb)^{15})^* \cup \{(abb)^6\}\big)ba$ requires at least $3+5+7-1$ states.

The last two results are simplified to the case of the unary alphabet as follows (obtained by setting $x = a$ and $u = v = \varepsilon$).

Corollary 6. *Let $\mathcal{A}$ be an n-state 2DFA over $\Sigma = \{a\}$. Then there exist $k \geqslant 1$ and numbers $p_1, \ldots, p_k \geqslant 1$ and $\ell \geqslant 1$, with $p_1 + \ldots + p_k + \ell \leqslant n+1$, such that there exists a 1DFA for $L(\mathcal{A})$ with a tail of length ℓ and period $p = \operatorname{lcm}(p_1, \ldots, p_k)$.*

Corollary 7. *Let $L \subseteq a^*$ be a regular language with the minimal 1DFA with tail ℓ and period p. Let $p = p_1 \cdots p_k$, where $p_1, \ldots, p_k$ are powers of distinct primes, be the prime factorization of p. Then, every 2DFA recognizing L must have at least $p_1 + \ldots + p_k + \max(\ell, 1) - 1$ states.*

6 Transformation to sweeping automata

A 2DFA is called *sweeping* [18] if in every computation its head changes the direction of motion only on the markers. For an arbitrary alphabet, as independently proved by Berman [1] and by Micali [12], the succinctness blowup from general 2DFAs to sweeping 2DFAs is exponential. For a unary alphabet, Mereghetti and Pighizzini [11] established a transformation of an n-state

2NFA to a sweeping 2NFA with $O(n^2)$ states. Regarding the deterministic case, Chrobak [4] mentioned in passing that "it is easy to show that any unary 2DFA can be substituted by an equivalent sweeping 2DFA without increasing the number of its states". This claim was not substantiated, and the best result known in the literature is the $O(n^2)$ bound for unary 2NFAs from the work of Mereghetti and Pighizzini [11]. The framework developed in this paper allows finally settling this question:

Theorem 2. *Let $n \geqslant 1$. Then for every unary deterministic two-way automaton $\mathcal{A}$ with n states, there exists an equivalent sweeping deterministic two-way automaton with $n+1$ states. For $n > 1$, this bound is the best possible.*

In short, the intuition of Chrobak was generally right, though one extra state is needed. The lower bound is witnessed by a 2DFA with acceptance only on the right-end marker (that is, with the standard definition of acceptance), for which every equivalent sweeping 2DFA needs $n+1$ states.

The upper bound is proved by constructing a new sweeping automaton, that simulates the transformation $\tilde{f}$, where f is the behaviour of the original 2DFA on the letter. This is based upon the following correspondence of $\tilde{f}$ to automata.

Lemma 16. *Let $f \in \mathcal{TT}_n$ be the behaviour of a unary 2DFA $\mathcal{A}$ on the letter. If $\mathcal{A}$, having started from a configuration (q_i, k), reaches the $(k+1)$th letter of the input without previously getting to $\vdash$, and $(q_j, k+1)$ is the configuration in which it reaches the $(k+1)$th letter for the first time, then $\tilde{f}((\triangleright, i)) = (\triangleright, j)$.*

Symmetrically, if $\mathcal{A}$, having started from a configuration (q_i, k), reaches the $(k-1)$th letter of the input without previously getting to $\dashv$, and $(q_j, k-1)$ is the configuration in which it reaches $(k-1)$th letter for the first time, then $\tilde{f}((\triangleleft, i)) = (\triangleleft, j)$.

Proof. The assumption is that $\delta^*_{a^k}(q_i, k) = (q_j, k+1)$. By the definition of the behaviour of $\mathcal{A}$, this means that $f^{\mathcal{A}}_{a^k}((\triangleleft, i)) = (\triangleright, j)$. Proposition 1 and Lemma 5 show $f^{\mathcal{A}}_{a^k}((\triangleleft, i)) = f^{\bullet k}((\triangleleft, i)) = f^{\ell}((\triangleleft, i))$, where ℓ is the smallest number such that $\mathrm{d}((\triangleleft, i), \ell) = -1$. Due to (2), $f^{\ell}((\triangleright, i))$ equals $(\triangleright, j)$ as well, and since $\mathrm{d}((\triangleright, i), m) = -\mathrm{d}((\triangleleft, i), m)$ for all $m \geqslant 1$, the number ℓ is also the smallest number such that $\mathrm{d}((\triangleright, i), \ell) = 1$. This proves that $\tilde{f}((\triangleright, i)) = f^{\ell}((\triangleright, i)) = (\triangleright, j)$, as required. □

Applying Lemma 16 several times leads to the following statement:

Lemma 17. *Let $f \in \mathcal{TT}_n$ be the behaviour of a unary 2DFA $\mathcal{A}$ on the letter and let $\ell \geqslant 1$. Let q_j be the state of $\mathcal{A}$ when it reaches position $k+\ell$ for the first time, starting from a configuration (q_i, k). If no end marker was visited before reaching the configuration $(q_j, k+\ell)$, then $\tilde{f}^{\ell}((\triangleright, i)) = (\triangleright, j)$.*

Symmetrically, let q_j be the state of $\mathcal{A}$ when it reaches position $k-\ell$ for the first time, starting from a configuration (q_i, k). If no end marker was visited before reaching the configuration $(q_j, k-\ell)$, then $\tilde{f}^{\ell}((\triangleleft, i)) = (\triangleleft, j)$.

Proof of Theorem 2. The construction of a sweeping 2DFA given below is straightforward in itself: it produces $p_1 + \ldots + p_k + \ell$ states, where $p_1, \ldots, p_k$ are the lengths of the cycles in $\tilde{f}$, and ℓ is the length of its longest tail. The nontrivial part of the argument is the upper bound $n + 1$ on this sum, given in Corollary 6.

Because the index of the language $L(\mathcal{A})$ is at most n by Theorem 1, if the language is finite or co-finite, there exists even a 1DFA with $n + 1$ states recognizing it. So assume that $L(\mathcal{A})$ is neither finite nor co-finite. Let $f \in \mathcal{TT}_n$ be the behaviour of $\mathcal{A}$ on the letter, and let ℓ be the length of the longest $\tilde{f}$-tail. According to Theorem 1, there are at most $n + 1 - \ell$ nodes in C, the set of all nodes of $\hat{f}$-cycles. The new sweeping automaton $\mathcal{B}$ will use the set of states $Q_1 = Q_\triangleright \cup Q_\triangleleft$, where $Q_\triangleright$ is the set of right-moving states and $Q_\triangleleft$ is the set of left-moving states:

$$Q_\triangleright = \{q_1, \ldots, q_\ell\} \cup (C \cap N_\triangleright), \qquad Q_\triangleleft = C \cap N_\triangleleft$$

The automaton $\mathcal{B}$ begins its computation in state q_1 and proceeds by counting the first ℓ symbols using states $q_1, \ldots, q_\ell$:

$$\begin{aligned} \delta'(q_1, \vdash) &= (q_1, +1), \\ \delta'(q_i, a) &= (q_{i+1}, +1) \quad (\text{for } i \in \{1, \ldots, \ell - 1\}). \end{aligned}$$

If the right-end marker is encountered, $\mathcal{B}$ behaves according to the membership of this string in $L(\mathcal{A})$: if $a^{i-1} \in L(\mathcal{A})$, then the transition $\delta'(q_i, \dashv)$ accepts, otherwise it rejects. Because $L(\mathcal{A})$ is neither finite nor cofinite, the automaton $\mathcal{A}$ cannot decide whether to accept a string longer than $a^{\ell-1}$ without ever going beyond the prefix $\vdash a^\ell$. Therefore, $f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, 1)$ is defined as $(\triangleright, k)$, for some $k \in \{1, \ldots, n\}$, and by Lemma 17 the node $(\triangleright, k)$ belongs to an $\hat{f}$-cycle. The sweeping automaton $\mathcal{B}$ has states corresponding to all nodes in $\hat{f}$-cycles, and at this point it enters such a state $(\triangleright, k)$, and then continues counting a's modulo this $\hat{f}$-cycle, thus indirectly simulating a more complicated cyclic behaviour of $\mathcal{A}$:

$$\begin{aligned} \delta'(q_\ell, a) &= (f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, 1), +1), \\ \delta'((\triangleright, i), a) &= (\hat{f}(\triangleright, i), +1). \end{aligned}$$

Eventually, $\mathcal{B}$ reaches the right-end marker. The transition $\delta'((\triangleright, i), \dashv)$ is defined according to the behaviour of $\mathcal{A}$ on the suffix $a^\ell \dashv$, starting from $\dashv$ in state q_i. Three cases will be distinguished. Assume $\mathcal{A}$ eventually passes through a^ℓ and leaves the string to the left in state k, that is, $f^{\mathcal{A}}_{a^\ell \dashv}(\triangleleft, i)$ is of the form $(\triangleleft, k)$, for some $k \in \{1, \ldots, n\}$. Then Lemma 17 guarantees that $(\triangleleft, k)$ belongs to an $\hat{f}$-cycle, and if $\mathcal{B}$ enters the same cycle ℓ states earlier, then it will reach the state $(\triangleleft, k)$ after reading the suffix a^ℓ. This is achieved by defining

$$\delta'((\triangleright, i), \dashv) = ((\triangleleft, j), -1),$$

where $(\triangleleft, j)$ is the (uniquely defined) element of the same $\hat{f}$-cycle as $(\triangleleft, k)$, so that $\hat{f}^\ell(\triangleleft, j) = (\triangleleft, k)$. The second case is when $\mathcal{A}$ accepts at the right-end marker, that is, if $f^{\mathcal{A}}_{a^\ell \dashv}(\triangleleft, i)$ is of the form $(\triangleright, k)$: then the state $((\triangleright, i), \dashv)$ of $\mathcal{B}$ is defined to be accepting. Finally, if $f^{\mathcal{A}}_{a^\ell \dashv}(\triangleleft, i)$ is undefined, then $\delta'((\triangleright, i), \dashv)$ is undefined as well.

The right-to-left motion of the sweeping automaton $\mathcal{B}$ is done in states from $Q_\triangleleft$, using the transitions

$$\delta'((\triangleleft, i), a) = (\hat{f}(\triangleleft, i), -1).$$

Eventually, $\mathcal{B}$ returns to the left-end marker, where its behaviour is defined symmetrically to the case of the right-end marker. If $f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, i)$ is of the form $(\triangleright, k)$, then define

$$\delta'((\triangleleft, i), \vdash) = ((\triangleright, j), +1)$$

where $\hat{f}^\ell(\triangleright, j) = f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, i)$; if $f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, i)$ is of the form $(\triangleleft, k)$, then $((\triangleleft, i), \vdash)$ is accepting; and if $f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, i)$ is undefined, $\delta'((\triangleleft, i), \vdash)$ is undefined as well.

Evidently, the new automaton $\mathcal{B}$ accepts the same words of length at most $\ell - 1$ as $\mathcal{A}$. Let $w \in a^*$ be a word of length at least ℓ. When the automaton $\mathcal{A}$ reaches position $\ell + 2$ for the first time, it is in a state q_k with $f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, 1) = (\triangleright, k)$. On the other hand, the computation of $\mathcal{B}$ on the same input w begins by going through the configurations $(q_1, 1)$, $(q_1, 2)$, $(q_2, 3)$, $\ldots$, $(q_\ell, \ell + 1)$ and reaches $(f^{\mathcal{A}}_{\vdash a^\ell}(\triangleright, 1), \ell + 2)$. It will be proved by induction that in this way the whole computation of $\mathcal{A}$ is simulated by the computation of $\mathcal{B}$: If the original automaton gets from the left-end marker to a configuration (q_k, i), with $i \geq \ell + 2$ without passing through the right-end marker, then the new automaton gets to the configuration $((\triangleright, k), i)$; if the original automaton gets from the right-end marker to a configuration (q_k, i), with $i \leq |w| - \ell + 1$ without passing through the left-end marker, then the new automaton gets to the configuration $((\triangleleft, k), i)$; if the original automaton stops, then the new one stops as well, with the same output.

First assume that $\mathcal{A}$ is in the configuration (q_k, i), with $i \geq \ell + 2$, and $\mathcal{B}$ is in the configuration $((\triangleright, k), i)$. If $i \leq |w| + 1$ then the automata currently read letter a. Because $(\triangleright, k)$ belongs to an $\tilde{f}$-cycle, by Lemma 12 the node $(\triangleleft, k)$ belongs to an $\tilde{f}$-tail that does not lead to a cycle. Therefore $\tilde{f}^\ell(\triangleleft, k)$ is not defined, and Lemma 17 ensures that $\mathcal{A}$ cannot reach the left-end marker before the right-end marker. Therefore, the automaton $\mathcal{A}$ eventually gets into a configuration $(q_m, i + 1)$, which satisfies $\tilde{f}(\triangleright, k) = (\triangleright, m)$ due to Lemma 16. This shows that $\mathcal{B}$ goes from the configuration $((\triangleright, k), i)$ directly to the configuration $((\triangleright, m), i + 1)$, which was to be proved.

In the case $i = |w| + 2$, both automata are at the right-end marker. If $f^{\mathcal{A}}_{a^\ell \dashv}(\triangleleft, i)$ is either of the form $(\triangleright, k)$ or undefined, then both automata accept or reject w, respectively. So it remains to deal with the situation when $f^{\mathcal{A}}_{a^\ell \dashv}(\triangleleft, i)$ is of the form $(\triangleleft, k)$. Then $\mathcal{A}$ eventually reaches the configuration $(q_k, |w| - l + 1)$. Meanwhile, the new automaton $\mathcal{B}$ goes to $((\triangleleft, j), |w| + 1)$

such that $\hat{f}^\ell(\lhd, j) = (\lhd, k)$, and continues to $(\hat{f}(\lhd, j), |w|), \ldots, (\hat{f}^\ell(\lhd, j), |w| - l + 1) = ((\lhd, k), |w| - l + 1)$, which settles also this case.

The other case of automata currently going from the right-end marker is proved symmetrically.

Aiming to prove that $n + 1$ states are necessary, consider the languages $L_n = a(a^2)^* \cup \{a^{n-2}\}$ for n even and $L_n = (a^2)^* \cup \{a^{n-2}\}$ for n odd. The language L_n is recognized by a two-way automaton with the set of states $Q = \{q_1, \ldots, q_n\}$, which starts its computation by counting modulo 2 using the following transitions:

$$\begin{aligned} \delta(q_1, \vdash) &= (q_1, +1), \\ \delta(q_1, a) &= (q_2, +1), \\ \delta(q_2, a) &= (q_1, +1). \end{aligned}$$

Once the right-end marker is reached, the string is either accepted because of its parity,

$$\begin{aligned} \delta(q_1, \dashv) &= (q_1, +1) \quad \text{(if } n \text{ is odd)}, \\ \delta(q_2, \dashv) &= (q_2, +1) \quad \text{(if } n \text{ is even)}, \end{aligned}$$

or the automaton proceeds back to the left in state q_n:

$$\begin{aligned} \delta(q_2, \dashv) &= (q_n, -1) \quad \text{(if } n \text{ is odd)}, \\ \delta(q_1, \dashv) &= (q_n, -1) \quad \text{(if } n \text{ is even)}, \end{aligned}$$

On its way to the left, the automaton decrements its state

$$\delta(q_i, a) = (q_{i-1}, -1) \quad (\text{for } 3 \leqslant i \leqslant n),$$

and if the input string is of length exactly $n - 2$, the automaton arrives to the left-end marker in state q_2, and immediately turns back:

$$\delta(q_2, \vdash) = (q_2, +1).$$

This time the automaton will pass though the string from left to right "in counter-phase", and will accordingly reach the right-end marker in an accepting state.

If the string is shorter than $n - 2$, the automaton reaches the left-end marker in state q_i with $i \geqslant 3$, for which the transition is undefined.

Finally, if the length of the string is $n - 1$ or greater, then the automaton enters the state q_2 before reaching the left-end marker. Note that for every state q_i visited during the right-to-left movement, the state $q_j \in \{q_1, q_2\}$ in which the automaton first visited this square satisfies $j = i \pmod 2$: this property holds for the last square and q_n and extends inductively for every next step. Therefore, when the automaton enters q_2 from the right, it has

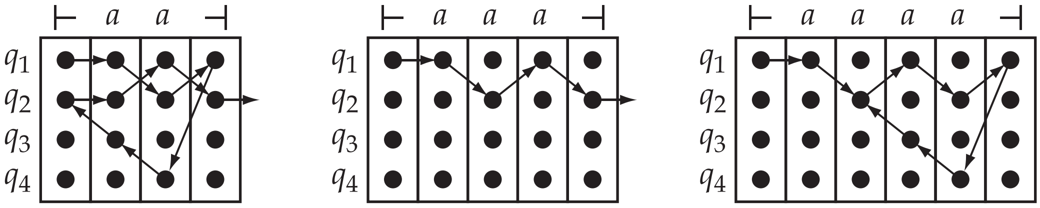

Figure 6: Computation of the 2DFA for L_n with $n = 4$ in Theorem 2.

already been in this state, and thus enters an infinite loop. Note that the behaviour in this case makes the automaton non-sweeping.

In order to show that the language L_n cannot be recognized by an n-state sweeping automaton, note first that L_n has index $n-1$ and period 2. Assume that a sweeping automaton $\mathcal{A} = (\Sigma, Q, q_1, \delta)$ for L_n exists, and let $f \in \mathcal{TT}_n$ be the behaviour of the letter in $\mathcal{A}$. Then, by Theorem 1, the partial function $\hat{f}$ must be defined on $n+1$ nodes and has to consist of a tail of length $n-1$ and a cycle of length 2. Furthermore, the tail may noy lead to a cycle, because in this case both the tail and the cycle have to be in $N_\triangleright$ (or both in $N_\triangleleft$), which requires $(n-1)+2 = n+1$ elements on that side. Hence, either this orphan $\tilde{f}$-tail of length $n-1$ is in $N_\triangleright$ and the $\tilde{f}$-cycle of length 2 is in $N_\triangleleft$, or vice versa. As the following arguments are symmetric, only the former case will be considered.

First, it should be verified that if α and β belong to the $\hat{f}$-tail and $\hat{f}(\alpha) = \beta$, then in fact $f(\alpha) = \beta$. If this is not the case, then $f(\alpha) \nsim \alpha$. Let $i \geq 3$ be the least number such that $\mathrm{d}(\alpha, i) = 1$ and let $\gamma = f^{i-1}(\alpha)$. If $f^{i-2}(\alpha) \sim \alpha$, then $\tilde{f}(f^{i-2}(\alpha)) = \gamma$. If $f^{i-2}(\alpha) \sim \alpha$, then $f^{i-2}(\alpha) = (\triangleleft, j)$ for some $j \leq n$ and $\tilde{f}(\triangleright, j) = \gamma$. Because the $\hat{f}$-tail consists of all nodes of $N_\triangleright$ except for one, in both cases the node γ must belong to this tail. This means that $\hat{f}(\gamma) = \hat{f}(\alpha)$, and since α and γ lie on the same $\hat{f}$-tail, they are actually equal, which contradicts the assumption $f(\alpha) \nsim \alpha$.

Knowing that the $\hat{f}$-tail is in fact an f-tail as well, one can assume, without loss of generality, that $f((\triangleright, j)) = f((\triangleleft, j)) = (\triangleright, j+1)$ for all $j \leq n-2$. This immediately implies that at most one of the nodes $(\triangleleft, j)$, for $j \leq n-2$, belongs to an f-cycle. By Lemma 10 the f-cycle containing the $\hat{f}$-cycle consists either of three nodes from $N_\triangleleft$ and one node from $N_\triangleright$, or simply of two nodes of $N_\triangleleft$. In the former case, these three nodes from $N_\triangleleft$ must be $(\triangleleft, n-2)$, $(\triangleleft, n-1)$ and $(\triangleleft, n)$. However, then it is easy to see that such an f-cycle has to satisfy $f(\triangleright, n-1) = (\triangleleft, n)$ and $f(\triangleleft, n-1) = (\triangleleft, n-2)$, which is not allowed. Therefore, the latter case is true. In this case, the two nodes of the cycle must be $(\triangleleft, n-1)$ and $(\triangleleft, n)$, which means that $f(\triangleright, n-1) = f(\triangleleft, n-1) = (\triangleleft, n)$ and $f(\triangleright, n) = f(\triangleleft, n) = (\triangleleft, n-1)$.

Altogether, there are only two possibilities for the behaviour of the automaton $\mathcal{A}$ on the letter. The first one is the following: $\delta(q_j, a) = (q_{j+1}, +1)$ for $j = 1, \ldots, n-2$, $\delta(q_{n-1}, a) = (q_n, -1)$ and $\delta(q_n, a) = (q_{n-1}, -1)$. Because the automaton is sweeping, $\delta(q_1, \vdash) = (q, +1)$, where q is one of the states $q_1,\ldots,q_{n-2}$. However, then if the input word is of length at least $n-1$, the automaton makes a turn when scanning this word, which is a contradiction.

The second possible behaviour of $\mathcal{A}$ is left-right symmetric to the first one: $\delta(q_j, a) = (q_{j+1}, -1)$ for $j = 1, \ldots, n-2$, $\delta(q_{n-1}, a) = (q_n, +1)$ and $\delta(q_n, a) = (q_{n-1}, +1)$. In this case, $\delta(q_1, \vdash) = (q, +1)$, where q is either q_{n-1} or q_n. This means that the automaton reaches the right-end marker in state q_{n-1} or q_n depending on whether the length of the input is odd or even. In any case, both configurations $(q_{n-1}, \dashv)$ and $(q_n, \dashv)$ are reached for infinitely many input words. If the automaton continues from any of these configurations back to the left, it has to use the states $q_1,\ldots,q_{n-2}$, which means that on any word of length at least $n-1$, it turns inside the input. Therefore, the automaton has to stop in both configurations, and so it cannot recognize the language L_n. This completes the proof. □

7 Transformation to one-way automata

Now consider the question of the number of states in 1DFAs and 1NFAs needed to represent languages recognized by n-state 2DFAs. Chrobak [4] was the first to find out that both tradeoffs are asymptotically equivalent to the function

$$g(n) = \max\{\mathrm{lcm}\{p_1, \ldots, p_k\} \mid p_1 + \cdots + p_k \leqslant n\},$$

known as *Landau's function* and estimated as $g(n) = e^{(1+o(1))\sqrt{n \ln n}}$ [9].

For a 2DFA over an alphabet $\{a\}$, with $f \in \mathcal{TT}_n$ representing the behaviour of a, the numbers $p_1, \ldots, p_k$ in the definition of g correspond, according to Theorem 1, to the cycles in $\tilde{f}$. The length of the tail ℓ in Theorem 1 is actually reflected by the number $n - (p_1 + \cdots + p_k)$. The contribution of ℓ into the size of a 1DFA is taken into account in the following definition of a modified Landau's function:

$$g'(n) = \max\{\mathrm{lcm}\{p_1, \ldots, p_k\} + \ell \mid p_1 + \cdots + p_k + \ell = n\}$$

The difference between $g'(n)$ and $g(n)$ is asymptotically insignificant, and hence $g'(n) = e^{(1+o(1))\sqrt{n \ln n}}$.

Theorem 3. *Let $n \geqslant 1$. Then for every unary two-way automaton $\mathcal{A}$ with n states, where the action of the letter is deterministic, there exists an equivalent complete 1DFA with $g'(n) + 1$ states. For $n \geqslant 3$, this bound is tight already for the transformation of complete 2DFAs with acceptance on both sides to 1NFAs.*

Proof. By Corollary 6, there exists a 1DFA for $L(\mathcal{A})$ with $p + \ell$ states, where $p = \mathrm{lcm}(p_1, \ldots, p_k)$, with $k \geqslant 1$, $p_1, \ldots, p_k \geqslant 1$, $\ell \geqslant 1$ and $p_1 + \ldots + p_k + \ell \leqslant n+1$. Because $\ell \geqslant 1$, it holds that $p + (\ell - 1) \leq g'(n)$, and so $p + \ell \leq g'(n) + 1$.

In order to prove that this bound is tight, let $n \geqslant 3$ and consider $k \geqslant 1$, $\ell \geqslant 1$ and $p_1, \dots, p_k \geqslant 2$ are powers of distinct primes, such that $p_1 + \ldots + p_k + (\ell - 1) = n$ and $g'(n) = p_1 \cdots p_k + (\ell - 1)$. Since $n \geq 3$, one can assume that $p_k \geq 3$. Denote $p = p_1 \cdots p_k$. Consider the language

$$L_n = a^{p+\ell-1}(a^p)^* \cup \{a^i \mid i \equiv \ell \pmod{p_k} \ \&\ i \equiv \ell - 1 \pmod{p_1 \cdots p_{k-1}}\}.$$

First, a 2DFA $\mathcal{A}$ with n states, which recognizes L_n, will be constructed. The states of $\mathcal{A}$ will be denoted (r, s), for $r \in \{1, \dots, k\}$ and $s \in \{0, \dots, p_r - 1\}$, and t, for $t \in \{1, \dots, \ell - 1\}$. The initial state is $(1, 0)$, and the computation begins by checking that the length of the string is $\ell - 1$ modulo $p_1, \dots, p_{k-1}$ and ℓ modulo p_k:

$$\begin{aligned}
\delta((1,0), \vdash) &= ((1,0), +1) \\
\delta((r,s), a) &= ((r, s+1 \bmod p_r), +1) \quad (r \leqslant k,\ r \text{ odd},\ 0 \leqslant s < p_r), \\
\delta((r, \ell - 1 \bmod p_r), \vdash) &= ((r+1, 0), +1) \quad (r < k,\ r \text{ odd}), \\
\delta((r,s), a) &= ((r, s+1 \bmod p_r), -1) \quad (r \leqslant k,\ r \text{ even},\ 0 \leqslant s < p_r), \\
\delta((r, \ell - 1 \bmod p_r), \dashv) &= ((r+1, 0), -1). \quad (r < k,\ r \text{ even}).
\end{aligned}$$

If k is even, then: $\delta((k, \ell \bmod p_k), \vdash)$ is accepting, $\delta((k, \ell - 1 \bmod p_k), \vdash) = (1, +1)$ if $\ell > 1$ and $\delta((k, 0), \vdash) = ((k, 0), +1)$ if $\ell = 1$. If k is even, then: $\delta(t, a) = (t + 1, +1)$ for $t \in \{1, \dots, \ell - 2\}$ and $\delta(\ell - 1, a) = ((k, 0), +1)$. If k is odd, then: $\delta(t, a) = (t + 1, -1)$ for $t \in \{1, \dots, \ell - 2\}$ and $\delta(\ell - 1, a) = ((k, 0), -1)$. If k is odd, then: $\delta((k, \ell \bmod p_k), \dashv)$ is accepting, $\delta((k, \ell - 1 \bmod p_k), \dashv) = (1, -1)$ if $\ell > 1$ and $\delta((k, 0), \dashv) = ((k, 0), -1)$ if $\ell = 1$. The rest is undefined.

If the input word belongs to the second component of L_n, the automaton passes successfully through the states of the form (r, s) and accepts in the state $(k, \ell \bmod p_k)$. By the Chinese remainder theorem, the length of the input word is equal to $\ell - 1$ modulo p if and only if it reaches one of the end markers (depending on the parity of k) in the state $(k, \ell - 1 \bmod p_k)$. Then it uses the states $1, \dots, \ell - 1$ to decide whether the word belongs to the first component of L_n, that is, whether it is of length at least ℓ: If the length is at least ℓ, the automaton turns back and returns to the end marker in the accepting state $(k, \ell \bmod p_k)$. If the length does not exceed $\ell - 2$, then the automaton reaches the other end marker in one of the states $1, \dots, \ell - 1$, where it rejects. Finally, if the length is exactly $\ell - 1$, then the automaton reaches the end marker in the state $(k, 0)$, which is either rejecting or initial, so the automaton rejects either directly or by entering an infinite loop.

With the aim of proving that any 1NFA for the language L_n requires at least $p + \ell = g'(n) + 1$ states, assume that $\mathcal{B}$ is a 1NFA with fewer than $p + \ell$ states, which recognizes L_n. An accepting path of the word $a^{p+\ell-1} \in L_n$ passes through $p + \ell$ states of $\mathcal{B}$, so two of these states must be equal. In other words, there exist $0 \leq r < s \leq p + \ell - 1$ such that the state reached after reading a^r is the same as the state reached after reading a^s. This means that the word $a^{p+\ell-1-s+r}$ belongs to L_n. Therefore $p+\ell-1-s+r \equiv p+\ell \pmod{p}_k$, that is, $s-r \equiv -1 \pmod{p}_k$. In particular, this implies that $s-r$ is not divisible by p, and so the word $a^{p+\ell-1+s-r} \in L_n$ also belongs to the second component of L_n. Hence, it satisfies $p+\ell-1+s-r \equiv p+\ell \pmod{p}_k$, that is, $s-r \equiv 1 \pmod{p}_k$, contradicting the assumption $p_k \geqslant 3$. □

References

[1] P. Berman, "A note on sweeping automata", *ICALP 1980*, LNCS 85, 91–97.

[2] P. Berman, A. Lingas, "On complexity of regular languages in terms of finite automata", Report 304, Institute of Computer Science, Polish Academy of Sciences, Warsaw, 1977.

[3] J.-C. Birget, "Concatenation of inputs in a two-way automaton", *Theoretical Computer Science*, 63:2 (1989), 141–156.

[4] M. Chrobak, "Finite automata and unary languages", *Theoretical Computer Science*, 47 (1986), 149–158. Errata: 302 (2003), 497–498.

[5] V. Geffert, C. Mereghetti, G. Pighizzini, "Converting two-way nondeterministic unary automata into simpler automata", *Theoretical Computer Science*, 295:1–3 (2003), 189–203.

[6] M. Holzer, M. Kutrib, "Descriptional and computational complexity of finite automata", *Language and Automata Theory and Applications* (LATA 2009, Tarragona, Spain, April 2–8, 2009), LNCS 5457, 23–42.

[7] G. Jirásková, A. Okhotin, "On the state complexity of operations on two-way finite automata", *Developments in Language Theory* (DLT 2008, Kyoto, Japan, 16–19 September, 2008), LNCS 5257, 443–454.

[8] C. A. Kapoutsis, "Removing bidirectionality from nondeterministic finite automata", *Mathematical Foundations of Computer Science* (MFCS 2005, Gdańsk, Poland, August 29–September 2, 2005), LNCS 3618, 544–555.

[9] E. Landau, "Über die Maximalordnung der Permutationen gegebenen Grades" (On the maximal order of permutations of a given degree), *Archiv der Mathematik und Physik, Ser. 3*, 5 (1903), 92–103.

[10] C. Mereghetti, G. Pighizzini, "Optimal simulations between unary automata", *SIAM Journal on Computing*, 30:6 (2001), 1976–1992.

[11] C. Mereghetti, G. Pighizzini, “Two-way automata simulations and unary languages”, *Journal of Automata, Languages and Combinatorics*, 5:3 (2000), 287–300.

[12] S. Micali, “Two-way deterministic finite automata are exponentially more succinct than sweeping automata”, *Information Processing Letters*, 12:2 (1981), 103–105.

[13] F. R. Moore, “On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata”, *IEEE Transactions on Computers*, 20 (1971), 1211–1214.

[14] D. Perrin, “Finite Automata”, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science vol. B*, MIT Press, Cambridge (1990), 1–57.

[15] M. O. Rabin, D. Scott, “Finite automata and their decision problems”, *IBM Journal of Research and Development*, 3 (1959), 114–125.

[16] W. J. Sakoda, M. Sipser, “Nondeterminism and the size of two way finite automata”, *STOC 1978*, 275–286.

[17] J. C. Shepherdson, “The reduction of two-way automata to one-way automata”, *IBM Journal of Research and Development*, 3 (1959), 198–200.

[18] M. Sipser, “Lower bounds on the size of sweeping automata”, *STOC 1979*, 360–364.

[19] M. Vardi, “A note on the reduction of two-way automata to one-way automata”, *Information Processing Letters*, 30:5 (1989), 261–264.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978-952-12-2327-3
ISSN 1239-1891