Nondeterminism is Essential in Small 2FAs with Few Reversals

Christos A. Kapoutsis

LIAFA – Université Paris VII

Abstract. On every n-long input, every two-way finite automaton (2FA) can reverse its head $O(n)$ times before halting. A 2FA *with few reversals* is an automaton where this number is only $o(n)$. For every h, we exhibit a language that requires $\Omega(2^h)$ states on every deterministic 2FA with few reversals, but only h states on a nondeterministic 2FA with few reversals.

1 Introduction

A long-standing open question in the theory of computation, posed already in the 70s [11,10], is whether every two-way nondeterministic finite automaton (2NFA) has a deterministic equivalent (2DFA) with at most polynomially more states.

The answer is conjectured to be negative. Indeed, this has been confirmed in several special cases: for automata that are *single-pass* (halting upon reaching an endmarker [11]) or *sweeping* (reversing only on endmarkers [12,9]) or *almost oblivious* (exhibiting $o(n)$ distinct trajectories on n-long inputs [5]) or *moles* (exploring the implicit configuration graph [7]). However, for *unary* automata a non-trivial upper bound is known: the simulating 2DFA need never be more than quasi-polynomially larger [2]. We also know that the final answer, both for general and for unary alphabet, may have implications for the old question whether nondeterminism is essential in space-bounded Turing machines [1,3,8].

Here we confirm the general conjecture in yet another special case: for automata that reverse their input head (anywhere on the tape, but) only $o(n)$ times on every n-long input before halting. These '2FAs *with few reversals*' stand very naturally between sweeping 2FAs, which perform only $O(1)$ reversals and only on the endmarkers, and general 2FAs, which perform $O(n)$ reversals (cf. Lemma 1).

Theorem 1. For every h, there is a language that requires $\Omega(2^h)$ states on every 2DFA *with few reversals* but only h states on a 2NFA *with few reversals*.

Here, the family of witness languages is ONE-WAY LIVENESS [10] (as usual [12,5,7]) and the h-state 2NFAs are actually one-way (so that their 'few' reversals are in fact 'zero'). So, this can be seen as a generalization of the lower bound of [12].

Given Theorem 1, two questions arise. First, *does the theorem really generalize* [12], or can it perhaps follow from it by proving that the gap from few-reversal

* Research funded by a Marie Curie Intra-European Fellowship (PIEF-GA-2009-253368) within the European Union Seventh Framework Programme (FP7/2007-2013).

to sweeping 2DFAs is only polynomial? Second, *does the full conjecture really generalize the theorem*, or can it perhaps follow from it by proving that the gap from general to few-reversal 2DFAs is only polynomial? We answer affirmatively.

Theorem 2. For every h, there is a language that requires $2^{\Omega(h)}$ states on every *sweeping* 2DFA but only $O(h)$ states on a 2DFA *with* 2 *reversals*.

Theorem 3. For every h, there is a language that requires $\Omega(2^h)$ states on every 2DFA *with few reversals* but only $O(h^2)$ states on a *general* 2DFA.

Note that Theorems 1-3 answer Hromkovič's Research Problems 2 and 3 from [4].

We consider the second half of Theorem 1 (upper bound) known, and leave the proofs of Theorems 2 and 3 for longer versions of this report. We thus prove only the first part of Theorem 1 (lower bound). We work as in [12]. After fixing notation (Sect. 2), we define *generic strings* (Sect. 3.1), study their *blocks* (Sect. 3.2), build a family of *hard instances* using such blocks (Sect. 4.1), and apply the linear algebra bound on vectors derived from these instances (Sect. 4.2).

2 Preparation

Let $[n] := \{0, \ldots, n-1\}$. For A a set, $\overline{A}$ is the complement and $\mathrm{id}_A : A \to A$ the identity on A. For $f : A \rightharpoonup B$ a partial function, $X \subseteq A$, and $b \in B$, we define $f[X] := \{f(a) \mid a \in X \;\&\; f(a)\,\text{defined}\}$ and $f^{-1}(b) := \{a \in A \mid f(a) = b\}$. If in addition $g : B \rightharpoonup C$, then the composition $f \circ g : A \rightharpoonup C$ is defined on a iff both $f(a)$ and $g(f(a))$ are defined. If $A = B$, then f^n is the n-fold composition of f with itself, and $f \leq g$ means that $f(a)\,\text{defined} \Longrightarrow g(a)\,\text{defined} \;\&\; g(a) = f(a)$.

Fact 1. *The relation $\leq$ is a partial order on $\mathcal{F}_A := \{f \mid f : A \rightharpoonup A\}$, and id_A is maximal in it. Moreover, $g \leq g' \Longrightarrow f \circ g \leq f \circ g'$ for all f,g,$g' \in \mathcal{F}_A$.*

For z a string, $|z|$ and z_j denote its length and its j-th symbol ($1 \leq j \leq |z|$). A (*promise*) *problem* over Σ is any pair $(L, \tilde{L})$ of disjoint subsets of Σ^*. A machine *solves* $(L, \tilde{L})$ if it accepts all $w \in L$ but no $w \in \tilde{L}$. If $\tilde{L} = \overline{L}$, then L is a *language*.

Liveness. For $h \geq 1$, the alphabet $\Sigma_h := \{\text{all } G \subseteq [h] \times [h]\}$ is all two-column directed graphs with h nodes per column and only rightward arrows (Fig. 1a). An n-long $z \in \Sigma_h^*$ is viewed as an $(n{+}1)$-column graph (without arrow directions, for simplicity); its *connectivity* is $\xi := \{(a, b) \mid \text{there is an } n\text{-arrow path from node } a \text{ of column 0 to node } b \text{ of column } n\}$; if $\xi = \emptyset$, then z is *dead*, otherwise it is *live*. We define $\mathrm{OWL}_h = \text{ONE-WAY LIVENESS}_h := \{z \in \Sigma_h^* \mid z \text{ is live}\}$ [10].

Machines. A *two-way deterministic finite automaton* (2DFA) is any tuple $M = (Q, \Sigma, \delta, q_s, q_a, q_r)$, where Q is a set of *states*, Σ an *alphabet*, q_s,q_a,$q_r \in Q$ are the *start*, *accept*, and *reject* states, and $\delta : Q \times (\Sigma \cup \{\vdash, \dashv\}) \to Q \times \{\mathtt{l}, \mathtt{r}\}$ is the (total) *transition function*, for $\vdash$,$\dashv \notin \Sigma$ two endmarkers and $\mathtt{l}$,$\mathtt{r}$ the two directions. An input $w \in \Sigma^*$ is presented to M endmarked, as $\vdash w \dashv$. Computation starts at q_s and on $\vdash$. In each step, the next state and head move are derived from δ and

the current state and symbol. Endmarkers are never violated, except for $\dashv$ if the next state is q_a or q_r; i.e., $\delta(\cdot,\vdash)$ is always of the form $(\cdot,\mathtt{r})$, and $\delta(\cdot,\dashv)$ is always $(q_a,\mathtt{r})$ or $(q_r,\mathtt{r})$ or of the form $(\cdot,\mathtt{l})$. So, the computation either loops, or falls off $\dashv$ into q_r, or falls off $\dashv$ into q_a. In the last case, we say M *accepts* w.

In general, the *computation of M from state p on the j-th symbol of string z*, denoted $\textsc{comp}_{M,p,j}(z)$, is the longest sequence $c=((q_t,j_t))_{0\le t<m}$ with $0<m\le\infty$, $(q_0,j_0)=(p,j)$, and every next (q_t,j_t) derived from the previous one via δ and z in the natural way. We say (q_t,j_t) is the *t-th point* and m the *length*. If $m=\infty$ then c *loops*; otherwise, $j_{m-1}=0$ or $|z|{+}1$ and c *hits left* or *hits right*, respectively, *into* q_{m-1} (Fig. 1b). We say $c'=((q'_t,j'_t))_{0\le t<m'}$ *parallels* c if it is a 'shifted copy' of it: $m'=m$ and $q'_t=q_t$ & $j'_t=j_t+j_*$ for some j_* and all t.

The L-*computation of M from p on z* is $\textsc{lcomp}_{M,p}(z):=\textsc{comp}_{M,p,1}(z)$ and is called a LR-*traversal*, L-*turn*, or L-*loop*, depending on whether it hits right, hits left, or loops. Similarly, the R-computation $\textsc{rcomp}_{M,p}(z):=\textsc{comp}_{M,p,|z|}(z)$ is a RL-*traversal*, R-*turn*, or R-*loop*. Two L-/R-computations *resemble* each other if they share the same first state, type (L/R-turn/loop, LR/RL-traversal), and last state (if it exists). The (*full*) *computation of M on $w\in\Sigma^*$* is $\textsc{comp}_M(w):=\textsc{lcomp}_{M,q_s}(\vdash w\dashv)$. Hence, M accepts w iff $\textsc{comp}_M(w)$ hits right into q_a.

For $w=uzv$, the *decomposition of $c:=\textsc{comp}_M(w)$ by z* is the unique sequence $c_0,c_1,\ldots$ of computations, called *segments*, derived by splitting c wherever it enters or exits z (for each point (q_t,j_t) produced by crossing the u-z or z-v boundary, replace (q_t,j_t) by two copies of it and split between the copies). Note that every c_i for even i (resp., odd i) is a computation on $\vdash u$ or $v\dashv$ (resp., z), and c halts iff there exists a last segment c_m and m is even and c_m falls off $\dashv$.

We say M is *nondeterministic* (2NFA) if δ maps $Q\times(\Sigma\cup\{\vdash,\dashv\})$ to the *powerset* of $Q\times\{\mathtt{l},\mathtt{r}\}$. Then every $\textsc{comp}_{M,p,j}(z)$ is a *set* of computations, and M accepts w iff some $c\in\textsc{comp}_M(w)$ hits right into q_a.

A *reversal* in a computation c is any point (q_t,j_t) whose predecessor and successor exist and lie on the same side with respect to it (Fig. 1b): $t\ne 0,m{-}1$ and either $j_{t-1},j_{t+1}<j_t$ (*backward* reversal) or $j_t<j_{t-1},j_{t+1}$ (*forward* reversal). We write $r(c)$ for the total number of reversals in c. Note that $0\le r(c)\le\infty$, and $r(c)=\infty$ iff c is looping. For $n\ge 0$, we write $r_M(n)$ for the maximum $r(c)$ over all full computations c of M on n-long inputs.

Lemma 1. *For every s-state* 2DFA *M and every length n, either $r_M(n)=\infty$ or $r_M(n)$ is even and at most $(s-1)(n+2)$.*

3 Building Hard Instances

Hard instances for 2DFAs are built in three stages. We start with *generic strings*, which buy us some basic stability in the machine's behavior. We then use generic strings to build *blocks*, where we draw a set of requirements for how the machine may compute. Finally, in order to force her to meet these requirements, we iterate the blocks into long strings, and ask her to decide correctly there. This general strategy is from [12]. Its instantiation here for 2DFAs improves on [7, §3].

From now on, we fix 2DFA $M{=}(Q,\Sigma,\delta,q_s,q_a,q_r)$ and drop it from subscripts.

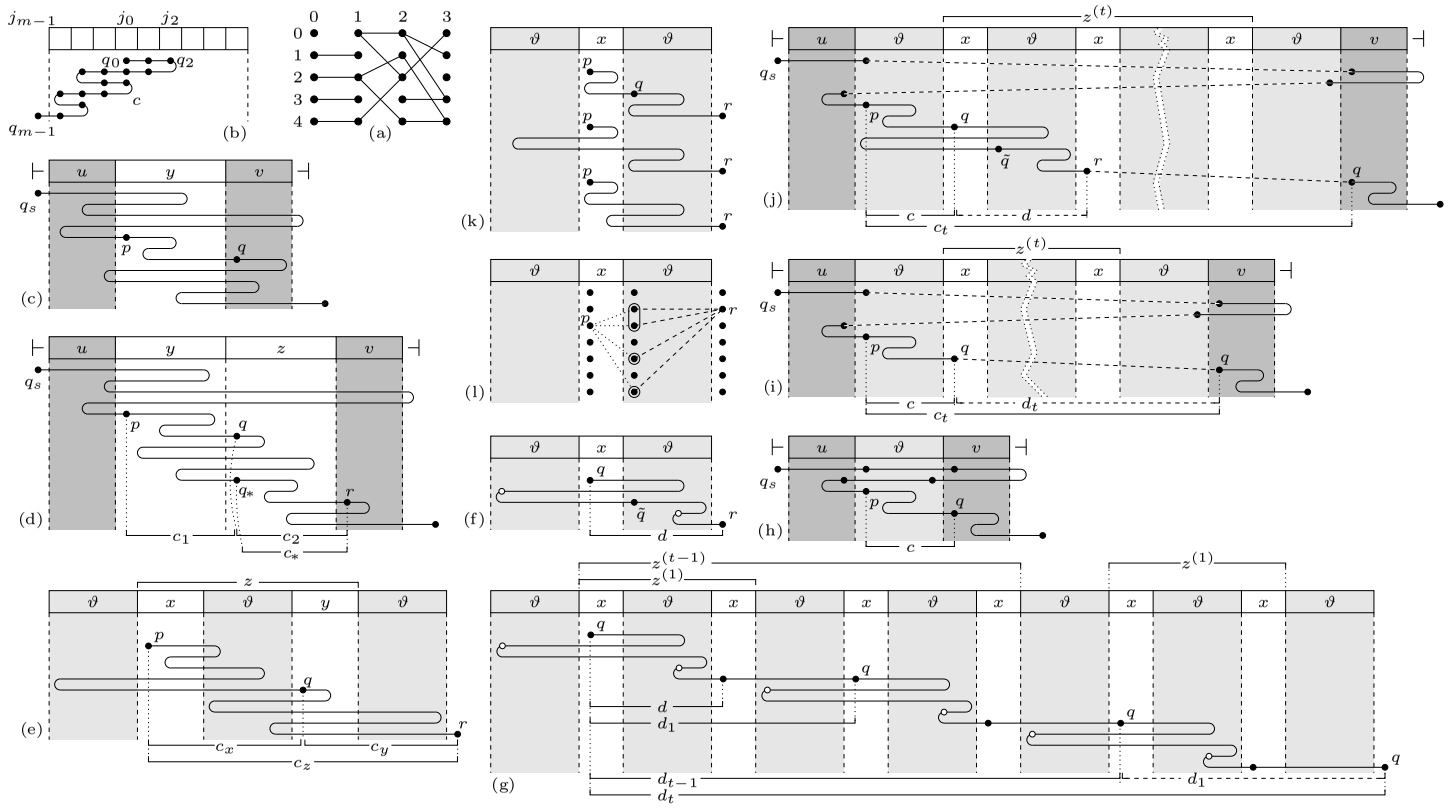


Fig. 1. (a) A string of 3 symbols over Σ_5, drawn as an indexed 4-column undirected graph; here $\xi = \{(2,4),(4,0)\}$. (b) A left-hitting computation c with $m = 15$, $r(c) = 5$; points $2, 8, 12$ are backward reversals, points $6, 11$ are forward reversals. (c) Computing on uyv. (d) Computing on $uyzv$. (e) Computing on overlapping blocks $\vartheta x\vartheta$,$\vartheta y\vartheta$. (f) Witnessing $\alpha_x(q) = r$, with two forward reversals. (g) Induction for proving Lemma 5 (here $l = 2$, $t = 3$). (h) Witnessing $q \in A$. (i) Proving Lemma 5. (j) Proving Fact 6. (k) Three right-hitting $\textsc{comp}_{p,|\vartheta|+1}(\vartheta x\vartheta)$: one simple (top), two non-simple. (l) Understanding S_p: each column is a copy of Q; circles and dashed edges represent the q for which $\textsc{lcomp}_q(\vartheta)$ hits right into r; dotted edges represent the $\delta_{\text{LR}}(p, x, q)$ that are being summed.

3.1 Generic Strings

For each $y \in \Sigma^*$, consider all states that can be produced by LR-traversals of y inside full computations of M (Fig. 1c), called the LR-*outcomes* of y:

$$Q_{\mathrm{LR}}(y) := \big\{ q \in Q \mid \text{there exist } p \text{ and } u,v \text{ such that } \mathrm{LCOMP}_p(y) \text{ appears in } \mathrm{COMP}(uyv) \;\&\; \text{hits right into } q \big\}, \quad (1)$$

where a computation on y 'appears in $\mathrm{COMP}(uyv)$' if it parallels one of the odd-indexed segments in the decomposition of $\mathrm{COMP}(uyv)$ by y.

We now consider any extension yz of y and compare $Q_{\mathrm{LR}}(yz)$ with $Q_{\mathrm{LR}}(y)$ and $Q_{\mathrm{LR}}(z)$. For the first comparison, we define a partial function $\alpha_{y,z} : Q_{\mathrm{LR}}(y) \rightharpoonup Q$ as follows (Fig. 1d): for each $q \in Q_{\mathrm{LR}}(y)$, examine $\mathrm{COMP}_{q,|y|+1}(yz)$; if it hits right into some state r, set $\alpha_{y,z}(q) := r$; if it hits left or loops, leave $\alpha_{y,z}(q)$ undefined.

Fact 2a. *For all $y,z \in \Sigma^*$: $Q_{\mathrm{LR}}(yz) \subseteq \alpha_{y,z}[Q_{\mathrm{LR}}(y)] \cap Q_{\mathrm{LR}}(z)$.*

Proof. Let $r \in Q_{\mathrm{LR}}(yz)$. Then there exist p and u,v such that $c := \mathrm{LCOMP}_p(yz)$ appears in $\mathrm{COMP}(uyzv)$ and hits right into r (Fig. 1d). We know c crosses the y-z boundary at least once. Let q and q_* be the states right after the first crossing and after the last crossing, respectively. The prefix of c up to the first crossing is $c_1 := \mathrm{LCOMP}_p(y)$ and hits right into q, while the remaining suffix is $c_2 := \mathrm{COMP}_{q,|y|+1}(yz)$ and hits right into r. The suffix of c after the last crossing is $c_* = \mathrm{LCOMP}_{q_*}(z)$ and hits right into r. Now, c_1 is a LR-traversal of y that appears in $\mathrm{COMP}(uyzv)$ and produces q, so $q \in Q_{\mathrm{LR}}(y)$. By this and c_2, we know $\alpha_{y,z}(q) = r$. Therefore, $r \in \alpha_{y,z}[Q_{\mathrm{LR}}(y)]$. Moreover, c_* is a LR-traversal of z that appears in $\mathrm{COMP}(uyzv)$ and produces r. Therefore, $r \in Q_{\mathrm{LR}}(z)$. □

Symmetrically, we let the set $Q_{\mathrm{RL}}(y)$ of RL-*outcomes* of y be all states producible by RL-traversals of y inside full computations of M. Then $\beta_{z,y} : Q_{\mathrm{RL}}(y) \rightharpoonup Q$ is introduced so that $\beta_{z,y}(q)$ is r, if $\mathrm{COMP}_{q,|z|}(zy)$ hits left into r, or undefined, if the computation loops or hits right. Then a fact symmetric to Fact 2a holds.

Fact 2b. *For all $y,z \in \Sigma^*$: $Q_{\mathrm{RL}}(zy) \subseteq \beta_{z,y}[Q_{\mathrm{RL}}(y)] \cap Q_{\mathrm{RL}}(z)$.*

By the first inclusion of Fact 2a, we know $|Q_{\mathrm{LR}}(y)| \geq |Q_{\mathrm{LR}}(yz)|$. Similarly, Fact 2b implies $|Q_{\mathrm{RL}}(zy)| \leq |Q_{\mathrm{RL}}(y)|$. Hence, extending a string in either direction can never increase the respective number of outcomes. Thus, sufficiently long extensions will minimize this number. Such extensions are called *generic strings*.

Definition. *Let $T \subseteq \Sigma^*$ be arbitrary. A string $y \in T$ is* LR-generic (for M) over T *if $|Q_{\mathrm{LR}}(y)| = |Q_{\mathrm{LR}}(yz)|$ for all $yz \in T$. It is* RL-generic *if $|Q_{\mathrm{RL}}(zy)| = |Q_{\mathrm{RL}}(y)|$ for all $zy \in T$. It is* generic *if it is both* LR- *and* RL-*generic.*

Lemma 2. *Every $\emptyset \neq T \subseteq \Sigma^*$ admits* LR- *and* RL-*generic strings. Also, if y_{L} is* LR-*generic and y_{R} is* RL-*generic, then every $y_{\mathrm{L}}xy_{\mathrm{R}} \in T$ is generic over T.*

Alternatively, genericity can be characterized via $\alpha_{y,z}$ and $\beta_{z,y}$, as follows.

Lemma 3. *Let $y \in T \subseteq \Sigma^*$. Then y is* LR*-generic over T iff $\alpha_{y,z}$ is total and bijective from $Q_{\text{LR}}(y)$ to $Q_{\text{LR}}(yz)$ for all $yz \in T$. Similarly, y is* RL*-generic over T iff $\beta_{z,y}$ is total and bijective from $Q_{\text{RL}}(y)$ to $Q_{\text{RL}}(zy)$ for all $zy \in T$.*

Proof. We focus on the first equivalence (the second one follows symmetrically) and on the 'only if' direction —the 'if' direction is immediate, since the existence of any total bijection from $Q_{\text{LR}}(y)$ to $Q_{\text{LR}}(yz)$ implies $|Q_{\text{LR}}(y)| = |Q_{\text{LR}}(yz)|$.

Let y be LR-generic over T and pick $yz \in T$. We know $\alpha_{y,z}$ partially maps $Q_{\text{LR}}(y)$ to Q (by definition) and covers $Q_{\text{LR}}(yz)$ (Fact 2a). Namely, each $r \in Q_{\text{LR}}(yz)$ has a distinct $q \in Q_{\text{LR}}(y)$ with $\alpha_{y,z}(q) = r$. So, if there were $q \in Q_{\text{LR}}(y)$ with $\alpha_{y,z}(q)$ undefined or outside $Q_{\text{LR}}(yz)$ or equal to $\alpha_{y,z}(q')$ for another $q' \in Q_{\text{LR}}(y)$, we would have $|Q_{\text{LR}}(y)| > |Q_{\text{LR}}(yz)|$, contrary to y being generic. Hence, $\alpha_{y,z}(q)$ is defined and in $Q_{\text{LR}}(yz)$ and distinct, for all $q \in Q_{\text{LR}}(y)$. Namely, $\alpha_{y,z}$ is a total injection from $Q_{\text{LR}}(y)$ to $Q_{\text{LR}}(yz)$. By Fact 2a, it is also a surjection. □

3.2 Blocks

Fix $\emptyset \neq T \subseteq \Sigma^*$, fix a generic ϑ over T, and let $A := Q_{\text{LR}}(\vartheta)$ and $B := Q_{\text{RL}}(\vartheta)$.

Every string of the form $\vartheta x \vartheta$ is a *block* (*on* ϑ), and x is its *infix*. We say the pair $(\alpha_x, \beta_x) := (\alpha_{\vartheta,x\vartheta}, \beta_{\vartheta x,\vartheta})$ are the *inner behavior* of M on the block.

Recall that $\alpha_x : A \rightharpoonup Q$ and $\beta_x : B \rightharpoonup Q$. In the special case where each function is the identity, the prefix ϑx and the suffix $x\vartheta$ are 'invisible' to M.

Lemma 4. *Suppose $(\alpha_x, \beta_x) = (\text{id}_A, \text{id}_B)$. Pick any u,v and let $c_0, c_1, \ldots$ and $d_0, d_1, \ldots$ be the decompositions of $\text{COMP}(u\vartheta v)$ and $\text{COMP}(u\vartheta x\vartheta v)$ by ϑ and $\vartheta x\vartheta$, respectively. Then c_i parallels d_i for all even i, and resembles d_i for all odd i. Thus, M behaves (accepts, rejects, or loops) identically on $u\vartheta v$ and $u\vartheta x\vartheta v$.*

In blocks of the form $\vartheta(x\vartheta y)\vartheta$, where ϑ is an infix of the infix itself, the inner behavior of M depends on its inner behaviors on the sub-blocks $\vartheta x\vartheta$ and $\vartheta y\vartheta$.

Fact 3. *Let $z = x\vartheta y$. Then $\alpha_x \circ \alpha_y \leq \alpha_z$ and $\beta_y \circ \beta_x \leq \beta_z$. In addition, if α_z is total and injective, then so is α_x; if β_z is total and injective, then so is β_y.*

Proof. To prove $\alpha_x \circ \alpha_y \leq \alpha_z$, let $p \in A$ and assume $(\alpha_x \circ \alpha_y)(p)$ is defined and equal to some $r \in Q$. Then $\alpha_x(p)$ is defined and equal to some $q \in Q$, and $\alpha_y(q)$ is defined and equal to r. By $\alpha_x(p) = q$, we know $c_x := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta)$ hits right into q. (Fig. 1e.) By $\alpha_y(q) = r$, we also know $c_y := \text{COMP}_{q,|\vartheta|+1}(\vartheta y\vartheta)$ hits right into r. Now, concatenating c_x, c_y gives exactly $c_z := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta y\vartheta)$. Hence c_z hits right into r. Therefore $\alpha_z(p)$ is defined and equal to $(\alpha_x \circ \alpha_y)(p)$.

Now suppose α_z is total and injective. *If α_x is not total*, then $\alpha_x(p)$ is undefined for some $p \in A$, namely $c_x := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta)$ hits left or loops. But c_x is a prefix of $c_z := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta y\vartheta)$, so c_z also hits left or loops. Hence $\alpha_z(p)$ is undefined, and α_z is not total—contradiction. *If α_x is not injective*, then $\alpha_x(p) = \alpha_x(p')$ for two distinct $p, p' \in A$, namely $c_x := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta)$ and $c'_x := \text{COMP}_{p',|\vartheta|+1}(\vartheta x\vartheta)$ hit right into the same state. But c_x and c'_x are prefixes of $c_z := \text{COMP}_{p,|\vartheta|+1}(\vartheta x\vartheta y\vartheta)$ and $c'_z := \text{COMP}_{p',|\vartheta|+1}(\vartheta x\vartheta y\vartheta)$, so c_z and c'_z continue identically after the $\vartheta x\vartheta$-$y\vartheta$ boundary, hitting right into the same state. Hence $\alpha_z(p) = \alpha_z(p')$, and α_z is not injective—contradiction. □

We will need a variant of Fact 3 for blocks of the form $\vartheta(x\vartheta x\vartheta\cdots x\vartheta x)\vartheta$, where the infix is multiple ϑ-separated copies of x. Let $x^{(k)} := x(\vartheta x)^{k-1}$ for $k \geq 1$. Note that $\vartheta x^{(k)}\vartheta = \vartheta(x\vartheta)^k = (\vartheta x)^k\vartheta$ and $(x^{(k)})^{(l)} = x^{(lk)}$ for all k, l.

Fact 4. *Let $k \geq 1$. Then $(\alpha_x)^k \leq \alpha_{x^{(k)}}$ and $(\beta_x)^k \leq \beta_{x^{(k)}}$. In addition, if $\alpha_{x^{(k)}}$ is total and injective, then so is α_x; if $\beta_{x^{(k)}}$ is total and injective, then so is β_x.*

Proof. We prove $(\alpha_x)^k \leq \alpha_{x^{(k)}}$ inductively. Case $k = 1$ is trivial. For the inductive step, assume $(\alpha_x)^k \leq \alpha_{x^{(k)}}$. Then $(\alpha_x)^{k+1} = \alpha_x \circ (\alpha_x)^k \leq \alpha_x \circ \alpha_{x^{(k)}}$ (Fact 1) and $\alpha_x \circ \alpha_{x^{(k)}} \leq \alpha_{x^{(k+1)}}$ (Fact 3 for $z = x\vartheta(x^{(k)}) = x\vartheta x(\vartheta x)^{k-1} = x(\vartheta x)^k = x^{(k+1)}$). So, $(\alpha_x)^{k+1} \leq \alpha_{x^{(k+1)}}$ (by transitivity of $\leq$), and we are done. The additional claim follows from that of Fact 3 when $z = x\vartheta(x^{(k-1)}) = x^{(k)}$. □

If any infix $x^{(k)}$ causes the inner behavior of M to just permute the outcomes of ϑ, then longer infixes force the behavior into the special case of Lemma 4.

Fact 5. *If $(\alpha_{x^{(k)}}, \beta_{x^{(k)}})$ permute (A, B), then $(\alpha_{x^{(tlk)}}, \beta_{x^{(tlk)}}) = (\mathrm{id}_A, \mathrm{id}_B)$ for some $l \geq 1$ and all $t \geq 1$.*

Proof. Let $z := x^{(k)}$ and suppose α_z and β_z are permutations of A and B. Pick $l \geq 1$ so that both permutations become identity after l iterations: $(\alpha_z)^l = \mathrm{id}_A$ and $(\beta_z)^l = \mathrm{id}_B$. Then $(\alpha_z)^l \leq \alpha_{z^{(l)}}$ (Fact 4), where $z^{(l)} = (x^{(k)})^{(l)} = x^{(lk)}$; i.e., $\mathrm{id}_A \leq \alpha_{x^{(lk)}}$, so $\alpha_{x^{(lk)}} = \mathrm{id}_A$ (Fact 1). Similarly, $\beta_{x^{(lk)}} = \mathrm{id}_B$. Now, let $t \geq 1$. By Fact 4, $(\alpha_{x^{(lk)}})^t \leq \alpha_{(x^{(lk)})^{(t)}}$. By $(\alpha_{x^{(lk)}})^t = (\mathrm{id}_A)^t = \mathrm{id}_A$ and $(x^{(lk)})^{(t)} = x^{(tlk)}$, we know $\mathrm{id}_A \leq \alpha_{x^{(tlk)}}$, so $\alpha_{x^{(tlk)}} = \mathrm{id}_A$ (Fact 1). Similarly, $\beta_{x^{(tlk)}} = \mathrm{id}_B$. □

We will now state a condition that forces the number of reversals to become more than sublinear. We say (A, B) *use reversals on x* if some $\textsc{comp}_{p,|\vartheta|+1}(\vartheta x\vartheta)$ for $p \in A$ or some $\textsc{comp}_{p,|\vartheta x|}(\vartheta x\vartheta)$ for $p \in B$ contains at least one reversal.

Lemma 5. *If (A, B) use reversals on x and (α_x, β_x) permute (A, B), then it cannot be $r_M(n) = o(n)$.*

Proof. Since (A, B) use reversals, there is a $d := \textsc{comp}_{q,|\vartheta|+1}(\vartheta x\vartheta)$ with $q \in A$ (or a $\textsc{comp}_{q,|\vartheta x|}(\vartheta x\vartheta)$ with $q \in B$, and we work similarly) containing ≥ 1 reversal (Fig. 1f); in fact, d contains ≥ 1 *forward* reversal (because d hits right, since α_x permutes $A \implies \alpha_x(q)$ is defined). Since (α_x, β_x) permute (A, B), there exists $l \geq 1$ such that $(\alpha_{x^{(tl)}}, \beta_{x^{(tl)}}) = (\mathrm{id}_A, \mathrm{id}_B)$ for all $t \geq 1$ (by Fact 5 for $k = 1$).

Let $z := x^{(l)}$. Then $z^{(t)} = x^{(tl)}$ and thus $(\alpha_{z^{(t)}}, \beta_{z^{(t)}}) = (\mathrm{id}_A, \mathrm{id}_B)$, for all t. Using this, we show that each $d_t := \textsc{comp}_{q,|\vartheta|+1}(\vartheta z^{(t)}\vartheta)$ reverses a lot (Fig. 1g).

Claim. For every $t \geq 1$, computation d_t contains $\geq t$ forward reversals.

Proof. By induction. For $t = 1$, $d_1 = \textsc{comp}_{q,|\vartheta|+1}(\vartheta z^{(1)}\vartheta)$. By $z^{(1)} = z = x^{(l)}$ and $l \geq 1$, we know $\vartheta z^{(1)}\vartheta$ has $\vartheta x\vartheta$ as prefix, hence d_1 has d as prefix, and thus contains ≥ 1 forward reversals. For $t > 0$, $d_t = \textsc{comp}_{q,|\vartheta|+1}(\vartheta z^{(t)}\vartheta)$. Since $\vartheta z^{(t)}\vartheta = \vartheta z^{(t-1)}\vartheta z\vartheta$, the prefix of d_t up to the $\vartheta z^{(t-1)}\vartheta$-$z\vartheta$ boundary is d_{t-1}, the state after crossing this boundary is $\alpha_{z^{(t-1)}}(q) = \mathrm{id}_A(q) = q$, and thus the remaining suffix $\textsc{comp}_{q,|\vartheta z^{(t-1)}\vartheta|+1}(\vartheta z^{(t-1)}\vartheta z\vartheta)$ parallels d_1. Hence, d_t contains the $\geq t-1$ forward reversals of d_{t-1} plus the ≥ 1 of d_1, for a total of $\geq t$. ⊡

Since $q \in A = Q_{\text{LR}}(\vartheta)$, there exist p,u,v such that $c := \text{LCOMP}_p(\vartheta)$ appears in $\hat{c} := \text{COMP}(u\vartheta v)$ and hits right into q (Fig. 1h). Consider the family of inputs $w_t := u\vartheta z^{(t)}\vartheta v$ for $t \geq 1$, and the respective computations $\hat{c}_t := \text{COMP}(w_t)$ (Fig. 1i). By Lemma 4 and $(\alpha_{z^{(t)}}, \beta_{z^{(t)}}) = (\text{id}_A, \text{id}_B)$, we know c resembles a segment c_t in the decomposition of $\hat{c}_t$ by $\vartheta z^{(t)}\vartheta$. So, c_t is a L-computation on $\vartheta z^{(t)}\vartheta$ from p. Since ϑ is a prefix of $\vartheta z^{(t)}\vartheta$, the prefix of c_t up to the ϑ-$z^{(t)}\vartheta$ boundary is c, the state after crossing the boundary is q, and the suffix $\text{COMP}_{q,|\vartheta|+1}(\vartheta z^{(t)}\vartheta)$ from then on parallels d_t. So, $\hat{c}_t$ also contains $\geq t$ forward reversals, and thus $\geq 2t$ reversals overall ($\hat{c}_t$ is full, so each forward reversal follows a backward one).

Now, each $\hat{c}_t$ works on input length $n_t := |w_t| = |u\vartheta x^{(tl)}\vartheta v| = |u\vartheta(x\vartheta)^{tl}v| = l|x\vartheta| \cdot t + |u\vartheta v|$. Thus $r_M(n_t) \geq r(\hat{c}_t) \geq 2t = (2/l|x\vartheta|) \cdot n_t - 2|u\vartheta v|/l|x\vartheta|$. So, $r_M(n)$ exceeds a linear function inifinitely often. Hence, $r_M(n) \neq o(n)$. □

Now fix $\tilde{T} \subseteq \overline{T}$. With respect to the problem $(T, \tilde{T})$, an infix x is *positive*, *negative*, or *neutral* if $\vartheta x\vartheta$ is in T, in $\tilde{T}$, or in neither. We will encounter cases which meet the promise that *either some $x^{(k)}$ are positive or all $x^{(k)}$ are negative*. We then say that ϑ,x *respect* $(T, \tilde{T})$; and that they *select* T (resp., $\tilde{T}$), if the promise is met by its left (resp., right) disjunct. If in addition M solves $(T, \tilde{T})$, then we can tell which disjunct is selected using a 'local' criterion for M on $\vartheta x\vartheta$. The next fact assembles this criterion; the next lemma states it more compactly.

Fact 6. *If positive $x^{(k)}$ exist, then (α_x, β_x) permute (A, B). Almost conversely, if M solves $(T, \tilde{T})$ and (α_x, β_x) permute (A, B), then non-negative $x^{(k)}$ exist.*

Proof. [⇒] Suppose $z := x^{(k)}$ is positive for some $k \geq 1$. We shall prove that $\alpha_x : A \rightharpoonup Q$ is a permutation of A (the claim for β_x follows similarly). For this, it is enough to prove two Claims: (1) α_x is total and injective, and (2) $\alpha_x[A] \subseteq A$.

Since z is positive, namely $\vartheta z\vartheta = \vartheta(x\vartheta)^k \in T$, we know $\alpha_z = \alpha_{\vartheta,(x\vartheta)^k}$ is a total bijection from $A = Q_{\text{LR}}(\vartheta)$ to $A' := Q_{\text{LR}}(\vartheta z\vartheta)$ (Lemma 3). But $A' \subseteq A$ (Fact 2a, since $\vartheta z\vartheta$ ends in ϑ) and $|A'| = |A|$ (since α_z is bijective), so $A' = A$. Thus, $\alpha_z = \alpha_{x^{(k)}}$ permutes A. By a symmetric argument, $\beta_z = \beta_{x^{(k)}}$ permutes B.

Since $\alpha_{x^{(k)}}$ is total and injective, Claim 1 is true (Fact 4). For Claim 2, let $r \in \alpha_x[A]$. Then there is $q \in A = Q_{\text{LR}}(\vartheta)$ with $\alpha_x(q) = r$. I.e., there exist p,q and u,v such that $c := \text{LCOMP}_p(\vartheta)$ appears in $\hat{c} := \text{COMP}(u\vartheta v)$ and hits right into q (Fig. 1h), and $d := \text{COMP}_{q,|\vartheta|+1}(\vartheta x\vartheta)$ hits right into r (Fig. 1f). Note that c is an odd-indexed segment in the decomposition of $\hat{c}$ by ϑ. Now pick any $t \geq 1$ with $(\alpha_{z^{(t)}}, \beta_{z^{(t)}}) = (\alpha_{x^{(tk)}}, \beta_{x^{(tk)}}) = (\text{id}_A, \text{id}_B)$ (Fact 5). Lemma 4 says c resembles an odd-indexed segment c_t in the decomposition of $\hat{c}_t := \text{COMP}(u\vartheta z^{(t)}\vartheta v)$ by $\vartheta z^{(t)}\vartheta$ (Fig. 1j). So, c_t is also a L-computation from p, on $\vartheta z^{(t)}\vartheta$. Since $\vartheta x\vartheta$ is a prefix of $\vartheta z^{(t)}\vartheta$, the prefix of c_t up to the first crossing of the right boundary of $\vartheta x\vartheta$ is c followed by a parallel of d. In particular, if $\tilde{q}$ is the state in d after the last crossing of the ϑx-ϑ boundary, then $\tilde{d} := \text{LCOMP}_{\tilde{q}}(\vartheta)$ hits right into r and appears in $\hat{c}_t = \text{COMP}((u\vartheta x)\vartheta(x^{(tk-1)}\vartheta v))$. Hence, $r \in Q_{\text{LR}}(\vartheta) = A$.

[⇐] Suppose M solves $(T, \tilde{T})$ and $(\alpha_x, \beta_x) = (\alpha_{x^{(1)}}, \beta_{x^{(1)}})$ permute (A, B). Pick any $t \geq 1$ with $(\alpha_{x^{(t\cdot 1)}}, \beta_{x^{(t\cdot 1)}}) = (\text{id}_A, \text{id}_B)$ (Fact 5). Pick $k = t \cdot 1$. Then M behaves identically on ϑ and $\vartheta x^{(k)}\vartheta$ (Lemma 4 with empty u,v). Since it accepts $\vartheta \in T$, it also accepts $\vartheta x^{(k)}\vartheta$, thus $\vartheta x^{(k)}\vartheta \notin \tilde{T}$. So, $x^{(k)}$ is positive or neutral. □

Lemma 6. *Suppose M solves $(T, \tilde{T})$ and ϑ,x respect $(T, \tilde{T})$. Then ϑ,x select T iff each outcome of ϑ is hit exactly once by the respective half of the inner behavior:*

$$\big(\forall r \in A\big)\big(|\alpha_x^{-1}(r)| = 1\big) \quad \& \quad \big(\forall r \in B\big)\big(|\beta_x^{-1}(r)| = 1\big)\,. \tag{2}$$

Proof. If ϑ,x select T, then positive $x^{(k)}$ exist, so α_x permutes A (Fact 6) and thus hits every $r \in A$ exactly once; similarly for β_x,B. Conversely, if $\alpha_x : A \rightharpoonup Q$ hits every $r \in A$, then it is total and injective and stays in A (or else its values are not enough to cover A), hence it bijects A into A, i.e., permutes it; similarly for β_x,B. So, nonnegative $x^{(k)}$ exist (Fact 6). So ϑ,x do not select $\tilde{T}$, but T. □

If M uses sublinearly many reversals, then we can simplify (2) by replacing $\alpha_x^{-1}(r), \beta_x^{-1}(r)$ by two simpler sets $\alpha_x^*(r), \beta_x^*(r)$, which we now introduce. Recall that $\alpha_x^{-1}(r)$ is all $p \in A$ for which $\textsc{comp}_{p,|\vartheta|+1}(\vartheta x \vartheta)$ hits right into r. Of course, each p may reach r after arbitrary meanders inside $\vartheta x \vartheta$. Now suppose we demand computations that stay inside $x\vartheta$ and cross the x-ϑ boundary only once; then $\alpha_x^*(r)$ is all $p \in A$ that reach r via such 'simple computations' (Fig. 1k):

$$\alpha_x^*(r) = \alpha_{\vartheta,x\vartheta}^*(r) := \{p \in A \mid (\exists q \in Q)(\textsc{lcomp}_p(x) \text{ hits right into } q \\ \& \ \textsc{lcomp}_q(\vartheta) \text{ hits right into } r)\}\,. \tag{3}$$

Symmetrically, $\beta_x^*(r) = \beta_{\vartheta x,\vartheta}^*(r)$ is all $p \in B$ for which $\textsc{comp}_{p,|\vartheta x|}(\vartheta x \vartheta)$ hits left into r having crossed the ϑx-ϑ and ϑ-$x\vartheta$ boundaries 0 and 1 times respectively.

The simplification of (2) is proved in the next lemma. Before that, the next fact studies the new sets. The boolean functions $\delta_{\text{LR}}(p, x, q)$ and $\delta_{\text{RL}}(q, x, p)$ are 1 iff $\textsc{lcomp}_p(x)$ hits right into q and iff $\textsc{rcomp}_p(x)$ hits left into q, respectively.

Fact 7. *For all $r \in Q$: $\alpha_x^*(r) \subseteq \alpha_x^{-1}(r)$ and $\beta_x^*(r) \subseteq \beta_x^{-1}(r)$. Moreover:*

$$|\alpha_x^*(r)| = \sum_{\substack{p \in A \ \& \ \textsc{lcomp}_q(\vartheta) \\ \text{hits right into } r}} \delta_{\text{LR}}(p, x, q) \qquad |\beta_x^*(r)| = \sum_{\substack{p \in B \ \& \ \textsc{rcomp}_q(\vartheta) \\ \text{hits left into } r}} \delta_{\text{RL}}(q, x, p)\,. \tag{4}$$

Proof. The inclusions are easy. For the left equality, consider any $p \in A$ and the inner sum $S_p := \sum_q \delta_{\text{LR}}(p, x, q)$ over all q for which $\textsc{lcomp}_q(\vartheta)$ hits right into r (Fig. 1l). Since M is deterministic, $S_p \leq 1$. And $S_p = 1$ iff one of these q is the witness required in (3); i.e., $S_p = 1 \iff p \in \alpha_x^*(r)$. So, the number $\sum_{p \in A} S_p$ of $p \in A$ for which $S_p = 1$, is the size of $\alpha_x^*(r)$. Similarly for the other equality. □

Lemma 7. *Suppose M solves $(T, \tilde{T})$ with $r_M(n) = o(n)$, and ϑ,x respect $(T, \tilde{T})$. Then ϑ,x select T iff each outcome of ϑ is hit by exactly one 'simple computation':*

$$\big(\forall r \in A\big)\big(|\alpha_x^*(r)| = 1\big) \quad \& \quad \big(\forall r \in B\big)\big(|\beta_x^*(r)| = 1\big)\,. \tag{5}$$

Proof. Suppose ϑ,x select T. Then (α_x, β_x) permute (A, B) (Fact 6), so (A, B) do not use reversals (Lemma 5 and $r_M(n) = o(n)$). Now pick any $r \in A$. Then $|\alpha_x^*(r)| \leq 1$, because $\alpha_x^*(r) \subseteq \alpha_x^{-1}(r)$ (Fact 7) and $|\alpha_x^{-1}(r)| = 1$ (Lemma 6). And $|\alpha_x^*(r)| \geq 1$, because the r-hitting $\textsc{comp}_{p,|\vartheta|+1}(\vartheta x \vartheta)$ of the unique $p \in \alpha_x^{-1}(r)$ uses no reversals (because (A, B) do not use reversals), thus $p \in \alpha_x^*(r)$. Overall, $|\alpha_x^*(r)| = 1$. Similarly for β_x^*,B. Conversely, if (5) is true, then (2) is true (since $\alpha_x^*(r) \subseteq \alpha_x^{-1}(r)$ and $\beta_x^*(r) \subseteq \beta_x^{-1}(r)$), and thus ϑ,x select T (Lemma 6). □

4 The Proof

Fix $h \geq 1$. Suppose $\Sigma = \Sigma_h$ and M solves OWL_h with $r_M(n) = o(n)$ reversals. We will prove that M needs exponentially many states, namely $|Q| = \Omega(2^h)$.

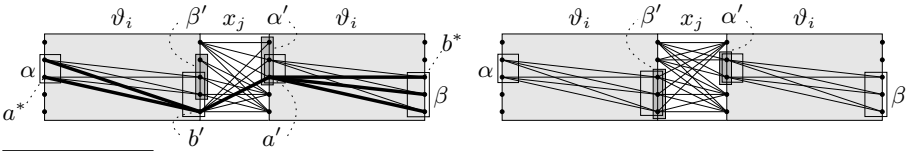
4.1 The Hard Instances

We focus on a family of hard instances of OWL_h similar to that of [6, §3.2]. This is all blocks $\vartheta x \vartheta$ where ϑ and x are drawn from two families $(\vartheta_i)_{i\in\mathcal{I}}$ and $(x_i)_{i\in\mathcal{I}}$ of generic and single-symbol strings, respectively. Here, i ranges over all pairs of non-empty subsets of $[h]$, namely $\mathcal{I} := \{(\alpha,\beta) \mid \emptyset \neq \alpha,\beta \subseteq [h]\}$.[1] These are totally ordered by the rule $(\alpha',\beta') < (\alpha,\beta) \Leftrightarrow_{\text{def}} \langle\alpha'\rangle\langle\beta'\rangle <_{\text{b}} \langle\alpha\rangle\langle\beta\rangle$, where $\langle\cdot\rangle$ is the natural h-bit encoding of subsets of $[h]$ and $<_{\text{b}}$ is the natural order on $2h$-bit positive integers. For each $i = (\alpha,\beta) \in \mathcal{I}$, the string ϑ_i is any fixed generic string over $T_i := \{z \in \Sigma^* \mid z \text{ has connectivity } \alpha \times \beta\}$, and $x_i := \overline{\beta \times \alpha}$ is the 1-long string of all arrows not in $\beta \times \alpha$. We also let $T_\emptyset := \{z \in \Sigma^* \mid z \text{ is dead}\}$.

We picture these blocks on a $|\mathcal{I}| \times |\mathcal{I}|$ matrix. Cell (i,j) hosts block $\vartheta_i x_j \vartheta_i$ and copies of the objects associated with it in Lemma 7: the sets $A_i := Q_{\text{LR}}(\vartheta_i)$, $B_i := Q_{\text{RL}}(\vartheta_i)$ and the functions $\alpha^*_{i,j} := \alpha^*_{\vartheta_i, x_j\vartheta_i}$, $\beta^*_{i,j} := \beta^*_{\vartheta_i x_j, \vartheta_i}$. Crucially, the assumptions of Lemma 7 are satisfied in all cells, and its conclusions follow a simple pattern on and below the diagonal (i.e., when $i \geq j$).

Fact 8. *For all $i,j \in \mathcal{I}$, the assumptions of Lemma 7 are satisfied by M,ϑ_i,x_j for $(T_i, T_\emptyset)$. Furthermore, if $i > j$ then ϑ_i,x_j select T_i; if $i = j$ then ϑ_i,x_j select $T_\emptyset$.*

Proof. Fix any $i = (\alpha,\beta)$ and $j = (\alpha',\beta')$. We first check the assumptions of the lemma. Easily, M solves $(T_i, T_\emptyset)$ (since all of T_i is live and all of $T_\emptyset$ is dead) with $r_M(n) = o(n)$ (by assumption), and ϑ_i is generic for M over T_i (by selection). To show that ϑ_i,x_j respect $(T_i, T_\emptyset)$, we take cases. *If $\vartheta_i x_j \vartheta_i$ is dead*, then all $\vartheta_i(x_j\vartheta_i)^k$ for $k \geq 1$ are dead (since all extensions of a dead string are dead), namely all $(x_j)^{(k)}$ are negative. *If $\vartheta_i x_j \vartheta_i$ is live*, then some path $a^* \rightsquigarrow b^*$ for $a^*,b^* \in [h]$ connects its outer columns (cf. next figure, left side). If b',a' are the visited nodes on the columns of x_j, then the path has the form $a^* \rightsquigarrow b' \to a' \rightsquigarrow b^*$ and ϑ_i has paths $a^* \rightsquigarrow b'$ and $a' \rightsquigarrow b^*$. Hence $(a^*,b'),(a',b^*) \in \xi$, for $\xi = \alpha \times \beta$ the connectivity of ϑ_i. Thus, $b' \in \beta$ and $a' \in \alpha$. Now, for any $a,b \in [h]$, consider the ath leftmost and bth rightmost nodes of $\vartheta_i x_j \vartheta_i$. If $a \notin \alpha \ \vee \ b \notin \beta$, then the two nodes do not connect, since neither can 'see through' ϑ_i; but if $a \in \alpha \ \& \ b \in \beta$, then $(a,b'),(a',b) \in \xi$, so the two nodes connect via a path $a \rightsquigarrow b' \to a' \rightsquigarrow b$. Hence, $\vartheta_i x_j \vartheta_i$ has connectivity ξ, namely $\vartheta_i x_j \vartheta_i \in T_i$, and $(x_j)^{(1)}$ is positive. Overall, ϑ_i,x_j respect $(T_i, T_\emptyset)$. In particular, ϑ_i,x_j *select T_i iff $\vartheta_i x_j \vartheta_i$ is live.*



[1] Here α,β (without subscripts) denote subsets of $[h]$. This causes no confusion with the names (with subscripts) for M's inner behavior, and preserves notational symmetry.

If $i > j$ (cf. left side), then $\langle\alpha'\rangle\langle\beta'\rangle <_{\mathrm{b}} \langle\alpha\rangle\langle\beta\rangle$. Thus $\alpha' \not\supseteq \alpha \ \vee \ \beta' \not\supseteq \beta$ (otherwise $\alpha' \supseteq \alpha$ & $\beta' \supseteq \beta$, thus the 1's of $\langle\alpha'\rangle$ and $\langle\beta'\rangle$ cover all 1's of $\langle\alpha\rangle$ and $\langle\beta\rangle$, hence $\langle\alpha'\rangle\langle\beta'\rangle \geq_{\mathrm{b}} \langle\alpha\rangle\langle\beta\rangle$, a contradiction). Suppose $\beta' \not\supseteq \beta$ (if $\alpha' \not\supseteq \alpha$, apply a similar argument). Pick any $a^* \in \alpha$, $b' \in \beta \setminus \beta'$, $a' \in \alpha$, and $b^* \in \beta$. Then $(a^*, b') \in \xi$ and $(b', a') \in \overline{\beta' \times \alpha'}$ and $(a', b^*) \in \xi$, therefore $\vartheta_i x_j \vartheta_i$ contains the path $a^* \rightsquigarrow b' \to a' \rightsquigarrow b^*$, and is live. Thus, ϑ_i,x_j select $\overline{T_i}$.

If $i = j$ (cf. right side), then x_j has connectivity $\xi' = \overline{\beta \times \alpha}$. If ϑ_i,x_j do not select $T_\emptyset$, then they select T_i, so $\vartheta_i x_j \vartheta_i$ is live. Pick any witnessing path, say of the form $a^* \rightsquigarrow b' \to a' \rightsquigarrow b^*$. Then $(a^*, b') \in \xi$ and $(b', a') \in \xi'$ and $(a', b^*) \in \xi$. Therefore $b' \in \beta$ and $(b', a') \in \overline{\beta \times \alpha}$ and $a' \in \alpha$, a contradiction. □

4.2 The Bound

Now consider the following two families of experiments.

First, fix ϑ_i and $r \in Q$, let x_j range over all possibilities, and observe how the sizes of $\alpha^*_{i,j}(r)$ and $\beta^*_{i,j}(r)$ vary with x_j. The result is two $1 \times |\mathcal{I}|$ vectors, $\mathrm{a}_{i,r} := (|\alpha^*_{i,j}(r)|)_{j\in\mathcal{I}}$ and $\mathrm{b}_{i,r} := (|\beta^*_{i,j}(r)|)_{j\in\mathcal{I}}$. Repeat for all ϑ_i and r, to obtain the two sets of vectors $\mathcal{A} := \{\mathrm{a}_{i,r} \mid i \in \mathcal{I}, r \in Q\}$ and $\mathcal{B} := \{\mathrm{b}_{i,r} \mid i \in \mathcal{I}, r \in Q\}$.

Second, fix p,$q \in Q$, let x_j range over all possibilities, and observe how the bits $\delta_{\mathrm{LR}}(p, x_j, q)$ and $\delta_{\mathrm{RL}}(q, x_j, p)$ vary. The result is two $1 \times |\mathcal{I}|$ binary vectors, $\mathrm{u}_{p,q} := (\delta_{\mathrm{LR}}(p, x_j, q))_{j\in\mathcal{I}}$ and $\mathrm{v}_{q,p} := (\delta_{\mathrm{RL}}(q, x_j, p))_{j\in\mathcal{I}}$. Repeat for all p and q, to obtain the two sets of vectors $\mathcal{U} := \{\mathrm{u}_{p,q} \mid p, q \in Q\}$ and $\mathcal{V} := \{\mathrm{v}_{q,p} \mid p, q \in Q\}$.

Fact 9a. *Every vector in $\mathcal{A} \cup \mathcal{B}$ is a linear combination of vectors from $\mathcal{U} \cup \mathcal{V}$.*

Proof. Fix $i \in \mathcal{I}$ and $r \in Q$. The left equality in Fact 7 implies that for all $j \in \mathcal{I}$:

$$\mathrm{a}_{i,r}(j) = |\alpha^*_{i,j}(r)| = |\alpha^*_{\vartheta_i, x_j \vartheta_i}(r)| = \sum_{\substack{p\in A_i \ \&\ \textsc{lcomp}_q(\vartheta_i) \\ \text{hits right into } r}} \delta_{\mathrm{LR}}(p, x_j, q) = \sum_{\substack{p\in A_i \ \&\ \textsc{lcomp}_q(\vartheta_i) \\ \text{hits right into } r}} \mathrm{u}_{p,q}(j)$$

and thus $\mathrm{a}_{i,r} = \sum \mathrm{u}_{p,q}$ for the specific ranges of p,q. Similarly, $\mathrm{b}_{i,r} = \sum \mathrm{v}_{q,p}$ if the sum ranges over all $p \in B_i$ and all q for which $\textsc{rcomp}_q(\vartheta_i)$ hits left into r. □

Fact 9b. *The set $\mathcal{A} \cup \mathcal{B}$ contains $|\mathcal{I}| - 1$ linearly independent vectors.*

Proof. We will find in $\mathcal{A} \cup \mathcal{B}$ a vector family $(\mathrm{c}_i)_{i\in\mathcal{I}}$ such that $i > j \implies \mathrm{c}_i(j) = 1$ and $i = j \implies \mathrm{c}_i(j) = 0$, for all i,$j \in \mathcal{I}$. This will be enough. Because then the numbers $\mathrm{c}_i(j)$ form a $|\mathcal{I}| \times |\mathcal{I}|$ matrix with 0s on the diagonal and 1s below it, which has rank $|\mathcal{I}| - 1$ (easily), and thus $|\mathcal{I}| - 1$ of the c_i must be independent.

Pick $i \in \mathcal{I}$. Since ϑ_i, x_i select $T_\emptyset$ (Fact 8), there exist $r \in A_i$ or $r \in B_i$ which are *not* hit by exactly 1 simple computation (Lemma 7), respectively $|\alpha^*_{i,i}(r)| \neq 1$ or $|\beta^*_{i,i}(r)| \neq 1$. One of these r must, in fact, be hit by 0 simple computations (otherwise, each r is hit by ≥ 1 value of $\alpha_{i,i}$ or $\beta_{i,i}$ and at least one is hit by ≥ 2 values, for an absurd total of $\geq |A_i| + |B_i| + 1$ values of $\alpha_{i,i}$ and $\beta_{i,i}$). If r_i is this unhit state, then $r_i \in A_i$ & $|\alpha^*_{i,i}(r_i)| = 0$ or $r_i \in B_i$ & $|\beta^*_{i,i}(r_i)| = 0$. In the former case, we let $\mathrm{c}_i := \mathrm{a}_{i,r_i}$; in the latter case, we let $\mathrm{c}_i := \mathrm{b}_{i,r_i}$.

By this definition, clearly $c_i(i) = 0$. Moreover, if $i > j$ then ϑ_i,x_j select T_i (Fact 8) and thus each $r \in A_i$ and $r \in B_i$ is hit by exactly 1 simple computation (Lemma 7). Hence, so is r_i. Therefore, depending on the case in c_i's definition, either $c_i(j) = a_{i,r_i}(j) = |\alpha^*_{i,j}(r_i)| = 1$ or $c_i(j) = b_{i,r_i}(j) = |\beta^*_{i,j}(r_i)| = 1$. □

So, $\mathcal{U} \cup \mathcal{V}$ span a space of dimension $\geq |\mathcal{I}| - 1$. Clearly then $|\mathcal{U} \cup \mathcal{V}| \geq |\mathcal{I}| - 1$, therefore $2|Q|^2 \geq (2^h - 1)^2 - 1$. Hence $|Q| = \Omega(2^h)$, and the proof is complete.

5 Conclusion

We confirmed the Sakoda-Sipser conjecture in the special case of 2FAs performing $o(n)$ reversals, by proving that OWL needs exponentially large 2DFAs of this kind.

The large alphabet of OWL_h is not a problem, as binary witnesses exist, too: Just encode the symbols of Σ_h into h^2-bit strings. Then 2DFAs still need $\Omega(2^h)$ states (same proof, as all reasoning is on cell boundaries), while 2NFAs need only $O(h^2)$. So, our title is valid even if 'small 2FAs' means '2FAs of small description'.

Theorem 1 says that all 2DFAs for OWL_h satisfy $r_M(n) \neq o(n) \vee |Q| = \Omega(2^h)$, but the stronger condition $r_M(n) = \Omega(n) \vee |Q| \geq 2^h$ is probably also true. Another possible direction for further work is to continue with Research Problem 4 of [4] and fully analyze the trade-off between size and number of reversals.

References

1. Piotr Berman and Andrzej Lingas. On complexity of regular languages in terms of finite automata. Report 304, Institute of Computer Science, Polish Academy of Sciences, Warsaw, 1977.
2. Viliam Geffert, Carlo Mereghetti, and Giovanni Pighizzini. Converting two-way nondeterministic unary automata into simpler automata. *Theoretical Computer Science*, 295:189–203, 2003.
3. Viliam Geffert and Giovanni Pighizzini. Two-way unary automata versus logarithmic space. In *Proceedings of DLT*, pages 197–208, 2010.
4. Juraj Hromkovič. Descriptional complexity of finite automata: concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 7(4):519–531, 2002.
5. Juraj Hromkovič and Georg Schnitger. Nondeterminism versus determinism for two-way finite automata: generalizations of Sipser's separation. In *Proceedings of ICALP*, pages 439–451, 2003.
6. Christos Kapoutsis. Small sweeping 2NFAs are not closed under complement. In *Proceedings of ICALP*, pages 144–156, 2006.
7. Christos Kapoutsis. Deterministic moles cannot solve liveness. *Journal of Automata, Languages and Combinatorics*, 12(1-2):215–235, 2007.
8. Christos Kapoutsis. Two-way automata versus logarithmic space. In *Proceedings of CSR*, pages 359–372, 2011.
9. Hing Leung. Tight lower bounds on the size of sweeping automata. *Journal of Computer and System Sciences*, 63(3):384–393, 2001.
10. William J. Sakoda and Michael Sipser. Nondeterminism and the size of two-way finite automata. In *Proceedings of STOC*, pages 275–286, 1978.
11. Joel I. Seiferas. Untitled manuscript, communicated to M. Sipser. October 1973.
12. Michael Sipser. Lower bounds on the size of sweeping automata. *Journal of Computer and System Sciences*, 21(2):195–202, 1980.