State Complexity of Overlap Assembly*

Janusz A. Brzozowski[1], Lila Kari[1], Bai Li[1], and Marek Szykuła[2]

[1] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
brzozo@uwaterloo.ca, lila@uwaterloo.ca, bai.li.2005@gmail.com
[2] Institute of Computer Science, University of Wrocław, Joliot-Curie 15, PL-50-383 Wrocław, Poland
msz@cs.uni.wroc.pl

Abstract. The *state complexity* of a regular language L_m is the number m of states in a minimal deterministic finite automaton (DFA) accepting L_m. The state complexity of a regularity-preserving binary operation on regular languages is defined as the maximal state complexity of the result of the operation where the two operands range over all languages of state complexities $\leq m$ and $\leq n$, respectively. We find a tight upper bound on the state complexity of the binary operation *overlap assembly* on regular languages. This operation was introduced by Csuhaj-Varjú, Petre, and Vaszil to model the process of self-assembly of two linear DNA strands into a longer DNA strand, provided that their ends "overlap". We prove that the state complexity of the overlap assembly of languages L_m and L_n, where $m \geq 2$ and $n \geq 1$, is at most $2(m-1)3^{n-1} + 2^n$. Moreover, for $m \geq 2$ and $n \geq 3$ there exist languages L_m and L_n over an alphabet of size n whose overlap assembly meets the upper bound and this bound cannot be met with smaller alphabets.

Keywords: overlap assembly, regular language, state complexity, tight upper bound

1 Introduction

The *state complexity* of a regular language is the number of states in a minimal deterministic finite automaton (DFA) accepting the language. The state complexity of a regularity-preserving binary operation on regular languages is the maximal state complexity of the result of the operation when the operands range over all languages of state complexities $\leq m$ and $\leq n$; it is a function of m and n. State complexity was introduced by Maslov [14] in 1970, but his short paper was relatively unknown for many years. A more complete study of state complexity was presented by Yu, Zhuang, and Salomaa [15] in 1994. Since the publication of [15], many authors have written on this subject; for an extensive bibliography see the recent surveys [1, 8]. In particular, the state complexities of

* This work was supported by the Natural Sciences and Engineering Research Council of Canada under grants No. OGP0000871 and R2824A01, and by the National Science Centre, Poland, under project number 2014/15/B/ST6/00615.

the so-called basic operations, namely Boolean operations, concatenation, star and reversal in various subclasses of the class of regular languages have been studied [1].

We consider the state complexity of a biologically inspired binary word and language operation called *overlap assembly*. Formally, overlap assembly is a binary operation which, when applied to two input words xy and yz (where y is their nonempty *overlap*), produces the output xyz. As a formal language operation, overlap assembly was introduced by Csuhaj-Varjú, Petre, and Vaszil [4] under the name "self-assembly", and studied by Enaganti, Ibarra, Kari, and Kopecki [6, 7]. A particular case of overlap assembly, called *chop operation*, where the overlap consists of a single letter, was studied in, e.g., [10]. Other similar operations have been studied in the literature, such as the *short concatenation* [3], which uses only the maximum-length (possibly empty) overlap y between operands, the Latin product of words [9] where the overlap consists of only one letter, and the operation $\bigotimes$ which imposes the restriction that the non-overlapping part xz is not empty [12]. Overlap assembly can also be considered as a particular case of semantic shuffle on trajectories with trajectory $0^*\sigma^+1^*$ [5] or as a generalization of the operation $\bigodot_N$ from [5] which imposes the length of the overlap to be $\geq N$.

In this paper we investigate the state complexity of overlap assembly as a binary operation on regular languages. Section 2 describes the biological motivation of overlap assembly. Section 3 introduces our notation, and describes an NFA that accepts the results of overlap assembly of two regular languages, given by their accepting DFAs. In Section 4 we prove that the state complexity of the overlap assembly of languages L_m and L_n, where $m \geq 2$ and $n \geq 1$, is at most $2(m-1)3^{n-1}+2^n$. Moreover, for $m \geq 2$ and $n \geq 3$ there exist languages L_m and L_n over an alphabet of size n whose overlap assembly meets the upper bound and, in addition, this bound cannot be met with smaller alphabets.

2 Overlap Assembly

The bio-operation of overlap assembly was intended to model the procedure whereby short DNA single strands can be concatenated (assembled) together into longer strands under the action of the enzyme DNA polymerase, provided they have ends that "overlap". Recall that DNA single strands are oriented words from the DNA alphabet $\Delta = \{A, C, G, T\}$, where one end of a strand is labeled by $5'$ and the other by $3'$, and two DNA single strands of opposite orientation, that are Watson-Crick (W/C) complementary, bind to each other to form a DNA double-strand. The W/C complementarity of DNA strands has been traditionally modeled [11, 13] as an antimorphic involution $\theta\colon \Delta^* \longrightarrow \Delta^*$, that is, an involution on Δ (θ^2 is the identity on Δ) extended to an antimorphism on Δ^*, whereby $\theta(uv) = \theta(v)\theta(u)$ for all $u, v \in \Delta^*$. In this formalism, the W/C complement of a DNA strand $u \in \Delta^+$ is $\theta(u)$.

Using the convention that a word x over the DNA alphabet represents the DNA single strand x in the $5'$ to $3'$ direction (usually depicted as the top strand

of a double DNA strand), the *overlap assembly* of a strand uv with a strand $\theta(w)\theta(v)$ is illustrated in Figure 1.

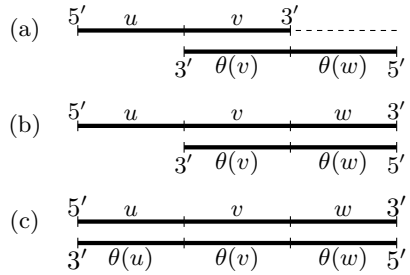

Fig. 1. (a) The two input DNA single-strands, uv and $\theta(w)\theta(v)$ bind to each other through their complementary segments v and $\theta(v)$, forming a partially double-stranded DNA complex. (b) DNA polymerase extends the $3'$ end of the strand uv. (c) DNA polymerase extends the $3'$ end of the other strand. The resulting DNA double strand is considered to be the output of the *overlap assembly* of the two input single strands.

Assuming that all involved DNA strands are initially double-stranded, that is, whenever the strand x is available, its W/C complement $\theta(x)$ is also available, this model was further simplified [4] as follows: Given words x, y over an alphabet Σ, the *overlap assembly of x with y* is defined as:

$$x \odot y = \{z \in \Sigma^+ \mid \exists u, w \in \Sigma^*, \exists v \in \Sigma^+ : x = uv, y = vw, z = uvw\}.$$

This can be naturally generalized to languages: Given languages L_m and L_n of state complexities m and n, respectively, the overlap assembly of L_m and L_n is defined as: $L_m \odot L_n = \{z \mid z = x \odot y,\ x \in L_m, y \in L_n\}$.

3 An ε-NFA for Overlap Assembly

A *deterministic finite automaton (DFA)* is a quintuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite non-empty set of *states*, Σ is a finite non-empty *alphabet*, $\delta\colon Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial* state, and $F \subseteq Q$ is the set of *final* states. We extend δ to functions $\delta\colon Q \times \Sigma^* \to Q$ and $\delta\colon 2^Q \times \Sigma^* \to 2^Q$ as usual. A DFA $\mathcal{D}$ *accepts* a word $w \in \Sigma^*$ if $\delta(q_0, w) \in F$. The language accepted by $\mathcal{D}$ is denoted by $L(\mathcal{D})$. If q is a state of $\mathcal{D}$, then the language $L_q(\mathcal{D})$ of q is the language accepted by the DFA $(Q, \Sigma, \delta, q, F)$. A state is *empty* (or *dead* or a *sink state*) if its language is empty. Two states p and q of $\mathcal{D}$ are *equivalent* if $L_p(\mathcal{D}) = L_q(\mathcal{D})$. A state q is *reachable* if there exists $w \in \Sigma^*$ such that $\delta(q_0, w) = q$. A DFA $\mathcal{D}$ is *minimal* if it has the smallest number of states and the smallest alphabet among all DFAs accepting $L(\mathcal{D})$. It is well known that a DFA is minimal if it uses the smallest alphabet, all of its states are reachable, and no two states are equivalent.

A *nondeterministic finite automaton (NFA)* is a quintuple $\mathcal{N} = (R, \Sigma, \eta, I, F)$, where R, Σ, and F are as Q, Σ, and F in a DFA respectively, $\eta\colon R\times\Sigma \to 2^R$, and $I \subseteq R$ is the *set of initial states*. Each triple (p, a, q) with $p, q \in R$, $a \in \Sigma$ is a *transition* if $q \in \eta(p, a)$. A sequence $((p_0, a_0, q_0), (p_1, a_1, q_1), \ldots, (p_{k-1}, a_{k-1}, q_{k-1}))$ of transitions, where $p_{i+1} = q_i$ for $i = 0, \ldots, k-2$ is a *path* in $\mathcal{N}$. The word $a_0a_1\cdots a_{k-1}$ is the word *spelled* by the path. A word w is *accepted* by $\mathcal{N}$ if there exists a path with $p_0 \in I$ and $q_{k-1} \in F$ that spells w. If $q \in \eta(p, a)$ we also use the notation $p \xrightarrow{a} q$. We extend this notation also to words, and write $p \xrightarrow{w} q$ for $w \in \Sigma^*$. An *ε-NFA* is an NFA in which transitions under the empty word ε are also permitted.

Given any two DFAs, we construct an ε-NFA that recognizes the overlap assembly of the languages accepted by the DFAs. This proves constructively that the family of regular languages is closed under overlap assembly.

Let $\mathcal{D}_m = (Q_m, \Sigma, \delta_m, 0, F)$ and $\mathcal{D}'_n = (Q'_n, \Sigma, \delta'_n, 0', F')$ be two DFAs with $\mathcal{D}_m$ recognizing L_m and $\mathcal{D}'_n$ recognizing L'_n, where $F = \{f_1, \ldots, f_h\}$ and $F' = \{f'_1, \ldots, f'_{h'}\}$. Let $Q_m = \{0, \ldots, m-1\}$, $Q'_n = \{0', \ldots, (n-1)'\}$ and let 0 and $0'$ be the initial states. We claim that the NFA $\mathcal{N}$, constructed as shown below, accepts the result of the overlap assembly of L_m and L'_n.

The NFA is defined as $\mathcal{N} = (R, \Sigma, \eta, \{r_0\}, F_\mathcal{N})$ where the set of states is $R = (Q_m \cup \{t\}) \times (Q'_n \cup \{s'\})$ with s', t new symbols not occurring in $Q_m \cup Q'_n$, the initial state is $r_0 = (0, s')$, and the set of final states is $F_\mathcal{N} = \{(t, q') \mid q' \in F'\}$. Intuitively, the NFA simulates reading the word first by $\mathcal{D}_m$, then by both $\mathcal{D}_m$ and $\mathcal{D}'_n$, and then by $\mathcal{D}'_n$. Hence the states in R contain a state of $\mathcal{D}_m$ and a state of $\mathcal{D}'_n$. The states with s' indicate that $\mathcal{D}'_n$ has not yet read any letter, while the states with t indicate that $\mathcal{D}_m$ has finished the reading. The set of transitions η is defined below. The informal explanations at the right of transition definitions assume two operands $uv \in L_m$ and $vw \in L'_n$ respectively. The word $z = uvw$ belongs to their overlap assembly.

i $\{(q_i, s') \xrightarrow{a} (q_j, s') \mid q_i \xrightarrow{a} q_j \in \delta_m\}$; read u.
ii $\{(q_i, s') \xrightarrow{a} (q_j, q'_k) \mid q_i \xrightarrow{a} q_j \in \delta_m,\ 0' \xrightarrow{a} q'_k \in \delta'_n\}$; read the first letter of v.
iii $\{(q_i, q'_k) \xrightarrow{a} (q_j, q'_\ell) \mid q_i \xrightarrow{a} q_j \in \delta_m,\ q'_k \xrightarrow{a} q'_\ell \in \delta'_n\}$; read the remainder of v.
iv $\{(f_i, q'_k) \xrightarrow{\varepsilon} (t, q'_k) \mid f_i \in F,\ q'_k \in Q'_n\}$; v has been read.
v $\{(t, q'_k) \xrightarrow{a} (t, q'_\ell) \mid q'_k \xrightarrow{a} q'_\ell \in \delta'_n\}$; these rules read w.

Figure 2 shows the construction of an NFA, denoted by $\mathcal{N}'$, for two particular two-state DFAs $\mathcal{D}_2$ and $\mathcal{D}'_2$ accepting the languages $L(D_2)$ (all words over $\{a, b\}^*$ that have an odd number of as) and $L(D'_2)$ (all words over $\{a, b\}^*$ that end in the letter a). Note that the overlap assembly of $L(D_2)$ and $L(D'_2)$ is $L(D'_2)$.

In the automaton $\mathcal{N}'$ of Figure 2, states $(0, s')$ and $(1, s')$ in the first row of the figure behave as specified in Rule (i), using the transitions of $\mathcal{D}_2$. Rule (ii) moves the states from the first row to the second row of the figure. In the second row the transitions are those of the direct product of $\mathcal{D}_2$ and $\mathcal{D}'_2$, as directed by Rule (iii). Note that neither Rule (i) nor Rule (ii) can be used again since s' does not appear as a component of any state after Rule (iii) is used. When $\mathcal{N}'$

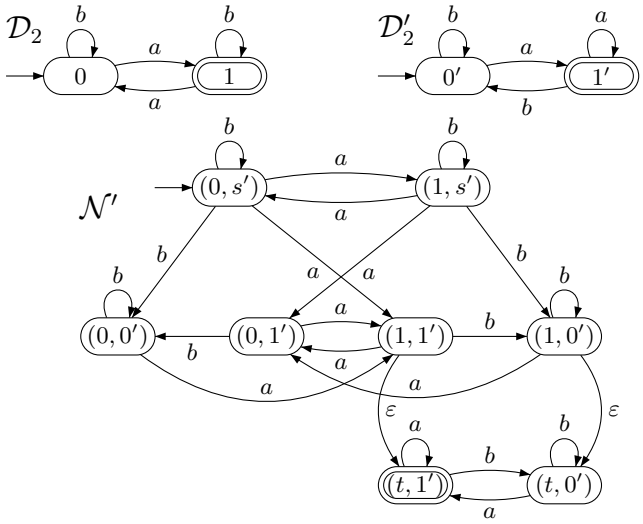

Fig. 2. An example of an NFA $\mathcal{N}'$ that accepts the overlap assembly of the languages accepted by the DFAs $\mathcal{D}_2$ and $\mathcal{D}_2'$.

is in a state where the first component is 1, which is a final state of $\mathcal{D}_2$, $\mathcal{N}'$ can move to the next row following Rule (iv), and change the first component of the state to t. Note that Rule (iii) cannot be used again since t appears as the first component of every state after Rule (iv) is used. Finally, $\mathcal{N}'$ moves to the third row and follows the transitions of $\mathcal{D}_2'$. Note that Rule (iv) cannot be used again because of t. While the NFA $\mathcal{N}'$ has eight states, converting it to a DFA and minimizing this DFA results in D_2'. The NFA $\mathcal{N}'$ accepts the overlap assembly of $L(D_2)$ and $L(D_2')$. In general, the following result holds:

Proposition 1. *Let L_m and L_n' be two regular languages accepted by the DFAs defined above, and let the NFA $\mathcal{N}$ be the automaton constructed as above. NFA $\mathcal{N}$ has the following properties:*

1. *If $uv \in L_m$ and $vw \in L_n'$, then $r_0 \xrightarrow{uvw} r_f$ in $\mathcal{N}$ where $r_f \in F_{\mathcal{N}}$.*
2. *If $r_0 \xrightarrow{z} r_f$ in $\mathcal{N}$, then there exist $u, w \in \Sigma^*$, $v \in \Sigma^+$ such that $z = uvw$, where $uv \in L_m$ and $vw \in L_n'$.*
3. *$\mathcal{N}$ accepts $L_m \odot L_n'$.*

Proof. 1. For the first claim, let $v = ax$, where $a \in \Sigma$. If $uv \in L_m$ then $0 \xrightarrow{uax} f_i$, for some $f_i \in F$ in $\mathcal{D}_m$. So there exist q_i and q_j in Q_m such that $0 \xrightarrow{u} q_i \xrightarrow{a} q_j \xrightarrow{x} f_i$ in $\mathcal{D}_m$. Similarly, if $vw \in L_n$, then there exist q_k' and q_ℓ' in Q_n' such that $0' \xrightarrow{a} q_k' \xrightarrow{x} q_\ell' \xrightarrow{w} f_j'$, for some $f_j' \in F'$ in $\mathcal{D}_n'$.
By construction we have in $\mathcal{N}$:

$$(0, s') \xrightarrow[(i)]{u} (q_i, s') \xrightarrow[(ii)]{a} (q_j, q_k') \xrightarrow[(iii)]{x} (f_i, q_\ell') \xrightarrow[(iv)]{\varepsilon} (t, q_\ell') \xrightarrow[(v)]{w} (t, f_j'),$$

which proves our first claim.

2. Suppose that $r_0 \xrightarrow{z} r_f$ in $\mathcal{N}$, where $r_f \in F_{\mathcal{N}}$. By the construction of $\mathcal{N}$, such a path must proceed by i applications of rule (i), one application of rule (ii), j applications of rule (iii), one ε-transition via rule (iv), and k applications of rule (v), where $i, j, k \geq 0$. Thus there exist u, v, and w in Σ^* such that $z = uvw$, $|u| = i$, $|v| = j + 1$, and $|w| = k$. Owing to the construction of $\mathcal{N}$, there must exist derivations $0 \xrightarrow{uv} f_i$ in $\mathcal{D}_m$ and $0' \xrightarrow{vw} f'_j$ in $\mathcal{D}'_n$, which means $uv \in L_m$ and $vw \in L'_n$.
3. If $x \in L_m$ and $y \in L'_n$, then by (1), for every u, v, w where $x = uv$ and $y = vw$, uvw is recognized by $\mathcal{N}$; so $L_m \odot L_n \subseteq L(\mathcal{N})$. Conversely, if a word z is recognized by $\mathcal{N}$, then by (2), $z = uvw$ for some u, v, w where $uv \in L_m$ and $vw \in L_n$; so $L(\mathcal{N}) \subseteq L_m \odot L_n$. Hence $L(\mathcal{N}) = L_m \odot L_n$. □

4 Tight Upper Bound for Overlap Assembly

To establish the state complexity of overlap assembly we need to determinize the ε-NFA $\mathcal{N} = (R, \Sigma, \eta, r_0, F_{\mathcal{N}})$ defined in Section 3, and then minimize the resulting DFA. The first step is to find an upper bound on the number of subsets S of the set R of states of $\mathcal{N}$. We begin by characterizing the reachable subsets of R. They will all have the form

$$S = \{(q, s')\} \cup (\{q\} \times S') \cup (\{t\} \times T'), \tag{1}$$

where $q \in Q_m$, $T' \subseteq S' \subseteq Q'_n$ if $q \notin F$, $T' = S' \subseteq Q'_n$ if $q \in F$, and S' is non-empty unless $S = \{(0, s')\}$. We call q the *selector* of S, subset $S' \setminus \{0'\}$ is its *core*, and subset T' is its *subcore*.

We illustrate this using the NFA of Figure 2. The initial subset is $\{(0, s')\}$; this has form (1) with $S' = T' = \emptyset$. From this initial subset we reach by b the subset $\{(0, s'), (0, 0')\} = \{0, s'\} \cup (\{0\} \times \{0'\})$; here $T' = \emptyset$ and $S' = \{0'\}$. By a we reach $\{(1, s')\} \cup \{(1, 1')\} \cup \{(t, 1')\} = \{(1, s')\} \cup (\{1\} \times \{1'\}) \cup (\{t\} \times \{1'\})$; here $S' = T' = \{1'\}$.

We now proceed to prove the claim about form (1).

Lemma 1. *Let $m \geq 2$, $n \geq 1$, and let $\mathcal{D}$ be the DFA obtained by determinization of the NFA for the overlap assembly $L_m \odot L_n$. Every reachable subset of $\mathcal{D}$ is of the form* (1)*. Moreover, if $q \notin F$, then S cannot be distinguished from $S \cup \{(q, 0')\}$.*

Proof. First we show that every reachable subset $S \subseteq R$ is of the desired form. We will prove this claim by induction. The initial subset $\{(0, s')\}$ has this form. Suppose that S has this form, consider a letter $a \in \Sigma$, and the subset $U = \eta(S, a)$. Observe that $(\delta_m(q, a), s')$ is the only pair in U containing s', because of the transitions (i) and because $\mathcal{D}_m$ is deterministic. Also, every state (q, p'),where $p' \in Q'_n \cup \{s'\}$, is mapped to a state $(\delta_m(q, a), r') \in \{\delta_m(q, a)\} \times Q'_n$ by the transitions (ii) and (iii). Finally, the states in $\{t\} \times T'$ are mapped only to states from $\{t\} \times Q'_n$ by the transitions (iv) and (v).

Note that subsets S with $S' = \emptyset$ are not reachable, unless S is the initial subset $\{(0, s')\}$.

We show that if $S = \{(q, s')\} \cup (\{q\} \times S') \cup (\{t\} \times T')$ is reachable, then $T' \subseteq S'$. Let $r' \in T'$. Then there exists a word xy such that:

$$(0, s) \xrightarrow{x} (q_1, p') \xrightarrow{\varepsilon} (t, p') \xrightarrow{y} (t, r'),$$

where $q_1 \in F$. We also have: $(q_1, p') \xrightarrow{y} (q_2, r')$. Thus $(q_2, r') \in S$, and so $r' \in S'$.

We observe that if $q \in F$, then by ε-transitions (transitions (iv)), every state $(q, r') \in S$ is mapped to (t, r'), thus $T' = S'$, which concludes the characterization of reachable subsets.

Finally, we show that if $q \notin F$, then S cannot be distinguished from $S \cup \{(q, 0')\}$. Indeed, let $a \in \Sigma$ be any letter. Then $\eta((q, 0'), a) = \eta((q, s'), a)$ because the transitions (iii) and (ii) coincide. Since $(q, s') \in S$, we have $\eta(S, a) = \eta(S \cup \{(q, 0')\}, a)$. □

From Lemma 1 two reachable subsets with a different selector, or a different core, or a different subcore are potentially distinguishable. If two reachable subsets have the same selector, core, and subcore, then they can differ only by state $(q, 0')$ if the selector q is not in F; thus they cannot be distinguished. If two reachable subsets have the same selector q that is in F, then they cannot differ just by $(q, 0')$, as by ϵ-transitions from $(q, 0')$ we immediately obtain $(t, 0')$.

Theorem 1. *For $m \geq 2$ and $n \geq 1$, the state complexity of $L_m \odot L_n$ is at most*

$$2(m-1)3^{n-1} + 2^n.$$

Proof. Using Lemma 1, we count the number of potentially reachable and distinguishable subsets $S = \{(q, s')\} \cup (\{q\} \times S') \cup (\{t\} \times T')$.

Reachable subsets: For every state $q \in Q_m$, we count the number of potentially reachable subsets with selector q. There are 2 cases:

- If q is non-final, we can choose any non-empty set $S' \subseteq Q'_n$ of cardinality k and any subset T' of S'. The number of ways of doing this is $\sum_{k=1}^{n} \binom{n}{k} 2^k$.
- If q is final, again we choose any non-empty set S', but now $T' = S'$ is fixed. The number of ways of doing this is $2^n - 1$.

There is also the initial subset $\{(0, s')\}$ which contributes 1 to the sum. In total, this yields:

$$(m - |F|) \cdot \left(\sum_{k=1}^{n} \binom{n}{k} 2^k \right) + |F| \cdot (2^n - 1) + 1.$$

Distinguishable subsets: The above formula gives the number of potentially reachable subsets, but overestimates the state complexity because not all subsets are distinguishable. Recall that by Lemma 1 if the selector q is not in F, then S cannot be distinguished from $S \cup \{(q, 0')\}$. Thus we do not need to count subsets S without $0'$, as $S \cup \{(q, 0')\}$ is potentially reachable and always equivalent to S. Hence, for a given $q \in Q_m \setminus F$ we choose S' to be any subset of Q'_n that contains $0'$, and again let T' be any subset of S'. This can be done in $\sum_{k=1}^{n} \binom{n-1}{k-1} 2^k$ ways.

Thus the total number of potentially reachable and distinguishable subsets is at most

$$(m-|F|)\cdot\left(\sum_{k=1}^{n}\binom{n-1}{k-1}2^k\right)+|F|\cdot(2^n-1)+1.$$

By algebra, we have $\sum_{k=1}^{n}\binom{n-1}{k-1}2^k = 2\cdot 3^{n-1}$, which is greater than 2^n-1; so this formula is maximized when $|F|=1$, and we conclude that the maximum state complexity of overlap assembly is $2(m-1)3^{n-1}+2^n$. □

Theorem 2. *At least n letters are required to meet the bound from Theorem 1.*

Proof. Let $q\in F$ be a final state of $\mathcal{D}_m$. For each $p'\in Q'_n$ we consider the subset $T_{p'}=\{(q,s'),(q,p'),(t,p')\}$. If the upper bound is met, then, in particular, all subsets S with $q\in F$ must be reachable in view of Lemma 1. These subsets were counted in the upper bound, and there are no other subsets of reachable form that could be equivalent to them when the upper bound is met. Hence, in particular all subsets $T_{p'}$ must be reachable.

Suppose that $T_{p'}$ is reachable by a word $w_{p'}a_{p'}$, for some letter $a_{p'}$. Note that (q,p') is the only one of the three states in $T_{p'}$ that can be reached by transitions (ii) of the NFA. Consider $\eta(r_0,w_{p'})$; it must contain (r,s') for some $r\in Q_m$, because by Lemma 1 every reachable subset has exactly one such pair. Thus, (r,s') must be mapped by transitions (ii) induced by $a_{p'}$ to (q,p'). Therefore, $\delta'_n(0',a_{p'})=p'$, which proves that $a_{p'}$ are different for every p'. □

We define the witness DFAs for $m,n\geq 2$. Let $\Sigma=\{a_0,\ldots,a_{n-1}\}$.

Let $\mathcal{W}_m=(Q_m,\Sigma,\delta_m,0,F)$ be defined as follows: $F=\{0\}$; $a_i\colon \mathbf{1}_m$ for $i\in\{0,2,\ldots,n{-}1\}$, where $\mathbf{1}_m$ is the identity transformation on Q_m; $a_1\colon (0,1,\ldots,m{-}1)$ is a cyclic permutation of Q_m.

Let $\mathcal{W}'_n=(Q'_n,\Sigma,\delta'_n,0',F')$ be defined as follows: $F=\{(n{-}1)'\}$; $a_0\colon (Q'_n\to 0')$ maps all the states of Q'_n to $0'$; $a_i\colon (1',2',3',\ldots,(i-1)',0',i',\ldots,(n-1)')$ for $i\in\{1,\ldots,n-1\}$. Here a_i permutes the states of Q'_n, mapping $1'$ to $2'$, $2'$ to $3'$, etc., then $(i{-}1)'$ to $0'$, $0'$ to i', and then i' to $(i{+}1)'$, etc., and $(n{-}1)'$ to $1'$.

The transitions of these DFAs with $m=3$ and $n=4$ states are illustrated in Figure 3. Let L_m and L'_n be the languages of $\mathcal{W}_m$ and $\mathcal{W}'_n$, respectively.

By a *cyclic shift* of a core subset $S'\subseteq\{1',\ldots,(n-1)'\}$ we understand any subset obtained by shifting the states along the cycle $(1',\ldots,(n{-}1)')$, i positions clockwise, i.e., the subset $\{(((p-1+i) \bmod (n-1))+1)' \mid p'\in S'\}$ for any $i\geq 0$. The *next* and *previous* cyclic shifts correspond to $i=1$ and $i=n-2$, respectively.

The transitions of letters $a_1,a_2,\ldots,a_{n-1}$ produce next cyclic shifts of the states in $\{1',\ldots,(n-1)'\}$, with the exception that state $0'$ replaces one of the states in the cycle. The idea behind the witness is that we can add an arbitrary state to the core using these letters and produce arbitrary cyclic shifts as well, as will be shown later. Letter a_0 plays an important role of reset, which is necessary to reach small subsets. The main difficulty is that a_1 shares both roles of producing cyclic shifts and switching the selector.

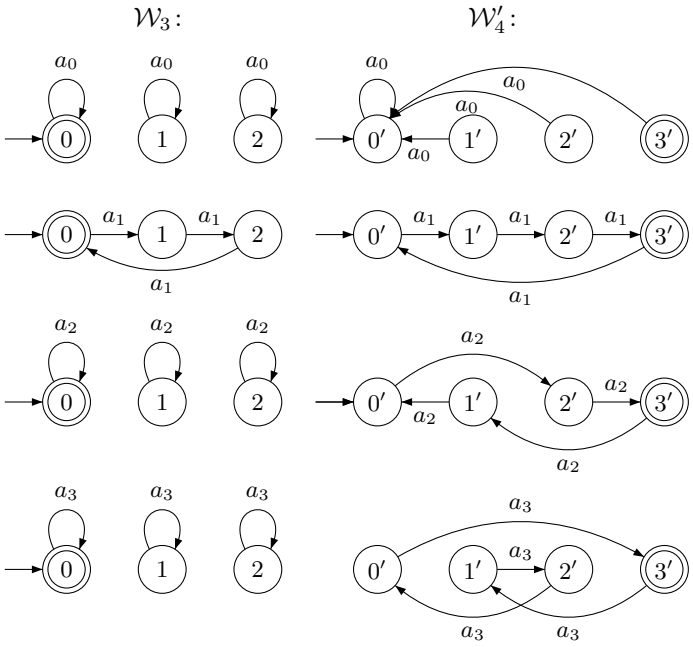

Fig. 3. The action of the letters in $\mathcal{W}_3$ and $\mathcal{W}'_4$.

Theorem 3. *For $m \geq 2$ and $n \geq 3$, $L_m \odot L'_n$ meets the upper bound.*

Proof. Reachability: It is enough to show that all subsets S from Lemma 1 are reachable, with the exception that if $q \notin F$ then it suffices to show reachability of either $S \setminus \{(q, 0')\}$ or $S \cup \{(q, 0')\}$.

• First we show that for all subsets $S = \{(q, s')\} \cup (\{q\} \times S')$, where $q \in Q_m \setminus \{0\}$ and $\emptyset \neq S' \subseteq Q'_n \setminus \{0'\}$, either $S \setminus \{(q, 0')\}$ or $S \cup \{(q, 0')\}$ is reachable. These subsets have core S' and an empty subcore.

We prove this by induction on the size $|S'|$ of the core. For $|S'| = 0$, apply $a_1^q a_0$ to $(0, s')$; this yields $\{(q, s'), (q, 0')\}$.

Consider $|S'| = 1$. If $q = 1$, then we just use a_1, which yields $\{(1, s'), (1, 1')\}$. To meet the other subsets $\{(1, s'), (1, p')\}$ for $p \geq 2$, from $\{(1, s'), (1, 1')\}$ we use $a_0 a_p$. For $q \geq 2$, we use $a_1^{q-1} a_0 a_1$, which yields $\{(q, s'), (q, 1')\}$. Then to meet the other subsets $\{(q, s'), (q, p')\}$ for $p \geq 2$, from $\{(q, s'), (q, 1')\}$ we also use $a_0 a_p$.

Consider $|S'| \geq 2$ and assume the induction hypothesis for subsets S with a smaller core. Since S' contains at least two states different from $0'$, there is a state $p' \in S' \setminus \{1'\}$. Let X' be the previous cyclic shift of $S' \setminus \{p'\}$. Since $p' \notin S' \setminus \{p'\}$, X' does not contain $(p-1)'$, but this is its only difference from the previous cyclic shift of S'. By the inductive assumption, $\{(q, s')\} \cup (\{q\} \times X')$ is reachable. We apply a_p to this subset, which maps X' to its next cyclic shift, and also (q, s') to (q, p'), which yields $\{(q, s'\} \cup (\{q\} \times S')$.

• Now we show reachability of subsets $S = \{(0,s')\} \cup (\{0\} \times S') \cup (\{t\} \times S')$, where $\emptyset \neq S' \subseteq Q'_n$. These are all potentially reachable subsets with selector 0.

First consider the case $0' \notin S'$. For $\{(m-1,s'),(m-1,1')\}$ we apply a_0a_1, which yields $\{(0,s'),(0,1'),(t,1')\}$. Then we continue the induction on $|S'|$ as before when $|S'| \geq 2$, with just $\{t\} \times S'$ added to the subsets.

Now consider the case $0' \in S'$. The case $S' = \{0'\}$ is easily covered by applying a_0 to $\{(0,s'),(0,1'),(t,1')\}$. If $S' = \{0',1'\}$, then from $\{(m-1,s'),(m-1,(n-1)')\}$ we apply a_1 and get $\{(0,s'),(0,0'),(0,1'),(t,0'),(t,1')\}$ as desired. Let $S' \neq \{0',1'\}$. We already know that $\{(0,s')\} \cup (\{0,t\} \times X')$ is reachable, where X' is the previous cyclic shift of $S' \setminus \{0'\}$. Since $|S'| \geq 2$ and $S' \neq \{0',1'\}$, there is a $p' \in S' \setminus \{1'\}$. We apply a_p to $\{(0,s')\} \cup (\{0,t\} \times X')$. We have $X' \setminus \{(p-1)'\}$ mapped to $S' \setminus \{p'\}$ and $(p-1)'$ mapped to $0'$, which gives $(\{0\} \times (S' \cup \{0'\} \setminus \{p'\})$ by transitions (iii), and $(0,p')$ is added by transitions (ii). Thus, after completing by ε-transitions this yields $\{(0,s')\} \cup (\{0,t\} \times S')$.

• Finally, we show that for all subsets $S = \{(q,s')\} \cup (\{q\} \times S') \cup (\{t\} \times T')$, where $q \neq 0$ and $\emptyset \neq T' \subseteq S' \subseteq Q'_n$, either $S \setminus \{(q,0')\}$ or $S \cup \{(q,0')\}$ is reachable.

Consider the special case $S' = T' = \{0'\}$. We reach it from $\{(0,s'),(0,0'),(t,0')\}$ by applying $a_1^q a_0$. For the rest, assume that $S' \setminus \{0'\}$ is non-empty.

We need an auxiliary argument that from $\{(0,s')\}$ we can reach a subset with selector q, core S', and an empty subcore, using a word from $\{a_1,a_2,\ldots,a_{n-1}\}^*$ (any word without a_0). We prove this by induction on the core size $|S' \setminus \{0'\}|$. For $|S' \setminus \{0'\}| = 1$, at the beginning we use a_1, which yields $\{(1,s'),(1,1')\}$. Now we can reach $\{(1,s'),(1,0'),(1,p')\}$ for any $p' \in \{2',\ldots,(n-1)'\}$ by using $a_2a_3\ldots a_p$. Then, from $\{(1,s'),(1,0'),(1,(n-1)')\}$ we reach $\{(2,s'),(2,0'),(2,1')\}$, and it remains to repeat the argument to reach every remaining subset of the form $\{(q,s'),(q,0'),(q,p')\}$ for $q \in Q_m \setminus \{0,1\}$ and $p' \in Q'_n \setminus \{0'\}$. For $|S' \setminus \{0'\}| \geq 2$ we follow the first part of the reachability argument as before, but we reach either $\{(q,s')\} \cup (\{q\} \times (S' \setminus \{0'\})$ or $\{(q,s')\} \cup (\{q\} \times (S' \cup \{0'\}))$, instead of just the former. Let $w \in \{a_1,a_2,\ldots,a_{n-1}\}^*$ be a word that reaches either $\{(q,s')\} \cup (\{q\} \times (S' \setminus \{0'\})$ or $\{(q,s')\} \cup (\{q\} \times (S' \cup \{0'\}))$.

Suppose that we start from the subset $S_0 = \{(0,s')\} \cup (\{0,t\} \times T'_0)$, where T'_0 is some subset such that $\emptyset \neq T'_0 \subseteq Q'_n$. We already know that for every T'_0, subset S_0 is reachable. After applying a_1w, we reach either

$$S_q = \{(q,s')\} \cup (\{q\} \times (S' \cup T'_q \setminus \{0'\})) \cup (\{t\} \times T'_q),$$

or $S_q \cup \{(q,0')\}$, where T'_q is obtained by applying some permutation π of Q'_n to T'_0. This is because $\{(0,s')\}$ is mapped by a_1w to $\{(q,s')\} \cup (\{q\} \times (S' \setminus \{0'\})$ or $\{(q,s')\} \cup (\{q\} \times (S' \cup \{0'\}))$, word a_1w acts as a permutation on $(\{t\} \times Q'_q)$, and $\{0\} \times T'_0$ is mapped to $(\{q\} \times T'_q)$. Note that a_1w does not depend on T'_0, so we can choose T'_0 arbitrarily. Let $T'_0 = \pi^{-1}(T')$, so $\pi(T'_0) = T'$. We obtain either

$$S_q = \{(q,s')\} \cup (\{q\} \times ((S' \setminus \{0'\}) \cup T') \cup (\{t\} \times T'),$$

$$\text{or } S_q = \{(q,s')\} \cup (\{q\} \times ((S' \cup \{0'\}) \cup T') \cup (\{t\} \times T').$$

Recall that $T' \subseteq S'$ and if $0' \in T$ then also $0' \in S'$; hence $(S' \setminus \{0'\}) \cup T'$ is either S' or $S' \setminus \{0'\}$, and $(S' \cup \{0'\}) \cup T' = S' \cup \{0'\}$. Thus, S_q is either $S \setminus \{(q,0')\}$ or $S \cup \{(q,0')\}$.

Distinguishability: Consider two reachable subsets

$$S_1 = \{(q_1, s')\} \cup (\{q_1\} \times S'_1) \cup (\{t\} \times T'_1),$$

$$\text{and } S_2 = \{(q_2, s')\} \cup (\{q_2\} \times S'_2) \cup (\{t\} \times T'_2),$$

with different selectors, different cores, or different subcores. Thus we have $q_1 \neq q_2$, or $T'_1 \neq T'_2$, or $(S'_1 \setminus \{(q_1, 0')\}) \neq (S'_2 \setminus \{(q_2, 0'\})$. These are precisely all the reachable and potentially distinguishable subsets in view of Lemma 1. Note that the initial subset also has this form, where $q_1 = 0$ and S'_1 and T'_1 are empty.

If $q_1 \neq q_2$, then without loss of generality let $q_1 < q_2$. We apply $a_1^{m-q_2} a_0 a_{n-1}^2$. For S_1, first $a_1^{m-q_2} a_0$ maps it to a subset $\{(q, s'), (0, s')\}$ or $\{(q, s'), (q, 0'), (t, 0')\}$ (if T'_1 is non-empty) for some $q \neq 0$. Then a_{n-1}^2 results in a subset that from the states from $(\{t\} \times Q'_n)$ contains at most $(t, 1')$, which is not final. On the other hand, S_2 by $a_1^{m-q_2} a_0$ is mapped to $\{(0, s'), (0, 0'), (t, 0')\}$. Then $a_{n_1}^2$ yields $\{(0, s'), (0, 0'), (t, 1'), (t, (n-1)')\}$, where $(t, (n-1)')$ is final.

So suppose that $q_1 = q_2$. If $q_1 \neq 0$ and $T'_1 \neq T'_2$, then we apply a_{n-1}^i for a suitable $i \geq 0$. Since a_{n-1} acts cyclically on all states $(\{t\} \times Q'_n)$ and no other states from the subsets are mapped to $(\{t\} \times Q'_n)$, we can repeat the cycle so that exactly one of $\eta(\{t\} \times T'_1, a_{n-1}^i)$ and $\eta(\{t\} \times T'_2, a_{n-1}^i)$ contains the final state $(t, (n-1)')$. If $q_1 = 0$ and $T'_1 \neq T'_2$, then also $S'_1 \neq S'_2$, so it remains to cover this case.

Suppose that $S'_1 \neq S'_2$. If $q_1 = q_2 = 0$, then also $T'_1 \neq T'_2$. We apply a_1, which maps S_1 to the subset $\{(1, s')\} \cup (\{1\} \times (\delta_m(S'_1, a_1) \cup \{2'\})) \cup (\{t\} \times \delta'_n(T'_1, a_1))$, and analogously S_2. Since $T'_1 \neq T'_2$ and a_1 acts cyclically on Q'_n, we have $\delta'_n(T'_1, a_1) \neq \delta'_n(T'_2, a_1)$. The case of these subsets has been covered in the previous paragraph.

There remains the case where $T'_1 = T'_2$, $S'_1 \neq S'_2$, $q_1 = q_2 \neq 0$. We follow the induction on the selector q_1 starting with $q_1 = m - 1$ and decreasing it. We will show for $q_1 = m - 1$ that we can reach subsets with selector 0 that still have different cores. We have already shown in the previous paragraph that the subsets with selector 0 and different cores can be distinguished. For $q_1 < m - 1$ we will show that we can reach subsets with the same property but with selector $q_1 + 1$, which will follow by the inductive assumption. So let p be the largest index such that, without loss of generality, $p' \in S'_1$ and $p' \notin S'_2$. Note that $p \neq 0$, because then the subsets cannot be distinguished. If $p < n-1$, then we apply a_1, which yields subsets with the desired property. If $p = n - 1$, then we first apply a_2, which yields the subset with $p' = 1'$, and then we can apply a_1 as before. □

5 Conclusions

We have found an upper bound of $2(m-1)3^{n-1} + 2^n$ on the state complexity of overlap assembly, a biologically inspired operation on regular languages, and we have shown that this bound is tight for languages over an alphabet of size n. For completeness, we state without proof some results about the unary and binary languages. Proofs can be found in [2].

Theorem 4. *Let $m, n \geq 1$, and let L_m and L_n be two unary languages of state complexities m and n, respectively. The state complexity of $L_m \odot L_n$ is at most $m + n$, and this bound is met by $L_m = \{a^{mk+n-1} \mid k \in \mathbb{Z}, mk + n - 1 \geq 0\}$ and $L_n = \{a^{nk+m-1} \mid k \in \mathbb{Z}, nk + m - 1 \geq 0\}$.*

For binary languages we have found an exponential lower bound on the complexity of overlap assembly; the proof is based on ideas similar to those in the proof of Theorem 3.

Theorem 5. *For every $m \geq 2$ and $n \geq 3$, there exist binary DFAs $\mathcal{B}_m$ and $\mathcal{B}'_n$ such that the state complexity of $L(\mathcal{B}_m) \odot L(\mathcal{B}'_n)$ is at least $m(2^{n-1} - 2) + 2$.*

References

1. Brzozowski, J.A.: Towards a theory of complexity of regular languages. J. Autom. Lang. Comb. (2018), to appear. Also at `http://arxiv.org/abs/1702.05024`
2. Brzozowski, J.A., Kari, L., Li, B., Szykuła, M.: State complexity of overlap assembly (2017), `http://arxiv.org/abs/1710.06000`
3. Carausu, A., Paun, G.: String intersection and short concatenation. Rev. Roumaine Math. Pures Appl. **26**, 713–726 (1981)
4. Csuhaj-Varjú, E., Petre, I., Vaszil, G.: Self-assembly of strings and languages. Theoret. Comput. Sci. **374**(1–3), 74–81 (2007)
5. Domaratzki, M.: Minimality in template-guided recombination. Information and Computation **207**(11), 1209–1220 (2009)
6. Enaganti, S.K., Ibarra, O.H., Kari, L., Kopecki, S.: On the overlap assembly of strings and languages. Nat. Comput. **16**(1), 175–185 (2016)
7. Enaganti, S.K., Ibarra, O.H., Kari, L., Kopecki, S.: Further remarks on DNA overlap assembly. Inform. and Comput. (2017)
8. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. J. Autom. Lang. Comb. **21**(4), 251–310 (2016)
9. Golan, J.S.: The theory of semirings with applications in mathematics and theoretical computer science. Addison-Wesley Longman Ltd. (1992)
10. Holzer, M., Jakobi, S., Kutrib, M.: The chop of languages. Theoret. Comput. Sci. **682**, 122–137 (2017)
11. Hussini, S., Kari, L., Konstantinidis, S.: Coding properties of DNA languages. In: Jonoska, N., Seeman, N.C. (eds.) Proc. DNA Computing, (DNA 7). LNCS, vol. 2340, pp. 57–69 (2002)
12. Ito, M., Lischke, G.: Generalized periodicity and primitivity for words. Mathematical Logic Quarterly **53**(1), 91–106 (2007)
13. Kari, L., Kitto, R., Thierrin, G.: Codes, involutions, and dna encodings. In: Brauer, W., Ehrig, H., Karhumäki, J., Salomaa, A. (eds.) Formal and Natural Computing. LNCS, vol. 2300, pp. 376–393 (2002)
14. Maslov, A.N.: Estimates of the number of states of finite automata. Dokl. Akad. Nauk SSSR **194**, 1266–1268 (Russian). (1970), English translation: Soviet Math. Dokl. **11** (1970) 1373–1375
15. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. **125**, 315–328 (1994)