A New Technique for Reachability of States in Concatenation Automata

Sylvie Davies

A New Technique for Reachability of States in Concatenation Automata

Sylvie Davies

Department of Pure Mathematics
University of Waterloo
sldavies@uwaterloo.ca

Abstract. We present a new technique for demonstrating the reachability of states in deterministic finite automata representing the concatenation of two languages. Such demonstrations are a necessary step in establishing the state complexity of the concatenation of two languages, and thus in establishing the state complexity of concatenation as an operation. Typically, ad-hoc induction arguments are used to show particular states are reachable in concatenation automata. We prove some results that seem to capture the essence of many of these induction arguments. Using these results, reachability proofs in concatenation automata can often be done more simply and without using induction directly.

1 Introduction

The *state complexity* of a regular language L, denoted $\mathrm{sc}(L)$, is the least number of states needed to recognize the language with a deterministic finite automaton. The *state complexity of an operation* on regular languages is the worst-case state complexity of the result of the operation, expressed as a function of the maximal allowed state complexity of the input languages.

To establish the state complexity of an operation, there are two steps. First, one derives an upper bound on the state complexity. Next, one searches for witnesses, that is, families of languages which attain the upper bound. One must not only find these witnesses but also *prove* that the desired state complexity bound is reached. Such proofs are the subject of this paper.

We are interested in the case where the operation is *concatenation* of languages. We assume that one is working within some subclass of the regular languages, and has derived an upper bound $f(m, n)$ for the worst-case state complexity of concatenation within this subclasses. We also assume one has found candidate witnesses for this upper bound, in the form of two sequences of languages $(L_m : m \geq 1)$ and $(K_n : n \geq 1)$ such that $\mathrm{sc}(L_m) \leq m$ and $\mathrm{sc}(K_n) \leq n$. The goal is to prove that for each pair (m, n), the concatenation $L_m K_n$ has state complexity $f(m, n)$. We may divide such a proof into three steps:

1. Construct a deterministic automaton $\mathcal{A}$ for $L_m K_n$ in the standard way.
2. Show that $\mathcal{A}$ contains at least $f(m, n)$ reachable states.
3. Show exactly $f(m, n)$ reachable states in $\mathcal{A}$ are pairwise distinguishable.

We present a new technique for dealing with step (2) of this process. The standard way to construct a deterministic finite automaton $\mathcal{A}$ for the concatenation of two languages yields an automaton in which the states are *sets*; to show a particular set is reachable, one typically proceeds by induction on the size of the set. We prove a result that seems to generalize many of these ad-hoc induction arguments, and can be used to establish reachability of sets without directly using induction. Additionally, we prove some helpful lemmas that make our main result easier to apply.

We have tested our technique by applying it to a variety of concatenation witnesses taken from the literature. The state complexity of concatenation has been studied in the class of all regular languages, as well as many subclasses. Table 1 lists some examples of subclasses that have been studied, and the state complexity of concatenation in each subclass. See the cited papers for definitions of each subclass and derivations/proofs of each complexity.

Subclass	Complexity	Subclass	Complexity
Regular [3, 9, 16, 20]	$\mathbf{(m-1)2^n + 2^{n-1}}$	Prefix-closed [2, 9]	$\mathbf{(m+1)2^{n-2}}$
Unary [17, 18, 20]	$\sim mn$ (asymptotically)	Prefix-free [9, 13, 15]	$m+n-2$
Finite unary [10, 19]	$m+n-2$	Suffix-closed [2, 8]	$mn-n+1$
Finite binary [10]	$\mathbf{(m-n+3)2^{n-2}-1}$	Suffix-free [8, 14]	$\mathbf{(m-1)2^{n-2}+1}$
Star-free [7]	$\mathbf{(m-1)2^n + 2^{n-1}}$	Right ideal [1, 6, 9]	$\mathbf{m+2^{n-2}}$
Non-returning [5, 12]	$\mathbf{(m-1)2^{n-1}+1}$	Left ideal [1, 6, 8]	$m+n-1$

Table 1. Subclasses of regular languages and the state complexity of the concatenation operation within each subclass. **Bold** type indicates that the complexity grows exponentially in terms of n.

If the state complexity of concatenation grows exponentially with n (indicated in Table 1 by **bold** type), it is typical to use an induction argument to prove the desired number of states is reachable. It is cases like this in which our technique is most likely to be useful. We selected 16 concatenation witnesses, all from subclasses in which the state complexity of concatenation is exponential in n, and tried to apply our technique to these witnesses. (In the interest of space, we present just two of these applications; the other 14 can be found in the arXiv version of this paper [11].) In many cases we were able to produce shorter and simpler proofs than the original authors, and we only found two cases in which our technique did not work or was not useful. This suggests that our technique is widely applicable and should be considered as a viable alternative to the traditional induction argument when attempting reachability proofs in concatenation automata.

The rest of the paper is structured as follows. Section 2 contains background material and definitions needed to understand the paper. Section 3 describes our new technique and proves the relevant results. Section 4 concludes the paper.

2 Preliminaries

2.1 Relations and Functions

A *binary relation* ρ between X and Y is a subset of $X \times Y$. If $\rho \subseteq X \times Y$ and $\tau \subseteq Y \times Z$, the *composition* of ρ and τ is the relation

$$\rho\tau = \{(x, z) \in X \times Z : \text{ there exists } y \in Y \text{ such that } (x, y) \in \rho \text{ and } (y, z) \in \tau\}.$$

For $x \in X$ and $\rho \subseteq X \times Y$, the *image* of x under ρ is the set $x\rho = \{y \in Y : (x, y) \in \rho\}$. For $x \notin X$ we define $x\rho = \emptyset$. The *converse* of a binary relation $\rho \subseteq X \times Y$ is the relation $\rho^{-1} = \{(y, x) : (x, y) \in \rho\} \subseteq Y \times X$. The set $y\rho^{-1} = \{x \in X : (x, y) \in \rho\}$ is called the *preimage* of y under ρ. Elements of this set are callled *preimages* of y; for example, if $x \in y\rho^{-1}$ we say that x is a preimage of y.

If we write $\mathcal{P}(S)$ for the power set of a set S (that is, the set of all subsets of S), then we can view ρ as a map $\rho\colon X \to \mathcal{P}(Y)$. We may also *extend ρ by union* to a map $\rho\colon \mathcal{P}(X) \to \mathcal{P}(Y)$ as follows: for $S \subseteq X$, we define

$$S\rho = \bigcup_{s \in S} s\rho.$$

We thus have two ways to make sense of an expression like $x\rho\tau$: it is the image of x under the composite relation $\rho\tau \subseteq X \times Z$, and it is also the image of the set $x\rho \subseteq Y$ under the map $\tau\colon \mathcal{P}(Y) \to \mathcal{P}(Z)$. Additionally, we have a way to make sense of a composition $\rho\tau\colon X \to \mathcal{P}(Z)$ of maps $\rho\colon X \to \mathcal{P}(Y)$ and $\tau\colon Y \to \mathcal{P}(Z)$: take the composition of the corresponding relations.

A *function* $f\colon X \to Y$ is a binary relation $f \subseteq X \times Y$ such that $|xf| = 1$ for all $x \in X$. Following our notation for binary relations, we write functions to the *right* of their arguments. Composition of functions is defined by composing the corresponding relations. Thus the order of composition is *left-to-right*; in a composition fg, first f is applied and then g.

A *transformation* of a set X is a function $t\colon X \to X$, that is, a function from X into itself. We say t is a *permutation* of X if $Xt = X$. We say t *acts as a permutation* on $S \subseteq X$ if $St = S$. If t acts as a permutation on S, then every element of S has at least one preimage under t, that is, for all $s \in S$, the set $st^{-1} = \{x \in X : xt = s\}$ is non-empty.

A *cyclic permutation* of a set $\{x_1, \ldots, x_k\} \subseteq X$ is a permutation p such that $x_i p = x_{i+1}$ for $1 \leq i < k$, $x_k p = x_1$, and $xp = x$ for all $x \in X \setminus \{x_1, \ldots, x_k\}$. We denote such a permutation as $(x_1, \ldots, x_k)$. A *transposition* is a cyclic permutation of a two-element set. We denote the identity transformation by id.

The notation $(S \to x)$ for $S \subseteq X$ and $x \in X$ denotes a transformation that sends every element of S to x and fixes every element of $S \setminus X$. For example, $(\{i\} \to j)$ maps i to j and fixes everything else, and $(X \to x)$ is a constant transformation that maps every element of X to x. In the case where $X = \{1, 2, \ldots, n\}$, the notation $({}_i^j x \to x + 1)$ denotes a transformation such that for each x with $i \leq x \leq j$, the transformation sends x to $x + 1$, and every other x is fixed. For example, the transformation $({}_2^{n-1} x \to x + 1)$ fixes 1, sends x to $x + 1$ for $2 \leq x \leq n - 1$, and fixes n. The notation $({}_i^j x \to x - 1)$ is defined similarly.

2.2 Automata

A *finite automaton* (FA) is a tuple $\mathcal{A} = (Q, \Sigma, T, I, F)$ where Q is a finite set of *states*, Σ is a finite set of *letters* called an *alphabet*, $T \subseteq Q \times \Sigma \times Q$ is a set of *transitions*, $I \subseteq Q$ is a set of *initial states*, and $F \subseteq Q$ is a set of *final states*.

We now define a binary relation $T_w \subseteq Q \times Q$ for each $w \in \Sigma^*$. Define $T_\varepsilon = \{(q, q) : q \in Q\}$; in terms of maps, this is the identity map on Q. For $a \in \Sigma$, define $T_a = \{(p, q) \in Q \times Q : (p, a, q) \in T\}$. For $w = a_1 \cdots a_k$ with $a_1, \ldots, a_k \in \Sigma$, define $T_w = T_{a_1} \cdots T_{a_k}$. The relation T_w is called the *relation induced by* w. The set $\{T_w : w \in \Sigma^*\}$ is a monoid under composition, called the *transition monoid* of $\mathcal{A}$. If w is a word but is *not* a word over Σ, we define T_w to be the empty relation. We sometimes write $p \xrightarrow{w} q$ to mean $q \in pT_w$.

If $\mathcal{A} = (Q, \Sigma, T, I, F)$ is a finite automaton such that $|I| = 1$ and T_a is a *function* for each $a \in \Sigma$, we say $\mathcal{A}$ is *deterministic*. We abbreviate "deterministic finite automaton" to DFA.

Let $\mathcal{A} = (Q, \Sigma, T, I, F)$ be an FA. A word $w \in \Sigma^*$ is *accepted* by $\mathcal{A}$ if we have $IT_w \cap F \neq \emptyset$. If $\mathcal{A}$ is a DFA with $I = \{i\}$, this condition becomes $iT_w \in F$. The *language* of $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of all words it accepts. A language recognized by an FA is called a *regular language*.

Given two regular languages L and K with DFAs $\mathcal{A} = (Q^\mathcal{A}, \Sigma^\mathcal{A}, T^\mathcal{A}, i^\mathcal{A}, F^\mathcal{A})$ and $\mathcal{B} = (Q^\mathcal{B}, \Sigma^\mathcal{B}, T^\mathcal{B}, i^\mathcal{B}, F^\mathcal{B})$, we may construct an FA $\mathcal{AB} = (Q, \Sigma, T, I, F)$ that accepts the concatenation LK as follows:

- $Q = Q^\mathcal{A} \cup Q^\mathcal{B}$. We assume without loss of generality that $Q^\mathcal{A} \cap Q^\mathcal{B} = \emptyset$.
- $\Sigma = \Sigma^\mathcal{A} \cup \Sigma^\mathcal{B}$.
- $T = T^\mathcal{A} \cup T^\mathcal{B} \cup \{(q, a, i^\mathcal{B}) : qT^\mathcal{A}_a \in F^\mathcal{A}, a \in \Sigma^\mathcal{A}\}$.
- $I = \{i^\mathcal{A}\}$ if $i^\mathcal{A} \notin F^\mathcal{A}$, and otherwise $I = \{i^\mathcal{A}, i^\mathcal{B}\}$.
- $F = F^\mathcal{B}$.

Next, we convert the FA $\mathcal{AB} = (Q, \Sigma, T, I, F)$ to a DFA recognizing the same language. We apply ththe usual subset construction to obtain the DFA $\mathcal{C} = (\mathcal{P}(Q), \Sigma, T^\mathcal{C}, I, F^\mathcal{C})$, where $T^\mathcal{C} = \{(S, a, ST_a) : S \subseteq Q, a \in \Sigma\}$, and $S \subseteq Q$ is in $F^\mathcal{C}$ if $S \cap F \neq \emptyset$. We call $\mathcal{C}$ the *concatenation DFA* for $\mathcal{A}$ and $\mathcal{B}$.

We make some observations and introduce some conventions to make it easier to work with the concatenation DFA.

- Since we are assuming $\mathcal{A}$ and $\mathcal{B}$ are DFAs, the only reachable states in $\mathcal{C}$ have the form $S^\mathcal{A} \cup S^\mathcal{B}$, where $S^\mathcal{A} \subseteq Q^\mathcal{A}$, $S^\mathcal{B} \subseteq Q^\mathcal{B}$, and $|S^\mathcal{A}| \leq 1$. Without loss of generality, we can assume the state set of $\mathcal{C}$ consists of states of this form, rather than all of $\mathcal{P}(Q)$.
- We mark the states of $\mathcal{A}$ with primes so they can be distinguished from the states of $\mathcal{B}$. So a variable named p or q generally means an element of $Q^\mathcal{B}$, while p' or q' means an element of $Q^\mathcal{A}$.
- We identify the set $S^\mathcal{A} \cup S^\mathcal{B}$ with the ordered pair $(S^\mathcal{A}, S^\mathcal{B})$. Hence we can view the states of $\mathcal{C}$ as these ordered pairs. Reachable states are either of the form $(\emptyset, S)$ or $(\{q'\}, S)$ with $q' \in Q^\mathcal{A}$, $S \subseteq Q^\mathcal{B}$.
- For convenience, we frequently make no distinction between singleton sets and the elements they contain, and so write (q', S) rather than $(\{q'\}, S)$.

– Rather than T_w, $T_w^{\mathcal{A}}$ and $T_w^{\mathcal{B}}$, we simply write w when it is clear from context which relation is meant. For example, $(q', S)w$ means $(q', S)T_w$ since (q', S) is a state of $\mathcal{C}$, and thus T_w is the natural relation to apply. From our convention for marking the states of $\mathcal{A}$ and $\mathcal{B}$ with primes, one can infer that $q'w$ means $q'T_w^{\mathcal{A}}$ and qw means $qT_w^{\mathcal{B}}$.
– Rather than $i^{\mathcal{A}}$ and $i^{\mathcal{B}}$, let $1'$ denote the initial state of $\mathcal{A}$ and let 1 denote the initial state of $\mathcal{B}$. We also assume without loss of generality that $Q^{\mathcal{A}} = \{1', 2', \ldots, m'\}$ and $Q^{\mathcal{B}} = \{1, 2, \ldots, n\}$ for some m and n.

Under these conventions, the transitions of $\mathcal{C}$ can be described as follows:

$$(q', S)a = \begin{cases} (\emptyset, Sa), & \text{if } a \in \Sigma^{\mathcal{B}} \setminus \Sigma^{\mathcal{A}}; \\ (q'a, \emptyset), & \text{if } a \in \Sigma^{\mathcal{A}} \setminus \Sigma^{\mathcal{B}} \text{ and } q'a \notin F^{\mathcal{A}}; \\ (q'a, 1), & \text{if } a \in \Sigma^{\mathcal{A}} \setminus \Sigma^{\mathcal{B}} \text{ and } q'a \in F^{\mathcal{A}}; \\ (q'a, Sa), & \text{if } a \in \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}} \text{ and } q'a \notin F^{\mathcal{A}}; \\ (q'a, Sa \cup 1), & \text{if } a \in \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}} \text{ and } q'a \in F^{\mathcal{A}}. \end{cases}$$

Recall that $T_w^{\mathcal{A}}$ is the empty relation if w is not a word over $\Sigma^{\mathcal{A}}$, and a similar statement can be made for $\mathcal{B}$. Thus the transitions admit a simpler description:

$$(q', S)a = \begin{cases} (q'a, Sa \cup 1), & \text{if } a \in \Sigma^{\mathcal{A}} \text{ and } q'a \in F^{\mathcal{A}}; \\ (q'a, Sa), & \text{otherwise.} \end{cases}$$

2.3 State Complexity

We say a DFA $\mathcal{A}$ is *minimal* if it has the least number of states among all DFAs that recognize $L(\mathcal{A})$. It is well known that each regular language has a unique minimal DFA (up to renamings of the states). The *state complexity* of a regular language L, denoted $\mathrm{sc}(L)$, is the number of states in its minimal DFA.

The following characterization of minimality is useful. Let $\mathcal{D} = (Q, \Sigma, T, i, F)$ be a DFA. A state $q \in Q$ is *reachable* if $iw = q$. For $p, q \in Q$, we say q is *reachable from* p if $pw = q$. Two states $p, q \in Q$ are *indistinguishable* if they are equivalent under the following equivalence relation: $p \sim q$ if for all $w \in \Sigma^*$, we have $pw \in F \iff qw \in F$. Otherwise they are *distinguishable* by some word w such that $pw \in F \iff qw \notin F$. A DFA is minimal if and only if all of its states are reachable and pairwise distinguishable.

Let $\circ$ be a binary operation on regular languages. The *state complexity of the operation* $\circ$ is the following function, where m and n are positive integers:

$$(m, n) \mapsto \max\{\mathrm{sc}(L \circ K) : \mathrm{sc}(L) \leq m, \mathrm{sc}(K) \leq n\}.$$

This is the worst-case state complexity of the result of the operation, expressed as a function of the maximal allowed state complexities of the input languages. When computing state complexity of operations we assume the input languages L and K are languages over the same alphabet. The case where the inputs are allowed to have different alphabets has been studied by Brzozowski [4]. An example of how to apply our results in this case is given in [11, Theorem 4].

3 Results

Let $\mathcal{A} = (Q^{\mathcal{A}}, \Sigma^{\mathcal{A}}, T^{\mathcal{A}}, 1', F^{\mathcal{A}})$ and $\mathcal{B} = (Q^{\mathcal{B}}, \Sigma^{\mathcal{B}}, T^{\mathcal{B}}, 1, F^{\mathcal{B}})$ be DFAs, with $Q^{\mathcal{A}} = \{1', 2', \ldots, m'\}$ and $Q^{\mathcal{B}} = \{1, 2, \ldots, n\}$ for positive integers m and n. Let $\mathcal{C} = (Q, \Sigma, T, I, F)$ denote the concatenation DFA of $\mathcal{A}$ and $\mathcal{B}$ as defined in Section 2.2.

Remark 1. Let $p', q' \in Q^{\mathcal{A}}$, let $X, Y, Z \subseteq Q^{\mathcal{B}}$, and let $w \in \Sigma^*$. Then in $\mathcal{C}$, if $(p', X)w = (q', Y)$, then $(p', X \cup Z)w = (q', Y \cup Zw)$.

Indeed, recall that the pair (p', X) stands for the set $\{p'\} \cup X$. Thus $(\{p'\} \cup X)w = \{p'w\} \cup Xw = \{q'\} \cup Y$. It follows that $p'w = q'$ and $Xw = Y$. Hence $(\{p'\} \cup X \cup Z)w = \{p'w\} \cup Xw \cup Zw = \{q'\} \cup Y \cup Zw$, which in our pair notation is $(q', Y \cup Zw)$. We will readily use this basic fact in proofs.

Before stating our main result formally, we give some motivating exposition. Fix a state $s' \in Q^{\mathcal{A}}$ and a subset B of $Q^{\mathcal{B}}$. The state s' is called the *focus state* or simply *focus*; it is often taken to be the initial state $1'$ but in general can be any state. The subset B is called the *base*. Fix a set T with $B \subseteq T \subseteq Q^{\mathcal{B}}$, called the *target*. Our goal is to give sufficient conditions under which starting from (s', B), we can reach (s', S) for all sets S with $B \subseteq S \subseteq T$. That is, we can reach any state of the concatenation DFA $\mathcal{C}$ in which the first component is the focus and the second component lies between the base and the target.

The idea is to first assume we can reach (s', B), the state consisting of the focus and the base. Now, for $q \in Q$, define a *q-word* to be a word w such that $(s', B)w = (s', B \cup q)$. We can think of this as a word that "adds" the state q to the base B. Our next assumption is that we have a q-word for each state q in the target T. To reach a set S with $B \subseteq S \subseteq T$, we will repeatedly use q-words to add each missing element of S to the base B.

There is a problem with this idea, which we illustrate with an example. Suppose w_p is a p-word and w_q is a q-word, and we want to reach $(s', B \cup \{p, q\})$. Starting from (s', B) we may apply w_p to reach $(s', B \cup p)$. But now if we apply w_q, we reach $(s', B \cup \{pw_q, q\})$. There is no guarantee that we have $pw_q = p$, and in many cases we will not. What we should really do is find a state r such that $rw_q = p$, use an r-word to reach $(s', B \cup r)$, and then apply w_q to reach $(s', B \cup \{p, q\})$. But this idea only works if p has a preimage under w_q, which may not be the case.

We resolve this by making a technical assumption, which ensures that preimages will always exist when we attempt constructions like the above. First, define a *construction set* for the target T to be a set of words consisting of exactly one q-word for each $q \in T$. If W is a construction set for T, we write $W[q]$ for the unique q-word in W.

We say a construction set is *complete* if there is a total order $\prec$ on the target T such that for all $p, q \in T$ with $p \prec q$, the state q has at least one preimage under the unique p-word $W[p]$, and at least one of these preimages lies in T. More formally, whenever $p \prec q$, the set $qW[p]^{-1} = \{s \in Q^{\mathcal{B}} : sW[p] = q\}$ intersects T non-trivially. Our final assumption is that we have a complete construction set for T.

Note that the definition of a q-word depends not only on q, but also on s' and B. Since a construction set for T is a set of q-words, the definition of construction set also depends on s' and B. For simplicity, we omit this dependence on s' and B from the notation for q-words and construction sets.

We summarize the definitions that have just been introduced:

- Fix a state $s' \in Q^{\mathcal{A}}$, called the *focus*, and a set $B \subseteq Q^{\mathcal{B}}$ called the *base*.
- For $q \in Q^{\mathcal{B}}$, a *q-word* is a word w such that $(s', B)w = (s', B \cup q)$.
- Given a *target* set T with $B \subseteq T \subseteq Q^{\mathcal{B}}$, a *construction set* for T is a set of words that contains exactly one q-word for each $q \in T$.
- The unique q-word in a construction set W is denoted by $W[q]$.
- A construction set for T is *complete* if there exists a total order $\prec$ on T such that for all $p, q \in T$ with $p \prec q$, we have

$$qW[p]^{-1} \cap T = \{s \in Q^{\mathcal{B}} : sW[p] = q\} \cap T \neq \emptyset.$$

Now, we state our main theorem, which formalizes the above construction.

Theorem 1. *Fix a state $s' \in Q^{\mathcal{A}}$ and sets B and T such that $B \subseteq T \subseteq Q^{\mathcal{B}}$. If there is a complete construction set for T, then all states of the form (s', S) with $B \subseteq S \subseteq T$ are reachable from (s', B) in $\mathcal{C}$. In particular, if (s', B) itself is reachable, then all states (s', S) with $B \subseteq S \subseteq T$ are reachable.*

Proof. Note that if $B \subseteq S \subseteq T$, we can write $S = R \cup B$ with $R \cap B = \emptyset$ and $R \subseteq T$. Thus it suffices to show that all states of the form $(s', R \cup B)$ with $R \cap B = \emptyset$ and $R \subseteq T$ are reachable from (s', B). We proceed by induction on $|R|$. When $|R| = 0$, the only state of this form is (s', B) itself.

Now suppose every state $(s', R \cup B)$ with $R \cap B = \emptyset$, $R \subseteq T$ and $0 \leq |R| < k$ is reachable from (s', B). We want to show this also holds for $|R| = k$. Let W be a complete construction set for T and let $\prec$ be the corresponding total order on T. Let p be the minimal element of R under $\prec$. Let w be $W[p]$, the unique p-word in W. For all $q \in R \setminus p$, we have $p \prec q$ and thus qw^{-1} contains an element of T (since W is complete).

Construct sets X and Y as follows: starting with $X = \emptyset$, for each $q \in R \setminus p$, choose an element of $qw^{-1} \cap T$ and add it to X. Then set $Y = X \setminus B$. Observe that X is a subset of T of size $|R \setminus p| = k - 1$. Hence Y is a subset of T of size at most $k - 1$ such that $Y \cap B = \emptyset$. It follows by the induction hypothesis that $(s', Y \cup B)$ is reachable from (s', B). But $Y \cup B = X \cup B$, so $(s', X \cup B)$ is reachable from (s', B). By the definition of X, we have $Xw = R \setminus p$. Since w is a p-word, we have $(s', B)w = (s', B \cup p)$, and thus

$$(s', X \cup B)w = (s', Xw \cup B \cup p) = (s', (R \setminus p) \cup B \cup p) = (s', R \cup B).$$

Hence $(s', R \cup B)$ is reachable from (s', B), as required. □

The definition of completeness is somewhat complicated, which makes it difficult to use Theorem 1. Thus, we prove some results giving simpler sufficient conditions for a construction set to be complete.

Before stating our first such result, we introduce some notation. Define $\Sigma_0 = \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}}$. We call Σ_0 the *shared alphabet* of $\mathcal{A}$ and $\mathcal{B}$. The following remark shows that when $\Sigma^{\mathcal{A}} \neq \Sigma^{\mathcal{B}}$, it is important to work exclusively with the shared alphabet when looking for complete construction sets. Of course, in the case where $\Sigma^{\mathcal{A}} = \Sigma^{\mathcal{B}}$ there is nothing to worry about.

Remark 2. A construction set for a non-empty target cannot be complete unless it is a subset of Σ_0^*. To see this, suppose W is a construction set and let $w \in W$. If w contains a letter from $\Sigma^{\mathcal{A}} \setminus \Sigma^{\mathcal{B}}$, then w is not a word over $\Sigma^{\mathcal{B}}$. Recall that if w is not a word over $\Sigma^{\mathcal{B}}$, then $T_w^{\mathcal{B}}$ is defined to be the empty relation. Thus the converse relation $(T_w^{\mathcal{B}})^{-1}$ is also empty, which means qw^{-1} is empty for all q. It follows W cannot be complete. On the other hand, suppose w contains a letter from $\Sigma^{\mathcal{B}} \setminus \Sigma^{\mathcal{A}}$. Then $(s', B)w = (\emptyset, Bw)$. Hence w is not a q-word for any q, and so w cannot be an element of a construction set, which is a contradiction. Thus all words in a complete construction set are words over the shared alphabet Σ_0.

Lemma 1. *Fix $s' \in Q^{\mathcal{A}}$ and sets $B \subseteq T \subseteq Q^{\mathcal{B}}$. Let $x_1, \ldots, x_j$ be words over Σ_0 that act as permutations on T, and let y be an arbitrary word over Σ_0. Choose $x_0 \in \{\varepsilon, x_1, \ldots, x_j\}$. Define $W = \{x_1, x_2, \ldots, x_j\} \cup \{x_0 y, x_0 y^2, \ldots, x_0 y^k\}$. If W is a construction set for T, then it is complete.*

Proof. For $1 \leq i \leq j$, let $w_i = x_i$. For $1 \leq i \leq k$, let $w_{j+i} = x_0 y^i$. Let $\ell = j + k$. Then we have $W = \{w_1, \ldots, w_\ell\}$. Let q_i be the state in T such that $(s', B)w_i = (s', B \cup q_i)$. Define an order $\prec$ on T so that $q_1 \prec q_2 \prec \cdots \prec q_\ell$. We claim this order makes W complete. Notice that $w_r = W[q_r]$, the unique q_r-word in W. Thus we must show that whenever $q_r \prec q_s$, we have $q_s w_r^{-1} \cap T \neq \emptyset$.

Suppose $r < s$ and $r \leq j$. Then $w_r = x_r$ acts as a permutation on T. Thus $q_s w_r^{-1} \cap T$ is non-empty, since $q_s \in T$.

Suppose $r < s$ and $r > j$. Since $s - r > 0$, we can write $w_s = x_0 y^{s-j} = x_0 y^{s-r} y^{r-j} = w_{j+s-r} y^{r-j}$. Thus $(s', B)w_s = (s', B \cup q_{j+s-r})y^{r-j} = (s', B \cup q_s)$. There are two possibilities: $q_{j+s-r} y^{r-j} = q_s$, or $q y^{r-j} = q_s$ for some $q \in B$.

In either case, $q_s (y^{r-j})^{-1} \cap T$ is non-empty. That is, there exists $q \in T$ such that $q y^{r-j} = q_s$. Since x_0 acts as a permutation on T, there exists $p \in T$ such that $p x_0 = q$. Thus $p x_0 y^{r-j} = p w_r = q_s$. It follows that $q_s w_r^{-1} \cap T$ is non-empty, as required. □

Usually, we will use one of the following corollaries instead of Lemma 1 itself.

Corollary 1. *Fix $s' \in Q^{\mathcal{A}}$ and sets $B \subseteq T \subseteq Q^{\mathcal{B}}$. Let x and y be words over Σ_0 such that x acts as a permutation on T. Suppose W is one of the following sets:*

1. $\{y, y^2, \ldots, y^k\}$.
2. $\{\varepsilon, y, y^2, \ldots, y^k\}$.
3. $\{x, xy, xy^2, \ldots, xy^k\}$.
4. $\{\varepsilon, x, xy, xy^2, \ldots, xy^k\}$.

If W is a construction set for T, then it is complete.

Proof. All statements follow easily from Lemma 1:

1. Set $j = 0$.
2. Set $j = 1$ and $x_0 = x_1 = \varepsilon$.
3. Set $j = 1$ and $x_0 = x_1 = x$.
4. Set $j = 2$, $x_1 = \varepsilon$ and $x_0 = x_2 = x$. □

Corollary 2. *Fix $s' \in Q^{\mathcal{A}}$ and sets $B \subseteq T \subseteq Q^{\mathcal{B}}$. Let $W \subseteq \Sigma_0^*$ be a construction set for T.*

1. *If every word in W acts as a permutation on T, then W is complete.*
2. *If there is a word $w \in W$ such that every word in $W \setminus w$ acts as a permutation on T, then W is complete.*

Proof. Both statements follow easily from Lemma 1:

1. Set $k = 0$ in Lemma 1.
2. Set $k = 1$, $x_0 = \varepsilon$ and $y = w$ in Lemma 1. □

In the special case where W contains ε, Corollary 2 admits the following generalization, which we found occasionally useful.

Lemma 2. *Fix $s' \in Q^{\mathcal{A}}$ and sets $B \subseteq T \subseteq Q^{\mathcal{B}}$. Let $W = \{\varepsilon, w_1, \ldots, w_k\}$ be a construction set for T, where $w_1, \ldots, w_k$ are non-empty words over Σ_0. Suppose that for every word $w \in W$, there exists a set S with $T \setminus B \subseteq S \subseteq T$ such that w acts as a permutation on S. Then W is complete.*

Proof. Write $B = \{q_1, \ldots, q_j\}$. Note that ε is a q_i-word for $1 \leq i \leq j$. Thus by the definition of a construction set, ε is the unique q_i-word in W for each $q_i \in B$, that is, $W[q_i] = \varepsilon$ for $1 \leq i \leq j$. In particular, each non-empty word in W is a q-word for some $q \in T \setminus B$. For $1 \leq i \leq k$, let q_{j+i} be the state such that $(s', B)w_i = (s', B \cup q_{j+i})$. Then $T = \{q_1, \ldots, q_{j+k}\}$. Note that $W[q_i] = \varepsilon$ if $1 \leq i \leq j$, and $W[q_i] = w_{i-j}$ if $j + 1 \leq i \leq j + k$.

Define an order $\prec$ on T by $q_1 \prec q_2 \prec \cdots \prec q_{j+k}$. We claim this order makes W complete. Choose $q_r, q_s \in T$ with $q_r \prec q_s$; we want to show that $q_s W[q_r]^{-1} \cap T \neq \emptyset$. Suppose $q_r \in B$. Then $W[q_r] = \varepsilon$, and we have $q_s\varepsilon^{-1} \cap T$ non-empty as required. Now if $q_r \notin B$, then since $q_r \prec q_s$ we also have $q_s \notin B$. In this case, $W[q_r] = w_{r-j}$, which acts as a permutation on some superset S of $T \setminus B$. Since $q_s \in T \setminus B$, it follows that q_s has a preimage under w_{r-j}, and furthermore this preimage lies in T, since S is a subset of T. Thus $q_s w_{r-j}^{-1} \cap T \neq \emptyset$ as required. This proves that W is complete. □

Note that *all words* referred to in the above lemmas and corollaries are words over Σ_0, the shared alphabet of $\mathcal{A}$ and $\mathcal{B}$. When working with automata that have different alphabets, it is important to use only words over the shared alphabet when trying to find a complete construction set.

The following "master theorem" summarizes the results of this section. We have attempted to state this theorem so that it can be cited without having to first define all the notions introduced in this section, such as q-words and construction sets and completeness.

Theorem 2. *Let $\mathcal{A} = (Q^{\mathcal{A}}, \Sigma^{\mathcal{A}}, T^{\mathcal{A}}, i^{\mathcal{A}}, F^{\mathcal{A}})$ and $\mathcal{B} = (Q^{\mathcal{B}}, \Sigma^{\mathcal{B}}, T^{\mathcal{B}}, i^{\mathcal{B}}, F^{\mathcal{B}})$ be DFAs. Let $\mathcal{C} = (Q, \Sigma, T, I, F)$ denote the concatenation DFA of $\mathcal{A}$ and $\mathcal{B}$, as defined in Section 2.2. Let $\Sigma_0 = \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}}$.*

Fix a state $s' \in Q^{\mathcal{A}}$ and sets $B \subseteq T \subseteq Q^{\mathcal{B}}$. Suppose that for each $q \in T$, there exists a word $w_q \in \Sigma_0^*$ such that $(s', B) \xrightarrow{w_q} (s', B \cup q)$ in $\mathcal{C}$. Let $W = \{w_q : q \in T\}$. Suppose that one of the following conditions holds:

1. *There exist words $x, y \in \Sigma_0^*$, where x acts as a permutation on T, such that W can be written in one of the following forms:*
   - *$W = \{y, y^2, \ldots, y^k\}$.*
   - *$W = \{\varepsilon, y, y^2, \ldots, y^k\}$.*
   - *$W = \{x, xy, xy^2, \ldots, xy^k\}$.*
   - *$W = \{\varepsilon, x, xy, xy^2, \ldots, xy^k\}$.*
2. *Every word in W acts as a permutation on T.*
3. *There exists $w \in W$ such that every word in $W \setminus w$ acts as a permutation on T.*
4. *W contains ε, and for every non-empty word $w \in W$, there exists a set S such that $T \setminus B \subseteq S \subseteq T$ and w acts as a permutation on S.*
5. *There exists a total order $\prec$ on T such that for all $p, q \in T$ with $p \prec q$, the set $qw_p^{-1} = \{s \in Q^{\mathcal{B}} : s \xrightarrow{w_p} q\}$ contains an element of T.*

If one of the above conditions holds, then every state of the form (s', X) with $B \subseteq X \subseteq T$ is reachable from (s', B) in $\mathcal{C}$.

To close this section, we give two examples of how our results can be applied. Many more examples can be found in [11].

Theorem 3 (Yu, Zhuang and Salomaa, 1994 [20]). *Define $\mathcal{A}$ and $\mathcal{B}$ as follows:*

	a	b	c	*Final States*
$\mathcal{A}$:	$(1', \ldots, m')$	$(Q^{\mathcal{A}} \to 1')$	id	$\{m'\}$
$\mathcal{B}$:	id	$(1, \ldots, n)$	$(Q^{\mathcal{B}} \to 2)$	$\{n\}$

Then $\mathcal{C}$ has $(m-1)2^n + 2^{n-1}$ reachable and pairwise distinguishable states.

Proof. The initial state of $\mathcal{C}$ is $(1', \emptyset)$. For $k \leq n-2$ we have

$$(1', \emptyset) \xrightarrow{a^m} (1', 1) \xrightarrow{b^k} (1', 1+k).$$

It follows that $\{a^m, a^m b, \ldots, a^m b^{n-1}\}$ is a construction set for $Q^{\mathcal{B}}$ (with $s' = 1'$ and $B = \emptyset$). By Corollary 1, it is complete (taking $x = a^m$ and $y = b$). Hence all states $(1', S)$ with $S \subseteq Q^{\mathcal{B}}$ are reachable. We can reach (q', S) for $q' \neq m'$ and $(m', S \cup 1)$ by words in a^*.

Let (p', S) and (q', T) be distinct states of $\mathcal{C}$. If $S \neq T$, let r be a state in the symmetric difference of S and T. Then b^{n-r} distinguishes the states. If $S = T$ and $p' < q'$, then $ca^{m-q}b^{n-2}$ distinguishes the states. □

Theorem 4 (Maslov, 1970 [16]). *Define $\mathcal{A}$ and $\mathcal{B}$ as follows:*

	a	b	*Final States*
$\mathcal{A}$:	$(1', \ldots, m')$	id	$\{m'\}$
$\mathcal{B}$:	$(n-1, n)$	$(\binom{n-1}{1} q \to q+1)$	$\{n\}$

Then $\mathcal{C}$ has $(m-1)2^n + 2^{n-1}$ reachable and pairwise distinguishable states.

Proof. The initial state is $(1', \emptyset)$. We have

$$(1', \emptyset) \xrightarrow{a^m} (1', 1) \xrightarrow{b^k} (1', 1+k).$$

Thus $\{a^m, a^m b, a^m b^2, \ldots, a^m b^{n-1}\}$ is a construction set for $Q^{\mathcal{B}}$ (with $s' = 1'$ and $B = \emptyset$). By Corollary 1, it is complete. Hence $(1', S)$ is reachable for all $S \subseteq Q^{\mathcal{B}}$. We can reach (q', S) for $q' \neq m'$ and $(m', S \cup 1)$ by words in a^*.

Let (p', S) and (q', T) be distinct states of $\mathcal{C}$. If $S \neq T$, let r be a state in the symmetric difference of S and T. Then b^{n-r} distinguishes the states. If $S = T$ and $p' < q'$, by b^n we reach (p', n) and (q', n). Then by a^{m-q} we reach $((p+m-q)', na^{m-q})$ and $(m', na^{m-q} \cup 1)$. These states differ in their second component, so they are distinguishable. □

4 Conclusions

We have introduced a new technique for demonstrating the reachability of states in DFAs for the concatenation of two regular languages, and tested this technique in a wide variety of cases. In addition to the examples of Theorems 3 and 4, we demonstrate in [11] that our results are applicable to three other regular language concatenation witnesses ([4, 3, 9]), a star-free witness ([7]), two non-returning witnesses ([5, 12]), a prefix-closed witness ([2]), two suffix-free witnesses ([8, 14]), and three right ideal witnesses ([1, 6, 9]). This suggests our technique is worth considering as an alternative to traditional induction proofs when working with concatenation automata.

Acknowledgements

I thank Jason Bell, Janusz Brzozowski and the anonymous referees for proofreading and helpful comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871.

References

1. Brzozowski, J.A., Davies, S., Liu, B.Y.V.: Most complex regular ideal languages. Discrete Math. Theoret. Comput. Sc. 18(3) (2016), paper #15
2. Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory Comput. Syst. 54, 277–292 (2014)

3. Brzozowski, J.A.: In search of most complex regular languages. Int. J. Found. Comput. Sc. 24(06), 691–708 (2013)
4. Brzozowski, J.A.: Unrestricted state complexity of binary operations on regular languages. In: Câmpeanu, C., Manea, F., Shallit, J. (eds.) DCFS 2016. LNCS, vol. 9777, pp. 60–72. Springer (2016)
5. Brzozowski, J.A., Davies, S.: Most complex non-returning regular languages. In: Pighizzini, G., Câmpeanu, C. (eds.) DCFS 2017. LNCS, vol. 10316, pp. 89–101. Springer (2017)
6. Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theoret. Comput. Sci. 470, 36–52 (2013)
7. Brzozowski, J.A., Liu, B.: Quotient complexity of star-free languages. Int. J. Found. Comput. Sc. 23(06), 1261–1276 (2012)
8. Brzozowski, J.A., Sinnamon, C.: Complexity of left-ideal, suffix-closed and suffix-free regular languages. In: Drewes, F., Martín-Vide, C., Truthe, B. (eds.) LATA 2017. LNCS, vol. 10168, pp. 171–182. Springer (2017)
9. Brzozowski, J.A., Sinnamon, C.: Complexity of right-ideal, prefix-closed, and prefix-free regular languages. Acta Cybernetica 23(1), 9–41 (2017)
10. Câmpeanu, C., Culik, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (eds.) WIA 1999. LNCS, vol. 2214, pp. 60–70. Springer (2001)
11. Davies, S.: A new technique for reachability of states in concatenation automata (2017), `https://arxiv.org/abs/1710.05061`
12. Eom, H.S., Han, Y.S., Jirásková, G.: State complexity of basic operations on non-returning regular languages. Fund. Inform. 144, 161–182 (2016)
13. Han, Y.S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In: Ésik, Z., Fülöp, Z. (eds.) AFL 2009. pp. 99–115. Institute of Informatics, University of Szeged, Hungary (2009)
14. Han, Y.S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410(27-29), 2537–2548 (2009)
15. Jirásková, G., Krausová, M.: Complexity in prefix-free regular languages. In: McQuillan, I., Pighizzini, G., Trost, B. (eds.) DCFS 2010. pp. 236–244. University of Saskatchewan (2010)
16. 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
17. Nicaud, C.: Average state complexity of operations on unary automata. In: Kutyłowski, M., Pacholski, L., Wierzbicki, T. (eds.) MFCS 1999. pp. 231–240. Springer (1999)
18. Pighizzini, G., Shallit, J.: Unary language operations, state complexity and Jacobsthal's function. International Journal of Foundations of Computer Science 13(01), 145–159 (2002)
19. Yu, S.: State complexity of regular languages. J. Autom. Lang. Comb. 6, 221–234 (2001)
20. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theor. Comput. Sci. 125(2), 315–328 (1994)