Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University

Parallel decompositions of finite automata

Master Thesis

Peter Gaži

Advisor:
Prof. RNDr. Branislav Rovan, PhD.

Bratislava, 2006

I hereby declare that I wrote this thesis by myself, only with the help of the referenced literature, under the careful supervision of my thesis advisor.

. . . .

Acknowledgements

I would like to thank to my advisor prof. Branislav Rovan for his invaluable guidance and many useful suggestions during my work on this thesis.

A special thank you goes to my family for their support and care, and to my friends for being such symbols of hope and determination.

Abstract

We have studied the possibilities of parallel decomposition of a deterministic finite automaton into a pair of automata such that they are both simpler then the original one, but the results of their independent computations on any input word can determine the result of computation of the original automaton. We have used the results describing the decompositions of sequential machines, and also defined several new kinds of decomposition. Then we have proved some conditions for existence of such decompositions and inspected relationships between them. We have also studied the classes of undecomposable and perfectly decomposable languages and we have shown that there exist automata for most degrees of decomposability from the interval given by these two boundaries.

Keywords. deterministic finite automaton, parallel decomposition, decomposition of state behavior

Contents

1 Introduction

Deterministic finite automaton is a simple computational model with a wide range of practical applications, including hardware design, natural language processing, model checking and many others.

The notion of finite state automaton was first introduced by McCulloch and Pitts [1] to model processes in neural cells. Since then, many formalizations were developed and studied. They can be divided into two major groups: transducers and acceptors.

The most widespread models of a transducer are the *Mealy type sequential machines* and their special case, the *Moore type sequential machines*. These were first formalized by E. F. Moore in [2], and according to Moore's formulation, the output of a machine was dependent only on its state. This concept was generalized by G. H. Mealy in [3], the output of his machines was dependent also on the last input symbol. In general, all transducers read a sequence of input symbols and generate an output sequence depending also on their internal state, which is modified during the computation.

On the other hand, acceptors do not produce any output sequence. They just read an input sequence (a word) and modify their internal state accordingly. After reading the whole input, an acceptor either accepts or rejects the word, depending on the state in which his computation had finished. Thus each acceptor defines a language as the set of all words it accepts. One of the most comprehensive resources on acceptors and the class of languages they can recognize is a book by J. E. Hopcroft and J. D. Ullman [4]. The model that is in the center of scope of this thesis, deterministic finite automaton, is the simplest type of acceptor studied.

For the first type of finite state machines — the transducers — there were studied many ways of composition of simpler machines into more complex ones. Some of the results can be found in books [5] and [6]. However, most of these compositions involve some communication between the parts of the composed machine during the computation. Also, since these are compositions of transducers, they do not take acceptance into account.

For a deterministic finite automaton, it is an interesting question to ask whether it can be decomposed into a pair of automata such that they are both simpler then the original one, but the results of their independent computations on any input word can determine the result of the computation of the original automaton. If we take accepting or rejecting of the input word as a result, we see that this could be solved by decomposing an automaton into two simpler automata such that the intersection of the languages they accept is the language accepted by the original automaton. This question is equivalent to the question of existence of a simpler *advisor* language such that if we knew that all input words are from this language, then the language of the original automaton could be recognized by some simpler automaton.

Definition 1.1. *Let L_1 be a language, let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Language accepted by the automaton A* with advisor *L_1 is the language*

$$L(A, L_1) = \{w \in L_1 | (q_0, w) \vdash_A^* (q, \varepsilon) \wedge q \in F\}.$$

An intuitive measure of the simplicity of a deterministic finite automaton we shall use is the number of its states. This notion of advisor is the basic motivation behind our effort.

In this thesis, we shall inspect the conditions for existence and properties of various kinds of parallel decomposition of deterministic finite automata.

In Section 2, we shall recall some well-known facts and introduce the notation used later.

Section 3 explores the properties of behavior decompositions of DFA, which come out of the concept of *parallel decomposition of state behavior* defined for sequential machines in [7]. We prove a necessary and sufficient condition for existence of these decompositions, relate it to the concept of advisors and then outline some reasons that can prevent the existence of such decomposition.

In Section 4, we define new types of decomposition by posing requirements that such decomposition should satisfy – these requirements describe what we should be able to determine about the result of the computation of the original automaton, knowing the results of computation of both automata in its decomposition. We also derive some conditions for existence of these new decompositions and inspect relationships between these new decompositions and the behavior ones.

We define the classes of all regular languages whose minimal automata are undecomposable for each type of decomposition, and study the properties of these classes in Section 5. We show that the undecomposability of the minimal automaton may depend on the input alphabet of this automaton, then we exhibit some undecomposable languages and use them to inspect the closure properties of the classes defined.

In the last section we study the degree to which certain automata are decomposable. From the previous sections we know that there are undecomposable and well-decomposable automata, but now we also exhibit automata that can be decomposed, but each automaton in the decomposition has to have only one state less than the original automaton. Then we show that for some types of decomposition most of the values in the interval between undecomposable and well-decomposable automata can be achieved.

2 Preliminaries and Notation

In order to unify the notation and recall the concepts used throughout the thesis, we shall introduce a few definitions and state some well-known results.

By letter Σ, we shall usually denote an *alphabet*, i.e. a finite set of symbols. Finite sequences of symbols from Σ are called *words* over alphabet Σ and we shall usually denote them by letters u, v, w, ... The length of such sequence w is denoted $|w|$ and called *length* of word w, the word with length 0 is represented by the symbol ε. The number of occurrences of a given letter a in a word w is denoted by $\#_a(w)$. The set of all words over alphabet Σ is denoted by Σ^*.

A *language* is any set of words over Σ and is usually denoted by L, i.e., $L \subseteq \Sigma^*$. Since any language L can be considered over many different alphabets, by Σ_L we shall denote the minimal alphabet such that $L \subseteq \Sigma_L^*$. More on these basic concepts can be found in [4].

Now let us define deterministic finite automaton, a simple computational model that will be in the center of our interest.

Definition 2.1. *A* deterministic finite automaton *(DFA) is a quintuple $(K, \Sigma, \delta, q_0, F)$, such that K is a finite set of states, Σ is a finite input alphabet, $q_0 \in K$ is an initial state, $F \subseteq K$ is a set of accepting states and $\delta\colon K \times \Sigma \to K$ is a transition function.*

Notation 2.1. Let $\overline{\delta}\colon K \times \Sigma^* \to K$ denote the extension of the transition function δ onto the set of all input words, such that $\overline{\delta}(q, \varepsilon) = q$ and $\overline{\delta}(q, au) = \overline{\delta}(\delta(q, a), u)$ for all $q \in K$, $a \in \Sigma$ and $u \in \Sigma^*$.

Definition 2.2. *A* configuration *of a DFA $A = (K, \Sigma, \delta, q_0, F)$ is a pair $(q, w) \in K \times \Sigma^*$, where q is the current state and w is the unprocessed part of the input word.*

Definition 2.3. *A* computation step *of a DFA $A = (K, \Sigma, \delta, q_0, F)$ is a binary relation $\vdash_A$ on the set of all configurations of A defined as follows: $(p, au) \vdash_A (q, u) \Leftrightarrow \delta(p, a) = q$. We shall often write $\vdash$ instead of $\vdash_A$ when A is understood.*

Definition 2.4. *A* language accepted by a DFA *$A = (K, \Sigma, \delta, q_0, F)$ is the set $L(A) = \{w \in \Sigma^* | (q_0, w) \vdash^* (q, \varepsilon) \wedge q \in F\}$, where $\vdash^*$ is the reflexive and transition closure of $\vdash$.*

The transition function of a DFA is often described by a *transition diagram*, which is an oriented graph such that each vertex corresponds to one state of the DFA and each edge represents a transition. More precisely, an edge from vertex p to vertex q labeled a represents the equality $\delta(p, a) = q$. To keep the diagrams as simple as possible, we shall not depict transitions that start and end in the same state. Since the transition function $\delta(p, a)$ is defined for all states p and all letters a, if the diagram does not contain an edge beginning in the vertex p and labeled a, it means that $\delta(p, a) = p$.

Notation 2.2. By $\mathcal{A}_k$ we shall denote the set of all deterministic finite automata having at most k states. The set of all languages that can be accepted by an automaton from $\mathcal{A}_k$ shall be denoted by $\mathcal{R}_k$.

The class of all languages accepted by deterministic finite automata is the well-known class of *regular languages*. These languages have many interesting properties, one of the most characteristic and useful from the automaton point of view is known as the Myhill-Nerode Theorem.

Theorem 2.1 (Myhill-Nerode). *Let $L \subseteq \Sigma^*$ be a language. The following statements are equivalent:*

1. *L is a regular language.*
2. *L is a union of some equivalence classes of some right-invariant equivalence relation with finite index.*
3. *Relation R_L defined by $uR_Lv \Leftrightarrow (\forall x \in \Sigma^*; ux \in L \Leftrightarrow vx \in L)$ is an equivalence relation with finite index.*

Proof of this theorem can be found for example in [4].

Corollary 2.2. *If L is a regular language, then there exists a unique DFA $A = (K, \Sigma, \delta, q_0, F)$ accepting this language and having the minimum possible number of states. Moreover, if $A' = (K', \Sigma, \delta', q_0', F')$ is a DFA such that $L(A') = L$, then there exists a mapping $f\colon K' \to K$ such that it holds $(\forall w \in \Sigma^*)$; $f(\overline{\delta}'(q_0', w)) = \overline{\delta}(q_0, w)$.*

So for every regular language there exists a unique minimal automaton. There is also a well-known efficient algorithm for minimization of a given DFA working in $O(n^2)$ time. Its improvement working in $O(n \log n)$ time is due to Hopcroft and can be found in [8]. An incremental modification of the minimization algorithm can be found in [9].

Another concept that we shall occasionally use in this thesis is the algebraic structure called *lattice*. We now give a brief definition of a lattice using the well-known terms of order theory, further details and definitions of the prerequisite terms can be found in [10].

Definition 2.5. *Let* $(L, \preceq)$ *be a partially ordered set. If there exists a least upper bound (join, denoted by* $x \vee y$*) and a greatest lower bound (meet, denoted by* $x \wedge y$*) for all pairs of elements* $x, y \in L$*, then we call* $(L, \preceq)$ *a* lattice.

A special case of a lattice is a *distributive* lattice, defined as follows.

Definition 2.6. *Let* $(L, \preceq)$ *be a lattice with the least upper bound* $\vee$ *and the greatest lower bound* $\wedge$*.* L *is* distributive*, if the following (equivalent) identities hold:*

$$\begin{aligned} x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \\ x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \end{aligned}$$

for all elements $x, y, z \in L$.

3 Behavior Decompositions of DFA

Let us take a closer look at the problem of decomposing a given deterministic finite automaton into two "simpler" automata, such that they are in some sense able to substitute the original automaton. As we have already mentioned, our only measure of simplicity of a DFA will be the number of its states.

Our first approach to this decomposition problem is inspired by the theory of sequential machines. Moore and Mealy type sequential machines are abstract computational models similar to deterministic finite automata. The important difference is that sequential machines do not have any specified initial states nor final states, only a transition function that determines the changes of the state of a sequential machine based on some input. Compared to deterministic finite automata, sequential machines also have an output function. As we only want to exploit the ideas behind the decompositions of sequential machines, we do not need to introduce the formal definition of this model.

The possibilities of decomposition of sequential machines were intensively studied and the results can be found for example in [5], [6] and [7]. We shall use the concept of a *parallel decomposition of state behavior* of sequential machines, which is mentioned in [7]. We shall modify this decomposition to fit the formalism and purpose of deterministic finite automata (i.e., to accept formal languages) without loosing the connection with the strongly related and useful concept of *S.P.partitions.*

Let us begin by formally defining this decomposition modified for finite automata, and then show some of its properties.

Definition 3.1. *A DFA* $A' = (K', \Sigma, \delta', q_0', F')$ *is said to* realize the state behavior *of a DFA* $A = (K, \Sigma, \delta, q_0, F)$ *if there exists an injective mapping* $\alpha \colon K \to K'$ *such that*

(i) $(\forall a \in \Sigma)(\forall q \in K); \delta'(\alpha(q), a) = \alpha(\delta(q, a))$.

(ii) $\alpha(q_0) = q_0'$

Moreover, A' *is said to* realize the state and acceptance behavior *of* A*, if in addition the following property holds:*

(iii) $(\forall q \in K); \alpha(q) \in F' \Leftrightarrow q \in F$.

Definition 3.2. *The* parallel connection *of two DFA* $A_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ *and* $A_2 = (K_2, \Sigma, \delta_2, q_2, F_2)$ *is the DFA* $A = A_1 || A_2 = (K_1 \times K_2, \Sigma, \delta, (q_1, q_2),$ $F_1 \times F_2)$ *such that* $\delta((p_1, p_2), a) = (\delta_1(p_1, a), \delta_2(p_2, a))$.

Definition 3.3. *A pair of deterministic finite automata* (A_1, A_2) *is a* state behavior (SB-) decomposition *of a DFA* A *if* $A_1||A_2$ *realizes the state behavior of* A. *The pair* (A_1, A_2) *is an* acceptance and state behavior (ASB-) decomposition *of* A *if* $A_1||A_2$ *realizes the state and acceptance behavior of* A. *The decomposition is* nontrivial *if both* A_1 *and* A_2 *have fewer states than* A.

The following theorem shows that existence of a nontrivial ASB-decomposition of some automaton implies the existence of a nontrivial advisor for this problem.

Theorem 3.1. *Let* A *be a DFA, let* n *denote the number of states of* A. *If there exists a nontrivial ASB-decomposition of* A, *then there exists an advisor* $L_1 \in \mathcal{R}_{n-1}$ *and an automaton* $A_2 \in \mathcal{A}_{n-1}$ *such that* $L(A) = L(A_2, L_1)$.

Proof. Let (A_1, A_2) be a nontrivial ASB-decomposition of A. The decomposition (A_1, A_2) is nontrivial, so both statements $L_1 = L(A_1) \in \mathcal{R}_{n-1}$ and $A_2 \in \mathcal{A}_{n-1}$ hold. It remains to prove that $L(A) = L(A_2, L_1)$.

"$\subseteq$": Consider $w \in L(A)$. The computation of A on the word w is accepting and so is the computation of $A_1||A_2$ on the word w, because $A_1||A_2$ realizes the state behavior of A. From the definition of parallel connection it follows that the accepting computation of $A_1||A_2$ on w determines accepting computations of both A_1 and A_2 on w. Hence $w \in L_1$ and $w \in L(A_2)$, which implies $w \in L(A_2, L_1)$.

"$\supseteq$": The proof of the reverse containment is similar. □

The reverse implication does not hold, as we shall be able to show later.

Now we derive a necessary and sufficient condition for the existence of a nontrivial SB-decomposition and a nontrivial ASB-decomposition of a given DFA. At first, we need some definitions:

Definition 3.4. *A partition* π *on a finite set* M *is a set* $\{M_1, M_2, \ldots, M_k\}$ *of nonempty mutually disjoint subsets of* M *such that* $M = \bigcup_{i=1}^{k} M_i$.

Note that each partition on a set M defines an equivalence relation on M and vice versa. We shall call the sets M_i blocks or equivalence classes.

Definition 3.5. *A partition* π *on a set of states of a deterministic finite automaton* $A = (K, \Sigma, \delta, q_0, F)$ *has* substitution property *(S.P.), if*

$$\forall p, q \in K; \quad p \equiv_\pi q \Rightarrow (\forall a \in \Sigma; \delta(p, a) \equiv_\pi \delta(q, a))$$

Definition 3.6. *Let* π_1, π_2 *be partitions on a given set* M, *then* *(i)* $\pi_1 \cdot \pi_2$ *is a partition on* M *such that* $a \equiv_{\pi_1 \cdot \pi_2} b \Leftrightarrow a \equiv_{\pi_1} b \wedge a \equiv_{\pi_2} b$,

(ii) $\pi_1 + \pi_2$ *is a partition on* M *such that* $a \equiv_{\pi_1 + \pi_2} b$ *if and only if there exists a sequence* $a = a_0, a_1, a_2, \ldots, a_n = b$, *such that* $a_i \equiv_{\pi_1} a_{i+1} \vee a_i \equiv_{\pi_2} a_{i+1}$ *for all* $i \in \{0, \ldots, n-1\}$,

(iii) $\pi_1 \preceq \pi_2$ *if it holds* $(\forall x, y \in M);\quad x \equiv_{\pi_1} y \Rightarrow x \equiv_{\pi_2} y$.

It is obvious that the relation defined in the previous definition is a partial order.

Partitions on a set of states of some DFA form an interesting structure, as stated by the following theorem (its proof can be found in [7]).

Theorem 3.2. *The set of all partitions on a given set (accompanied with the partial order* $\preceq$*, join realized by* $+$ *and meet realized by .) forms a lattice. The set of all S.P. partitions on the set of states of a given deterministic finite automaton forms a sublattice of the lattice of all partitions on this set.*

Notation 3.1. We shall denote the trivial partitions $\{\{q_0\}, \{q_1\}, \ldots, \{q_n\}\}$ and $\{\{q_0, q_1, \ldots, q_n\}\}$ by symbols 0 and 1, respectively. Let $|\pi|$ denote the number of blocks (equivalence classes) of a partition π.

Notation 3.2. Let π be a partition on a set S, let x be an element of S. The symbol $[x]_\pi$ shall denote the equivalence class of π containing x, i.e., $[x]_\pi = \{y \in S | y \equiv_\pi x\}$.

The original result from [7] handles the decomposition of state behavior of sequential machines (which have no final states). Since we are interested in decompositions of deterministic finite automata, we also have to take the final states into consideration. In our model of advisor, a word is accepted if and only if it belongs to the advisor and the advised automaton accepts it. In case the advisor language is also recognized by a DFA (as in this case), the advised automaton does not know in which of its final states the computation of the advisor has ended, it only "knows" that the word has been accepted. So any combination of a final state on the advisor and on the advised automaton lead to the acceptance of the word, therefore we introduce the following additional property that has to be verified when trying to find a decomposition.

Definition 3.7. *The partitions* $\pi_1 = \{R_1, \ldots, R_k\}$ *and* $\pi_2 = \{S_1, \ldots, S_l\}$ *on a set of states of a DFA* $A = (K, \Sigma, \delta, q_0, F)$ *are said to* separate the final states *of* A *if there exist indices* $i_1, \ldots, i_r$ *and* $j_1, \ldots, j_s$ *such that it holds* $(R_{i_1} \cup \ldots \cup R_{i_r}) \cap (S_{j_1} \cup \ldots \cup S_{j_s}) = F$.

Now we can prove the necessary and sufficient conditions for the existence of both defined types of decomposition. The first part of the proof is based on the proof of the analogous property for sequential machines in [7].

Theorem 3.3. *A deterministic finite automaton $A = (K, \Sigma, \delta, q_0, F)$ has a nontrivial SB-decomposition if and only if there exist two nontrivial S.P. partitions π_1 and π_2 on the set of states of A such that $\pi_1 \cdot \pi_2 = 0$. This decomposition is an ASB-decomposition if and only if these partitions separate the final states of A.*

Proof. "⇒": Let $A_1||A_2 = (K_1 \times K_2, \Sigma, \delta', (q_1, q_2), F_1 \times F_2)$, let (A_1, A_2) be a nontrivial SB-decomposition of a given DFA A, with α being the corresponding mapping. We shall define the partitions π_1, π_2 on the set of states of A as follows:

$$\begin{aligned} p \equiv_{\pi_1} q \quad &\Leftrightarrow \quad p_1 = q_1 \\ p \equiv_{\pi_2} q \quad &\Leftrightarrow \quad p_2 = q_2 \end{aligned}$$

where $\alpha(p) = (p_1, p_2)$ and $\alpha(q) = (q_1, q_2)$.

Let p and q be states of A such that $p \equiv_{\pi_1} q$ and let $a \in \Sigma$. Then $\alpha(p) = (p_1, p_2)$ and $\alpha(q) = (p_1, q_2)$. As $A_1||A_2$ is a parallel behavior decomposition, we have

$$\begin{aligned} \alpha(\delta(p, a)) = \delta'(\alpha(p), a) = (\delta_1(p_1, a), \delta_2(p_2, a)) \\ \alpha(\delta(q, a)) = \delta'(\alpha(q), a) = (\delta_1(p_1, a), \delta_2(q_2, a)) \end{aligned}$$

where δ_i denotes the transition function of A_i. As we can see, the first components are equal again, thus $\delta(p, a) \equiv_{\pi_1} \delta(q, a)$, which proves that π_1 is an S.P. partition. An analogous proof holds for π_2.

Let p and q be states of A such that $p \equiv_{\pi_1} q$ and $p \equiv_{\pi_2} q$. This means that $\alpha(p) = (p_1, p_2) = \alpha(q)$ and (since α is injective) $p = q$. Hence $\pi_1 \cdot \pi_2 = 0$.

As $A_1||A_2$ is a nontrivial decomposition, it holds that $|K_i| < |K|$, and therefore $|\pi_i| < |K|$, $i \in \{1, 2\}$. But $\pi_1 \cdot \pi_2 = 0$, hence $|\pi_i| > 1$, which together means that π_1 and π_2 are both nontrivial.

To complete the proof of this implication, it remains to prove that if (A_1, A_2) is an ASB-decomposition, then π_1 and π_2 separate the final states of A. From each partition, we shall take exactly those blocks that contain at least one final state of A. By union of all such blocks in each of the partitions we obtain two sets, and then we prove that the intersection of these sets is exactly the set F, as required by Definition 3.7. Formally, we want to prove that

$$\left(\bigcup_{q\in F}[q]_{\pi_1}\right)\cap\left(\bigcup_{q\in F}[q]_{\pi_2}\right)=F$$

The first set contains all states that are equivalent to some final state modulo π_1, therefore the first component of their α-image is some final state in A_1. For the same reason, the second set contains all states such that the second component of their α-image is some final state in A_2. Hence after performing the intersection, there remain exactly the states q that satisfy the condition $\alpha(q) \in F_1 \times F_2$, and these are (according to (iii) of Definition 3.1) exactly the final states of A.

"$\Leftarrow$": Let π_1 and π_2 be the given nontrivial S.P. partitions on the set of states of A such that $\pi_1 \cdot \pi_2 = 0$. We shall construct two automata A_1 and A_2 having states corresponding to blocks of these partitions and show that (A_1, A_2) is a nontrivial SB-decomposition of A. Let $A_i = (\pi_i, \Sigma, \delta_i, [q_0]_{\pi_i}, \{[q]_{\pi_i} | q \in F\})$ be a DFA with δ_i function defined by $\delta_i([q]_{\pi_i}, a) = [\delta(q,a)]_{\pi_i}$, $i \in \{1,2\}$ (this definition does not depend on the choice of q since π_i is an S.P. partition). To show that $A_1||A_2$ is a realization of A, we define the mapping $\alpha\colon K \to K_1 \times K_2$ by $\alpha(q) = ([q]_{\pi_1}, [q]_{\pi_2})$. Since $\pi_1 \cdot \pi_2 = 0$, α is injective. It remains to prove that the two conditions (i) and (ii) from Definition 3.1 hold.

(i) This can be verified by a simple computation:

$$\begin{aligned}\delta'(\alpha(q),a) &= \delta'(([q]_{\pi_1},[q]_{\pi_2}),a) = (\delta_1([q]_{\pi_1},a),\delta_2([q]_{\pi_2},a)) = \\ &= ([\delta(q,a)]_{\pi_1},[\delta(q,a)]_{\pi_2}) = \alpha(\delta(q,a))\end{aligned}$$

The first equality comes from the definition of the mapping α, the second from the definition of a parallel connection of two automata, the third from the definition of δ_i and the last one from the definition of α again.

(ii) This comes directly from the definitions of A_1, A_2 and α.

To complete the proof, we have to show that if π_1 and π_2 separate the final states of A, then also the property (iii) from Definition 3.1 is satisfied.

(iii) If $q \in F$, then $\alpha(q) = ([q]_{\pi_1}, [q]_{\pi_2})$, which by the definition of A_1 and A_2 is an element of $F_1 \times F_2$. This, by the definition of the parallel connection, implies that $\alpha(q) \in F'$. On the other hand, if $\alpha(q) \in F'$, by the definition of the parallel connection it implies that $\alpha(q) \in F_1 \times F_2$. That means that q is equivalent to some final state of A modulo π_1 and also to some (possibly different) final state of A modulo π_2. Therefore q appears in both sets intersected in Definition 3.7 which implies that it is a member of F, since π_1 and π_2 separate the final states of A.

$\square$

If a DFA has only one accepting state then the condition of separating final states is trivially met, hence we can formulate the following corollary.

Corollary 3.4. *A deterministic finite automaton $A = (K, \Sigma, \delta, q_0, \{q_F\})$ has a nontrivial ASB-decomposition if and only if there exist two nontrivial S.P. partitions π_1 and π_2 on the set of states of A such that $\pi_1 \cdot \pi_2 = 0$.*

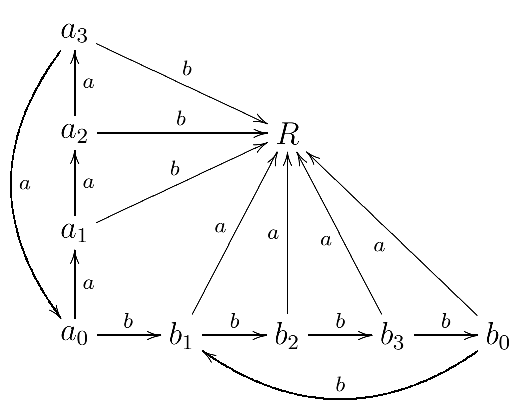
Since [7] describes an easy way to generate all S.P. partitions of a given sequential machine (and this can be directly applied to finite automata), we can easily find all SB- (and ASB-) decompositions of a given automaton.

Now we are able to show by a simple example that the converse of Theorem 3.1 does not hold.

Example 3.1. Consider the language $L = \{a^{4k}b^{4l} | a \geq 0, b \geq 1\}$. The minimal automaton accepting this language is

$$A = (\{a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3, R\}, \{a, b\}, \delta, a_0, \{b_0\})$$

with the transition function defined by the following transition diagram:



(recall that we shall always assume that all the transitions not depicted in the transition diagram start and end in the same state).

The idea behind this automaton is simple, the purpose of the a_i states is to count the a symbols (modulo 4), the b_i states count the b symbols (again modulo 4) and R is a reject state. This automaton does not have a nontrivial SB-decomposition (which implies no ASB-decompositions) as can be easily verified using Theorem 3.3 by listing all S.P. partitions. But if we use $L_1 = \{a^{4k}b^l | k \geq 0, l \geq 1\} \in \mathcal{R}_6$ as an advisor, we can accept L with a DFA having only 4 states and accepting language $\{w | \#_b(w) = 4l, l \geq 0\}$, hence there exists a nontrivial advisor for L.

The reason why the automaton from the previous example has no nontrivial SB-decomposition is that any S.P. partition on the set of its states that would have all a_i states in the same block (the corresponding automaton would not count the symbols a), would also have to contain all the b_i and R states in the same block and thus would not be able to count the b symbols, either. This is caused by the "terminal" character of the state R and it is in general stated by the following lemma. At first, we need a few definitions:

Definition 3.8. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFA, let $q \in K$, $a \in \Sigma$. We shall call the set $\{p \in K | \exists n \in \mathbb{N} \cup \{0\}; \overline{\delta}(q, a^k) = p\}$ the* a-tail of the state q *and denote it by $\langle q \rangle_a$. By* tail *of the state q, we shall mean the union of all a-tails, $a \in \Sigma$, i.e., $\langle q \rangle = \bigcup_{a \in \Sigma} \langle q \rangle_a$.*

Definition 3.9. *Let K be the set of states of some DFA A and let a be in Σ. We shall call a state $q \in K$* a-terminal *if $\langle q \rangle_a = \{q\}$. We shall call a state $q \in K$* terminal *if $\langle q \rangle = \{q\}$.*

Lemma 3.5. *Let π be an S.P.-partition on the set of states of a given DFA $A = (K, \Sigma, \delta, q_0, F)$, let $p, q \in K$, $p \equiv_\pi q$, let $a \in \Sigma$. If q is a-terminal, then $\forall p' \in \langle p \rangle_a; p' \equiv_\pi q$.*

Proof. Let $p' \in \langle p \rangle_a$, then there exists the minimal $n \in \mathbb{N} \cup \{0\}$ such that $\overline{\delta}(p, a^n) = p'$. We shall prove the lemma by induction on n. For $n = 0$, $p' = p$ and the equation $p \equiv_\pi q$ holds. Now suppose $n > 0$, then by the induction hypothesis $\overline{\delta}(p, a^{n-1}) \equiv_\pi q$, and π has substitution property, thus $\overline{\delta}(p, a^n) \equiv_\pi \delta(q, a) = q$, which completes the proof. □

Corollary 3.6. *Let π be an S.P.-partition on the set of states of a given DFA $A = (K, \Sigma, \delta, q_0, F)$, let $p, q \in K$, $p \equiv_\pi q$ and $a \in \Sigma$. If q is terminal, then $\forall p' \in \langle p \rangle; p' \equiv_\pi q$.*

4 Other Types of Decomposition

Now we shall try a different approach to the problem of DFA decomposition. We shall define exactly what it means that the smaller automata forming the decomposition can substitute the original automaton. It is obvious that if the decomposition is to be useful, the results of computations of the automata forming such a decomposition have to indicate what would be the result of the computation of the original automaton. We can be interested in knowing in which state the computation of the original automaton would end, or knowing just whether the original automaton would accept the word or not. We could also have access to the information in which states the smaller automata have finished their computations, or maybe we only know whether they have both accepted the word. Depending on these factors, we obtain different ways of decomposing a DFA.

The first possibility is that we want to decompose a DFA into two simpler automata in such a way that after performing the computations of both of these automata on any given input word, we can identify the state of the original automaton in which it would end its computation from the states in which these computations have ended. This decomposition is formalized by the following definition.

Definition 4.1. *A pair of deterministic finite automata* (A_1, A_2)*, where* $A_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ *and* $A_2 = (K_2, \Sigma, \delta_2, q_2, F_2)$*, forms a* state-identifying decomposition *(SI-decomposition) of a DFA* $A = (K, \Sigma, \delta, q_0, F)$*, if there exists a mapping* $\beta \colon K_1 \times K_2 \to K$*, such that it holds*

$$(\forall w \in \Sigma^*); \quad \beta(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) = \overline{\delta}(q_0, w)$$

If $|K_1| < |K|$ *and* $|K_2| < |K|$*, this decomposition is nontrivial.*

Another possible requirement on the decomposition could be that we want any input word to be accepted by both smaller automata if and only if it would be accepted also by the original automaton. This way, we could perform the computation of both smaller automata on any given input word and then decide whether this word would be accepted by the original automaton based only on the information whether it was accepted or rejected by the decomposition automata. This decomposition is formalized as follows.

Definition 4.2. *A pair of deterministic finite automata* (A_1, A_2)*, where* $A_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ *and* $A_2 = (K_2, \Sigma, \delta_2, q_2, F_2)$*, forms an* acceptance-identifying decomposition *(AI-decomposition) of a DFA* $A = (K, \Sigma, \delta, q_0, F)$*, if it holds* $L(A) = L(A_1) \cap L(A_2)$*. If* $|K_1| < |K|$ *and* $|K_2| < |K|$*, this decomposition is nontrivial.*

It is easy to see that the notion of a nontrivial AI-decomposition is equivalent to the concept of a nontrivial advisor we are interested in. Therefore Theorem 3.1 and Example 3.1 from the previous section also apply to AI-decomposition.

The third – and the weakest – requirement we could pose on a decomposition of a DFA is to require that there must exist a way to determine whether the original automaton would accept some given input word based on knowing the states in which the computations of both decomposition automata have finished.

Definition 4.3. *A pair of deterministic finite automata* (A_1, A_2)*, where* $A_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ *and* $A_2 = (K_2, \Sigma, \delta_2, q_2, F_2)$*, forms a* weak acceptance-identifying decomposition *(wAI-decomposition) of a DFA* $A = (K, \Sigma, \delta, q_0, F)$*, if there exists a relation* $R \subseteq K_1 \times K_2$ *such that it holds*

$$(\forall w \in \Sigma^*);\quad R(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) \Leftrightarrow w \in L(A)$$

If $|K_1| < |K|$ *and* $|K_2| < |K|$*, this decomposition is nontrivial.*

There exists a simple relationship between these types of decomposition.

Lemma 4.1. *If* (A_1, A_2) *is an SI-decomposition of a DFA A, then it is also a wAI-decomposition of A. If* (A_1, A_2) *is an AI-decomposition of a DFA A, then it is also a wAI-decomposition of A.*

Proof. This is a direct consequence of the definitions. □

For minimal automata, a relationship between AI- and SI-decompositions can be obtained.

Theorem 4.2. *Let* $A = (K, \Sigma, \delta, q_0, F)$ *be a minimal deterministic finite automaton, let* (A_1, A_2) *be its AI-decomposition. Then* (A_1, A_2) *is also an SI-decomposition of A.*

Proof. Since (A_1, A_2) is an AI-decomposition of A, by definition it holds that $L(A) = L(A_1) \cap L(A_2)$. Therefore if we use the well-known Cartesian product construction, we obtain the automaton $A_1 || A_2$ such that $L(A_1 || A_2) = L(A)$. Since A is the minimal automaton accepting the language $L(A)$, there exists a mapping $\beta\colon K' \to K$ such that it holds

$$(\forall w \in \Sigma^*);\quad \beta(\overline{\delta}'(q_0', w)) = \overline{\delta}(\beta(q_0'), w)$$

where δ' is the transition function of $A_1 || A_2$, K' is its set of states and q_0' is its initial state. Since $A_1 || A_2$ is a parallel connection (i.e., $K' = K_1 \times K_2$, q_0' is the pair of initial states of A_1 and A_2), it is easy to see that β is in fact exactly the mapping required by the definition of the SI-decomposition. □

For the other decompositions, we can derive the following sufficient conditions that exploit the concept of S.P. partitions.

Theorem 4.3. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton, let π_1 and π_2 be nontrivial S.P. partitions on the set of states of A, such that they separate the final states of A. Then A has a nontrivial AI-decomposition.*

Proof. Let π_1 and π_2 be the given nontrivial S.P. partitions on the set of states of A such that they separate the final states of A. Let $B_1, \ldots, B_k$ and $C_1, \ldots, C_l$ be the blocks of the partitions π_1 and π_2 respectively, such that $(B_1 \cup \ldots \cup B_k) \cap (C_1 \cup \ldots \cup C_l) = F$ (the existence of such blocks is guaranteed since π_1 and π_2 separate the final states of A). We shall construct two automata A_1 and A_2 having states corresponding to blocks of these partitions and show that (A_1, A_2) is a nontrivial AI-decomposition of A. Let

$$\begin{aligned} A_1 &= (\pi_1, \Sigma, \delta_1, [q_0]_{\pi_1}, \{B_1, \ldots, B_k\}) \\ A_2 &= (\pi_2, \Sigma, \delta_2, [q_0]_{\pi_2}, \{C_1, \ldots, C_l\}) \end{aligned}$$

be DFAs with the transition functions δ_i defined by $\delta_i([q]_{\pi_i}, a) = [\delta(q, a)]_{\pi_i}$, $i \in \{1, 2\}$ (this definition does not depend on the choice of q since π_i is an S.P. partition). We now need to prove that $L(A) = L(A_1) \cap L(A_2)$.

"$\subseteq$": Let $w \in L(A)$. Suppose that the computation of A on the word w ends in some accepting state $q_f \in F$. Then, from the construction of A_1 and A_2 it follows that the computation of A_i on the word w ends in the state corresponding to the block $[q_f]_{\pi_i}$ of the partition π_i. Since $q_f \in F$, it must hold $[q_f]_{\pi_1} \in \{B_1, \ldots, B_k\}$ and $[q_f]_{\pi_2} \in \{C_1, \ldots, C_l\}$, hence from the construction of A_i, these blocks correspond to the accepting states in the respective automata. Thus $w \in L(A_i)$ for $i \in \{1, 2\}$, therefore $L(A) \subseteq L(A_1) \cap L(A_2)$.

"$\supseteq$": Now suppose $w \in L(A_1) \cap L(A_2)$, Thus the computation of A_1 on w ends in one of the states $B_1, \ldots, B_k$, which means that the computation of A on w would end in a state from the union of blocks $B_1 \cup \ldots \cup B_k$. Using the same argument for A_2, we get that the computation of A on w would end in a state from $C_1 \cup \ldots \cup C_l$. Since $(B_1 \cup \ldots \cup B_k) \cap (C_1 \cup \ldots \cup C_l) = F$ we obtain that the computation of A ends in an accepting state, hence $w \in L(A)$ and $L(A_1) \cap L(A_2) \subseteq L(A)$.

Since both partitions are nontrivial, so is the AI-decomposition obtained. □

Theorem 4.4. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton, let π_1 and π_2 be nontrivial S.P. partitions on the set of states of A, such that $\pi_1 \cdot \pi_2 \preceq \{F, K - F\}$. Then A has a nontrivial wAI-decomposition.*

Proof. We shall construct A_1 and A_2 corresponding to the partitions π_1 and π_2 as follows: $A_i = (\pi_i, \Sigma, \delta_i, [q_0]_{\pi_i}, \emptyset)$, where $\delta_i([q]_{\pi_i}, a) = [\delta(q, a)]_{\pi_i}$ and $i \in \{1, 2\}$. The definition of the transition function δ_i does not depend on the choice of q as a consequence of the S.P. property of π_i. To show that (A_1, A_2) is a wAI-decomposition of A, we define the relation $R \subseteq \pi_1 \times \pi_2$ by the equivalence $R(B_1, B_2) \Leftrightarrow (B_1 \cap B_2 \subseteq F)$,where B_i is some block of the partition π_i.

Now we need to prove that

$$(\forall w \in \Sigma^*); \quad w \in L(A) \Leftrightarrow R(\overline{\delta}_1([q_0]_{\pi_1}, w), \overline{\delta}_2([q_0]_{\pi_2}, w))$$

Suppose that the computation of A on w ends in some state $p \in K$. From the definition of the transition functions δ_i it follows that the computation of A_i on the word w ends in the state corresponding to the block $[p]_{\pi_i}$, hence

$$R(\overline{\delta}_1([q_0]_{\pi_1}, w), \overline{\delta}_2([q_0]_{\pi_2}, w)) \Leftrightarrow R([p]_{\pi_1}, [p]_{\pi_2})$$

and by the definition of R, we get

$$R(\overline{\delta}_1([q_0]_{\pi_1}, w), \overline{\delta}_2([q_0]_{\pi_2}, w)) \Leftrightarrow [p]_{\pi_1} \cap [p]_{\pi_2} \subseteq F$$

Obviously $p \in [p]_{\pi_1} \cap [p]_{\pi_2}$. Also $[p]_{\pi_1} \cap [p]_{\pi_2}$ is a block of the partition $\pi_1 \cdot \pi_2$ and since $\pi_1 \cdot \pi_2 \preceq \{F, K - F\}$. it must hold that either $[p]_{\pi_1} \cap [p]_{\pi_2} \subseteq F$ or $[p]_{\pi_1} \cap [p]_{\pi_2} \subseteq K - F$. Therefore

$$R(\overline{\delta}_1([q_0]_{\pi_1}, w), \overline{\delta}_2([q_0]_{\pi_2}, w)) \Leftrightarrow p \in F$$

and the proof is complete. □

The ASB-decomposition is a combination of the SB-decomposition and the AI-decomposition, as the next theorem shows.

Theorem 4.5. *Let A be a DFA without unreachable states. (A_1, A_2) is an ASB-decomposition of A if and only if (A_1, A_2) is both an SB-decomposition and an AI-decomposition of A.*

Proof. "⇒" The statement about SB-decomposition comes directly from the definitions and the statement about AI-decompositions is a consequence of Theorem 3.3 and Theorem 4.3. This implication holds without the assumption of reachability of all states of A.

"⇐" Let (A_1, A_2) be an SB- and AI-decomposition of $A = (K, \Sigma, \delta, q_0, F)$. Let α be the mapping given by the definition of SB-decomposition. We need to prove that for all states q of A, $q \in F \Leftrightarrow \alpha(q) \in F_1 \times F_2$, where

F_i is the set of accepting states of A_i, $i \in \{1, 2\}$. Let $q \in K$ and let w be a word such that $\overline{\delta}(q_0, w) = q$. Then

$$q \in F \Leftrightarrow w \in L(A) \Leftrightarrow w \in L(A_1) \cap L(A_2) \Leftrightarrow \alpha(q) \in F_1 \times F_2$$

where the first equivalence is implied by the choice of w, the second holds because (A_1, A_2) is an AI-decomposition and the third is a consequence of the properties of α guaranteed by the definition of the SB-decomposition.

□

There is also a relationship between SB- and SI-decompositions, in fact SB- is a stronger version of a state-identifying decomposition, as the following two propositions show.

Definition 4.4. *Let $A_1 = (K_1, \Sigma, \delta_1, p_1, F_1)$ and $A_2 = (K_2, \Sigma, \delta_2, p_2, F_2)$ be DFAs. We shall call a pair of states $(q, r) \in K_1 \times K_2$* reachable*, if there exists a word $w \in \Sigma^*$ such that $\overline{\delta}_1(p_1, w) = q$ and $\overline{\delta}_2(p_2, w) = r$.*

Theorem 4.6. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFA and let (A_1, A_2) be its SB-decomposition. Then (A_1, A_2) also forms an SI-decomposition of A.*

Proof. Let $A_i = (K_i, \Sigma, \delta_i, q_i, F_i)$, $i \in \{1, 2\}$. Since (A_1, A_2) is an SB-decomposition of A, there exists an injective mapping $\alpha\colon K \to K_1 \times K_2$ such that it holds $\alpha(q_0) = (q_1, q_2)$ and

$$(\forall a \in \Sigma)(\forall p \in K); \alpha(\delta(p, a)) = (\delta_1(p_1, a), \delta_2(p_2, a)) \tag{1}$$

where $\alpha(p) = (p_1, p_2)$. Let us define a new mapping $\beta\colon K_1 \times K_2 \to K$ by

$$\beta(p_1, p_2) = \begin{cases} p & \text{if } \exists p \in K, \alpha(p) = (p_1, p_2) \\ q_0 & \text{otherwise} \end{cases} \tag{2}$$

Since α is injective, there exists at most one such p and this definition is correct. We now need to prove that β satisfies the condition from the definition of SI-decomposition, i.e., that

$$(\forall w \in \Sigma^*); \quad \beta(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) = \overline{\delta}(q_0, w)$$

This can be proved by a trivial induction using $\alpha(q_0) = (q_1, q_2)$ for the first step and (1) and (2) for the inductive step, having in mind that all the pairs of states we encounter in the computation of $A_1||A_2$ are thus reachable and therefore we always apply the first possibility in (2). □

Lemma 4.7. *Let A be a DFA without unreachable states and let (A_1, A_2) be its SI-decomposition, with β being the corresponding mapping. Then (A_1, A_2) is an SP-decomposition of A if and only if β is injective on all reachable pairs of states.*

Proof. "$\Rightarrow$" If (A_1, A_2) is an SP-decomposition of A, then β obviously has to be constructed in such a way, that it behaves on all reachable pairs of states as the β mapping in the equation (2) in the proof of Theorem 4.6, otherwise it would not fulfill the condition posed on the mapping by the definition of the SI-decomposition. Since the mapping α is a bijection between the set of states of A and the set of all reachable pairs of states of A_1 and A_2, β defined as its inverse on the set of reachable pairs of states will be injective on this set.

"$\Leftarrow$" Let (A_1, A_2) be an SI-decomposition of A and let β be injective on the set of reachable pairs of states, let β_r denote the mapping β restricted onto the set of all reachable pairs of states of A_1, A_2. Since A has no unreachable states, β_r is also surjective, thus we can define a new mapping $\alpha\colon K \to K_1 \times K_2$ by the equation $\alpha(q) = \beta_r^{-1}(q)$. Since β maps the initial state onto the initial state, so does α, and since β satisfies the condition from the definition of the SI-decomposition, it implies that also α satisfies the condition (i) from the definition of realization of state behavior. Therefore (A_1, A_2) is an SB-decomposition of A, with the corresponding mapping α.

□

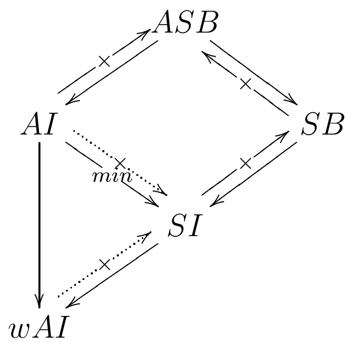
The converse of Theorem 4.6 does not hold, because Example 3.1 presents a minimal automaton that has an AI-decomposition but no SB-decomposition and according to Theorem 4.2, this AI-decomposition is also state-identifying.

The relationship between SB- and ASB-decomposition is obvious directly from their definitions. Later (in the proof of Theorem 5.8) an example of an automaton which has a nontrivial SB-decomposition but does not have a nontrivial ASB-decomposition is exhibited.

It is also easy to see that for any non-minimal automaton A without unreachable states, there exists a nontrivial AI- and wAI-decomposition (A_1, A_2) such that A_1 is the minimal automaton equivalent to A and A_2 has only one state. This decomposition is obviously not state-identifying.

Figure 1 summarizes all the relationships among the decomposition types that we have shown so far.

Now we show that for the case of so-called perfect decompositions, some of the types of decomposition mentioned coincide.



Description:

$A \longrightarrow B$ **:** every A-decomposition is also a B-decomposition

$A \cdots\times\cdots\rightarrow B$ **:** not every A-decomposition is also a B-decomposition

$A - \times \rightarrow B$ **:** there exists a DFA that has a nontrivial A-decomposition but does not have a nontrivial B-decomposition

Figure 1: Relationships between decomposition types of DFA

Definition 4.5. *Let t be a type of decomposition, $t \in \{ASB, SB, AI, SI, wAI\}$. Let A be a DFA having n states, let A_1 and A_2 be DFAs having k and l states, respectively. We shall call the pair (A_1, A_2) a* perfect t-decomposition *of A, if it forms a t-decomposition of A and $n = k \cdot l$.*

Theorem 4.8. *Let A be a DFA with no unreachable states and let (A_1, A_2) be a pair of DFA. Then (A_1, A_2) forms a perfect SI-decomposition of A if and only if (A_1, A_2) forms a perfect SB-decomposition of A.*

Proof. One of the implications is a consequence of Theorem 4.5. As to the second one, since (A_1, A_2) forms a perfect SI-decomposition of A, each of the pairs of states of A_1 and A_2 are reachable and each pair has to correspond to a different state of A in the mapping β, therefore β is bijective and the theorem follows from Lemma 4.7. □

Corollary 4.9. *Let A be a minimal DFA and let (A_1, A_2) be a pair of DFA. Then (A_1, A_2) forms a perfect AI-decomposition of A if and only if (A_1, A_2) forms a perfect ASB-decomposition of A.*

Proof. The proof directly follows from Theorem 4.5, Theorem 4.2 and Theorem 4.8. □

As a consequence of these facts, we can use the necessary and sufficient conditions stated in Theorem 3.3 to look for perfect AI- and SI-decompositions.

Now, let us inspect the relationship between decompositions of an automaton and the decompositions of the corresponding minimal automaton.

Theorem 4.10. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFA and let A_{min} be a minimal DFA such that $L(A) = L(A_{min})$. Let (A_1, A_2) be an SI-decomposition*

(AI-decomposition, wAI-decomposition) of A, then (A_1, A_2) also forms a decomposition of A_{min} of the same type.

Proof. First, note that this theorem does not state that any of the decompositions is nontrivial.

To prove the statement for SI-decompositions, suppose that (A_1, A_2) is an SI-decomposition of A, thus there exists a mapping $\alpha\colon K_1 \times K_2 \to K$ such that it holds

$$(\forall w \in \Sigma^*);\quad \alpha(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) = \overline{\delta}(q_0, w) \tag{3}$$

where δ_i and q_i are the transition function and the initial state of the automaton A_i. Since A_{min} is the minimal automaton corresponding to A, there exists some mapping $\beta\colon K \to K_{min}$ such that

$$(\forall w \in \Sigma^*);\quad \beta(\overline{\delta}(q_0, w)) = \overline{\delta}_{min}(\beta(q_0), w) \tag{4}$$

where δ_{min} is the transition function of A_{min} and K_{min} is the set of states of A_{min}. By the composition of these mappings we obtain the mapping $\beta \circ \alpha\colon K_1 \times K_2 \to K_{min}$. Using (3) and (4) we obtain

$$(\forall w \in \Sigma^*);\quad (\beta \circ \alpha)(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) = \beta(\overline{\delta}(q_0, w)) = \overline{\delta}_{min}(\beta(q_0), w)$$

and since β maps the initial state of A onto the initial state of A_{min}, this equation shows that $\beta \circ \alpha$ is the mapping that combines A_1 and A_2 into A_{min} in the way that the definition of SI-decomposition requires.

This statement is trivial for the AI-decomposition, since $L(A_1) \cap L(A_2) = L(A) = L(A_{min})$, where the first equation comes from the definition of AI-decomposition and the second equation is one of our assumptions.

The situation is similar for wAI-decomposition. Since (A_1, A_2) is a wAI-decomposition of A, there exists a relation $R \subseteq K_1 \times K_2$ such that it holds

$$(\forall w \in \Sigma^*);\quad R(\overline{\delta}_1(q_1, w), \overline{\delta}_2(q_2, w)) \Leftrightarrow w \in L(A) \Leftrightarrow w \in L(A_{min})$$

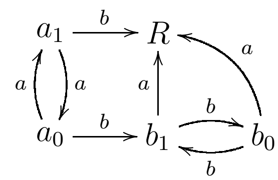
where q_i is the initial state of A_i. Therefore (A_1, A_2) is also a wAI-decomposition of A_{min}.

□

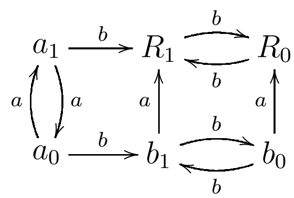
Based on the above theorem it thus suffices to inspect the SI- (AI-, wAI-) decomposability of the minimal automaton accepting a given language, and if we show its undecomposability, we know that the recognition of this language cannot be parallelized in the respective way.

However, this does not hold for SB- and ASB-decompositions, as exhibited by the following example.

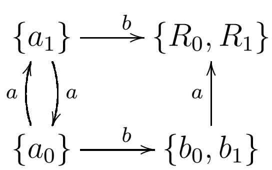
Example 4.1. Let us consider the language $L = \{a^{2k}b^{2l} | k \geq 0, l \geq 1\}$. The minimal automaton $A_{min} = (K, \Sigma_L, \delta, a_0, \{a_0, b_0\})$ has its transition function defined by the following transition diagram:



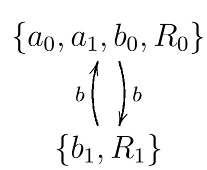
We can easily show that this automaton does not have any nontrivial SB- (and thus neither ASB-) decomposition by enumerating its S.P. partitions. Now let us examine the automaton $A' = (K', \Sigma_L, \delta', a_0, \{a_0, b_0\})$ with the transition function δ' defined by the following diagram:



Clearly, $L(A') = L(A_{min})$, but by inspecting the lattice of S.P. partitions of A', we can the find the partitions $\pi_1 = \{\{a_0\}, \{a_1\}, \{b_0, b_1\}, \{R_0, R_1\}\}$ and $\pi_2 = \{\{a_0, a_1, b_0, R_0\}, \{b_1, R_1\}\}$ such that $\pi_1 \cdot \pi_2 = 0$ and they separate the final states of A'. By Theorem 3.3 we can use these partitions to construct a nontrivial ASB-decomposition of A' formed by the automata A_1 and A_2 having their transition functions defined by the transition diagrams



and



This is an ASB-decomposition and therefore also an SB-decomposition. Note that both A_1 and A_2 have less states than A_{min}.

In the following theorem (inspired by a similar theorem in [7]) we state a condition under which the SB-decomposability of the minimal automaton tells us something about the SB-decomposability of all the other automata accepting the same language. More precisely, if this condition is satisfied and the minimal automaton does not have a nontrivial SB-decomposition, then none of the other automata can be decomposed into two simpler machines, both having less states than the minimal one, i.e., the unpleasant situation from the previous example cannot occur.

Theorem 4.11. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton and let $A_{min} = (K_{min}, \Sigma, \delta_{min}, q_{min}, F_{min})$ be the minimal DFA such that $L(A) = L(A_{min})$. Let (A_1, A_2) be a nontrivial SB-decomposition of A consisting of automata having k and l states. If the lattice of S.P. partitions of A is distributive, then there exists a an SB-decomposition of A_{min} consisting of automata having k' and l' states, such that $k' \leq k$ and $l' \leq l$.*

Proof. Since A_{min} is the minimal DFA such that $L(A) = L(A_{min})$, by Corollary 2.2 there exists a mapping $f : K \to K_{min}$ such that it holds

$$(\forall w \in \Sigma^*);\ f(\overline{\delta}(q_0, w)) = \overline{\delta}_{min}(q_{min}, w) \tag{5}$$

Using the mapping f, let us define a partition ρ on the set of states of A by the equation $p \equiv_\rho q \Leftrightarrow f(p) = f(q)$. As a consequence of (5), ρ is an S.P. partition.

Since (A_1, A_2) is a nontrivial SB-decomposition of A, we can use it to obtain S.P. partitions π_1 and π_2 on the set of states of A such that $\pi_1 \cdot \pi_2 = 0$. Let us define new partitions π_1' and π_2' on the set of states of A_{min} by the equation $f(p) \equiv_{\pi_i'} f(q) \Leftrightarrow p \equiv_{\rho+\pi_i} q$. Since it holds that $\rho + \pi_i \preceq \rho$, this definition does not depend on the choice of the states p and q. It holds that $|\pi_i'| = |\rho + \pi_i| \leq |\pi_i|$, therefore if we prove that π_1' and π_2' are S.P. partitions and $\pi_1' \cdot \pi_2' = 0$, we can use them to construct the desired decomposition.

The fact that π_i' is an S.P. partition on the set of states of A_{min} is a trivial consequence of the fact that $\rho + \pi_i$ is an S.P. partition on the set of states of A. We need to prove that $\pi_1' \cdot \pi_2' = 0$. Let us assume that p' and q' are states of A_{min} such that $p' \equiv_{\pi_1' \cdot \pi_2'} q'$ and p, q are some states of A such that $f(p) = p'$ and $f(q) = q'$. Then $p' \equiv_{\pi_1'} q'$ and $p' \equiv_{\pi_2'} q'$, and by definition of π_i' we get $p \equiv_{\rho+\pi_1} q$ and $p \equiv_{\rho+\pi_2} q$, which is equivalent to $p \equiv_{(\rho+\pi_1)\cdot(\rho+\pi_2)} q$. Since the lattice of all S.P. partitions of A is distributive, we have

$$(\rho + \pi_1) \cdot (\rho + \pi_2) = \rho + (\pi_1 \cdot \pi_2) = \rho + 0 = \rho$$

therefore $p \equiv_\rho q$, which by definition of ρ implies that $f(p) = f(q)$, in other words $p' = q'$. Hence $\pi_1' \cdot \pi_2' = 0$. □

5 Undecomposable Languages

We shall now turn our attention to the languages that cannot be nontrivially decomposed for some of the decomposition types defined.

Definition 5.1. *Let t be a type of decomposition, $t \in \{ASB, SB, AI, SI, wAI\}$. We shall call a DFA t-*undecomposable*, if it has no nontrivial t-decomposition. We shall call a language $L \in \mathcal{R}$ t-undecomposable, if the minimal DFA for L with the input alphabet Σ_L is t-undecomposable. We shall denote $\mathcal{U}_t$ the class of all t-undecomposable regular languages.*

Example 5.1. Consider the language $L = \{w \in \{a,b,c\}^* | \#_a(w) \geq 1\}$. Obviously, the minimal automaton accepting this language consists of 2 states, thus any nontrivial t-decomposition would have to consist of automata with only one state and it can be easily seen that such a t-decomposition does not exist. Hence, this automaton is t-undecomposable for $t \in \{ASB, SB, AI, SI, wAI\}$. Thus $L \in \mathcal{U}_{ASB} \cap \mathcal{U}_{SB} \cap \mathcal{U}_{SI} \cap \mathcal{U}_{AI} \cap \mathcal{U}_{wAI}$.

When deciding about decomposability of languages based on minimal automata accepting these languages, we have to take into account the input alphabet of this minimal automaton, as it was done in Definition 5.1. The reason for this is that a modification of the input alphabet can make the minimal automaton decomposable, as shown by the following theorem.

Theorem 5.1. *Let L be an AI-undecomposable regular language, with A being the minimal automaton with the input alphabet Σ_L accepting L. Let c be a letter such that $c \notin \Sigma_L$ and let A' be the minimal automaton accepting L with input alphabet $\Sigma' = \Sigma_L \cup \{c\}$. Then A' has a nontrivial AI-decomposition if and only if A does not contain a rejecting terminal state.*

Proof. Let us suppose that $A = (K, \Sigma_L, \delta, q_0, F)$ does not contain a rejecting terminal state. Since A is minimal, this implies that A contains no state q such that for its tail $\langle q \rangle$ it holds $\langle q \rangle \cap F = \emptyset$. Indeed, if such q existed but was non-terminal, we could substitute one rejecting terminal state for the whole tail $\langle q \rangle$, thus lowering the number of states, which contradicts the minimality of A. But $A' = (K', \Sigma', \delta', q'_0, F')$ has to contain a rejecting terminal state q_R, because in A' it has to hold $(\forall q \in K'); \delta'(q, c) = q_R$ where q_R has to satisfy $\langle q_R \rangle \cap F' = \emptyset$ (since all words containing c have to be rejected) and non-terminality of q_R would again contradict the minimality of A'.

Now we can prove that A' can be obtained from A by only adding the rejecting terminal state q_R and the transitions from all states to this one on the input letter c. Such automaton would obviously accept the language $L(A)$ and work over the alphabet $\Sigma' = \Sigma_L \cup \{c\}$. It would be also minimal, since no pair of original states can be equivalent, because they were not equivalent in A, and no original state could be equivalent to q_R, because none of them satisfies $\langle q \rangle \cap F = \emptyset$. Therefore, we really obtain the automaton A' in this way.

Now that we know how A' works, we can show that it is AI-decomposable. We shall decompose A' into two automata, one doing the work of A and the other filtering out all the words that contain the letter c. Formally, let $A_1 = (K, \Sigma', \delta_1, q_0, F)$ be a DFA such that $\delta_1(q, a) = \delta(q, a)$ for all $a \in \Sigma$ and $\delta_1(q, c) = q$. Then let $A_2 = (\{p_0, p_1\}, \Sigma', \delta_2, p_0, \{p_0\})$ be a DFA with the transition function δ_2 defined by the following transition diagram

$$p_0 \xrightarrow{c} p_1$$

For any $w \in \Sigma^*$ it holds that $w \in L(A_2)$ and $w \in L(A_1) \Leftrightarrow w \in L(A)$. For any word w containing c it holds $w \notin L(A_2)$ and thus $w \notin L(A_1) \cap L(A_2)$. Therefore (A_1, A_2) forms a nontrivial AI-decomposition of A.

To prove the converse, let us assume that A contains a rejecting terminal state q_R. Then the minimal automaton A' accepting L and having the input alphabet Σ' has the same number of states, in fact we can derive it from A by adding new c-transitions from each state into q_R. Now let (A_1, A_2) be a nontrivial AI-decomposition of this automaton. Then we could obtain an AI-decomposition of A from (A_1, A_2) by just removing all the c-transitions from both automata. Since A and A' have the same number of states, this decomposition would be nontrivial, which contradicts the undecomposability of A. Therefore A' must be undecomposable, too. □

Before inspecting the closure properties of the classes of undecomposable languages, we need to prove undecomposability of two types of languages. This is done by the following lemmas.

Lemma 5.2. *For each $n \in \mathbb{N}, n \geq 2$ the minimal DFA accepting the one-word language $L^{(n)} = \{a^{n-2}\}$ does not have a nontrivial wAI-decomposition.*

Proof. It can be easily seen that the minimal DFA A accepting $L^{(n)}$ (and having the one-letter input alphabet $\Sigma = \{a\}$) has n states and the transition function δ defined by the following transition diagram:

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \cdots\xrightarrow{a} q_{n-3} \xrightarrow{a} q_{n-2} \xrightarrow{a} q_{n-1}$$

with q_{n-2} being the only accepting state. Let us suppose that there exists a nontrivial wAI-decomposition of A, say (A_1, A_2), with $R \subseteq K_1 \times K_2$ being the corresponding relation from Definition 4.3. Then both A_1 and A_2 must have at most $n - 1$ states. Since the input alphabet now contains only one letter, we can easily describe the structure of these automata. Let m denote the number of states of A_1. Without loss of generality we can assume that all of them are reachable. We can label the states of A_1 as $q_0, q_1, \ldots, q_{m-1}$ so that for all $i \in \{0, 1, \ldots, m-2\}$ it holds $\delta_1(q_i, a) = q_{i+1}$ and $\delta_1(q_{m-1}, a) = q_j$ for some $j \in \{0, 1, \ldots, m-1\}$. This implies that the computation of A_1 on the word a^{n-2} ends in state q_r for some $r \in \{j, j+1, \ldots, m-1\}$. But so does the computation of A_1 on the words $a^{n-2+k.(m-j)}$ for all $k \in \mathbb{N}$. By repeating the same argument for A_2 (let us denote m' the number of states of A_2 and $q_{j'}$ the state of A_2 such that $\delta_2(q_{m'-1}, a) = q_{j'}$) we show that the computations of A_2 on the words $a^{n-2}, a^{n-2+k(m'-j')}$ for all $k \in \mathbb{N}$ end in the same state, say $q_{r'}$. Since $a^{n-2} \in L(A)$, it must hold $(q_r, q'_r) \in R$. But since the computation of A_1 on the word $a^{n-2+(m-j)(m'-j')}$ ends in q_r and the computation of A_2 on the same word ends in $q_{r'}$, this would imply that $a^{n-2+(m-j)(m'-j')} \in L(A)$, which is a contradiction, since $(m-j)(m'-j') > 0$. □

Lemma 5.3. *For each $n \in \mathbb{N}$ the minimal DFA accepting the language $L^{(n)} = \{a^k | k \geq n-1\}$ does not have a nontrivial wAI-decomposition.*

Proof. The minimal automaton A accepting $L^{(n)}$ is easy to construct, it consists of n states. Let us suppose that there exists a nontrivial wAI-decomposition of A, say (A_1, A_2), with $R \subseteq K_1 \times K_2$ being the corresponding relation from Definition 4.3. Then both A_1 and A_2 must have at most $n-1$ states. Again the input alphabet contains only one letter, so the structure of these automata can be easily described. Let m denote the number of states of A_1. Without loss of generality we can assume that all of them are reachable. We can denote the states of A_1 by $q_0, q_1, \ldots, q_{m-1}$ so that for all $i \in \{0, 1, \ldots, m-2\}$ it holds $\delta(q_i, a) = q_{i+1}$ and $\delta(q_{m-1}, a) = q_j$ for some $j \in \{0, 1, \ldots, m-1\}$ (the automaton consists of an initial sequence and a loop, possibly a trivial one). We can do the same for A_2, let m' denote the number of its states and let us label them $p_0, p_1, \ldots, p_{m'-1}$ in such a way that for all $i \in \{0, 1, \ldots, m'-2\}$ it holds $\delta(p_i, a) = p_{i+1}$ and $\delta(p_{m'-1}, a) = p_{j'}$ for some $j' \in \{0, 1, \ldots, m'-1\}$. Now let r and r' be the indices such that the computations of A_1 and A_2 on the word a^{n-1} end in states q_r and $p_{r'}$, respectively. Since $m \leq n-1$ and $m' \leq n-1$, it holds that $j \leq r \leq m-1$ and $j' \leq r' \leq m'-1$, i.e., the computation on a^{n-1} ends in the loop for both automata. We know that $a^{n-1} \in L(A)$, therefore $(q_r, p_{r'}) \in R$, but also $a^{n-1+k} \in L(A)$ for all $k \in \mathbb{N}$, so $(q_{j+(r-j+k \bmod (m-j))}, p_{j'+(r'-j'+k \bmod (m'-j'))}) \in R$ for all $k \in \mathbb{N}$. Let s (and t) denote the states that occur in the loop of A_1 (and A_2) immediately before the state q_r(and $q_{r'}$). If we instantiate $(m-j)(m'-j')-1$ for k, we obtain that $(s, t) \in R$, too. It is easy to see that also the computation of A_1 (and A_2) on the word a^{n-2} ends in the state s (and t), since these computations have to end in the loops of the respective automata, because the initial sequence can be at most $n-2$ states long. Thus a computation on a word of length $n-2$ must lead past this sequence of states. Therefore $(s,t) \in R$ implies $a^{n-2} \in L(A)$, which is a contradiction. □

Theorem 5.4. *The classes $\mathcal{U}_{ASB}$, $\mathcal{U}_{SB}$, $\mathcal{U}_{SI}$, $\mathcal{U}_{AI}$ and $\mathcal{U}_{wAI}$ are not closed under intersection.*

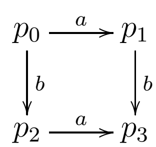
Proof. Consider the languages

$$\begin{aligned} L_1 &= \{w \in \{a,b,c\}^* | \#_a(w) \geq 1\} \\ L_2 &= \{w \in \{a,b,c\}^* | \#_b(w) \geq 1\} \end{aligned}$$

According to Example 5.1, these languages are not t-decomposable for any $t \in \{ASB, SB, AI, SI, wAI\}$. Obviously

$$L_1 \cap L_2 = \{w \in \{a,b,c\}^* | \#_a(w) \geq 1 \wedge \#_b(w) \geq 1\}$$

Using the well-known method of finding minimal automaton for a given language, we can see that the minimal DFA accepting the language $L_1 \cap L_2$ is $A = (\{p_0, p_1, p_2, p_3\}, \Sigma, \delta, p_0, \{p_3\})$, with the δ function given by the following transition diagram



This automaton is obviously decomposable into two finite automata $A_1 = (\{q_0, q_1\}, \Sigma, \delta_1, q_0, \{q_1\})$ and $A_2 = (\{r_0, r_1\}, \Sigma, \delta_2, r_0, \{r_1\})$ with transition functions given by these transition diagrams, respectively:

Now it is easy to see that (A_1, A_2) is a nontrivial ASB-decomposition of A, given by the mapping $\alpha(p_i) = (q_{i \bmod 2}, r_{i \operatorname{div} 2})$. Thus A is ASB-decomposable and so is the language $L_1 \cap L_2$. Using the results from the previous section it follows that this is also a decomposition of all other defined types, which completes the proof. □

Since the family of regular languages $\mathcal{R}$ is closed under intersection, we have the following corollary.

Corollary 5.5. *$\mathcal{U}_t \subsetneq \mathcal{R}$, $t \in \{ASB, SB, SI, AI, wAI\}$.*

Theorem 5.6. *The classes $\mathcal{U}_{SB}$, $\mathcal{U}_{SI}$ and $\mathcal{U}_{wAI}$ are closed under complement.*

Proof. If $\Sigma_L = \Sigma_{L^C}$ (i.e., the minimal alphabet of the language does not change by complementing the language), the theorem is easy to prove. Let $A = (K, \Sigma, \delta, q_0, F)$ be the minimal automaton accepting L. Then clearly the automaton $A' = (K, \Sigma, \delta, q_0, K - F)$ is a DFA such that $L(A') = L^C$. Moreover, it must be minimal, because if it had any equivalent states, these states would also be equivalent in A. But that means that minimal automata for L and L' operate in the same way. Thus if α (or β, or R) is a mapping (or a relation) proving that (A_1, A_2) is a nontrivial SB- (or SI-, or wAI-) decomposition of A', then α (or β, or $\neg R$) proves that (A_1, A_2) is also a nontrivial decomposition of the same type of the original automaton A, which would contradict its undecomposability. Therefore A' is $t-$undecomposable for all types t mentioned.

Let us consider the case when the minimal alphabet changes by complementing L. Since L^C is defined by $\{w \in \Sigma_L^* | w \notin L\}$, $\Sigma_{L^C} \subseteq \Sigma_L$ and it is a proper subset if and only if L contains all words from Σ_L^* that contain some given letter. Let us suppose that $\Sigma_L - \Sigma_{L^C} = \{c\}$. (This can be easily extended to more letters.) Let A' be the minimal automaton accepting L^C with input alphabet Σ_{L^C} and let A'' be the minimal automaton accepting L^C with input alphabet Σ_L. Using the arguments from the previous paragraph, A'' and A can be obtained from each other by only complementing the set of accepting states. Now there are two possibilities worth distinguishing: either A' contains or does not contain a rejecting terminal state q_R. If it contains such a state, then A'' can be obtained from A' by just adding c-transitions from each state into the state q_R, therefore A, A' and A'' have the same number of states. If A' contains no rejecting terminal state it has to be added, along with the respective transitions, when transforming A' into A'' since A'' has to reject all words containing c. Therefore in this case, A and A'' have one more state then A', and this state is terminal. The minimality of all constructed automata can be proved using the same arguments as in the previous paragraph. Now that we know the relationship between A, A' and A'', let us conclude the proof.

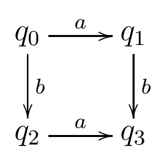
Assume that A' has a nontrivial wAI-decomposition formed by (A_1, A_2) and a relation R. We shall construct a nontrivial wAI-decomposition of A, thus obtaining a contradiction. If A' does not contain q_R, then A has one more state then A', therefore we can extend A_i ($i \in \{1, 2\}$) by one state and it will still have less states then A. Therefore we add a terminal rejecting state to both automata and add c-transitions from all states into these new ones. Thus we obtain two automata with input alphabet Σ_{L^C} and they form a wAI-decomposition of A'' with the relation R. Therefore it is also a wAI-decomposition of A, now with the relation $\neg R$. On the other hand, if A' does contain a rejecting terminal state q_R, then there must also exist a pair of states $(p_1, p_2) \in K_1 \times K_2$ such that $(\forall w \in \Sigma_{L^C}); (\delta_1(p_1, w), \delta_2(p_2, w)) \notin R$. Therefore, if we modify A_i by adding c-transitions from all states into state p_i, we again obtain two automata that form a nontrivial wAI-decomposition of A'' and A (with the relations R and $\neg R$, respectively).

The proof for SI-decompositions is similar, since if (A_1, A_2) is an SI-decomposition of A'' defined by a mapping β, then it is also an SI-decomposition of A defined by the same mapping β, and all the other arguments are the same as before.

To prove the theorem for SB-decompositions, we shall use Theorem 3.3. If A' has a nontrivial SB-decomposition, then there exist S.P. partitions π_1 and π_2 on the set of states of A' such that $\pi_1 \cdot \pi_2 = 0$. Now if A' contains q_R then π_1 and π_2 also determine a nontrivial SB-decomposition of A'' (and thus also of A). This is true because the only transitions that have been added when modifying A' to A'' were the c-transitions from all states to q_R and these cannot violate the S.P. property since they all point to the same state. On the other hand, if A' does not contain q_R, A' has again less states then A. Hence we can add a new block containing only q_R into both π_1 and π_2, thus obtaining two S.P.partitions on the set of states of A'' (or A) that are still not trivial and satisfy $\pi_1 \cdot \pi_2 = 0$, i.e., they determine a nontrivial SB-decomposition of both A'' and A. $\square$

Theorem 5.7. *The class $\mathcal{U}_{ASB}$ is not closed under complement.*

Proof. Consider the language $L = \{b, c\}^* \cup \{a, c\}^*$. Minimal DFA accepting this language is $A = (\{q_0, q_1, q_2, q_3\}, \Sigma, \delta, q_0, \{q_0, q_1, q_2\})$ with δ function defined again as follows:



ASB-undecomposability of this automaton can be easily proved using Theorem 3.3, as this automaton has only two nontrivial S.P. partitions, but they do not separate the final states of A. However, it can be seen that $L^C = \{w \in \{a, b, c\}^* | \#_a(w) \geq 1 \wedge \#_b(w) \geq 1\}$ and this language can be decomposed, because the two S.P. partitions mentioned above now do separate the final states of the automaton (the resulting decomposition was already shown in the proof of Theorem 5.4). Thus $\mathcal{U}_{ASB}$ is not closed under complement. $\square$

Theorem 5.8. *The classes $\mathcal{U}_{ASB}$, $\mathcal{U}_{AI}$, $\mathcal{U}_{SI}$ and $\mathcal{U}_{wAI}$ are not closed under union.*

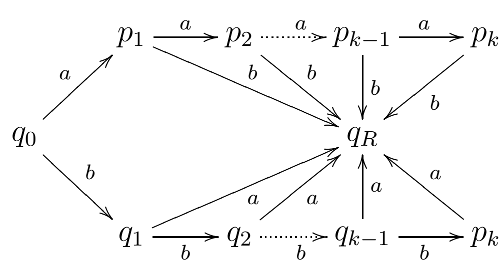
Proof. To prove the statement about $\mathcal{U}_{ASB}$, let us consider the languages

$$\begin{aligned} L_1 &= \{w \in \{a,b,c\}^* | \#_a(w) \bmod 3 = 0 \wedge \#_b(w) \bmod 5 = 0\} \cup \\ &\cup \{w \in \{a,b,c\}^* | \#_a(w) \bmod 3 = 2 \wedge \#_b(w) \bmod 5 = 4\} \\ L_2 &= \{w \in \{a,b,c\}^* | \#_a(w) \bmod 3 = 0 \wedge \#_b(w) \bmod 5 = 4\} \cup \\ &\cup \{w \in \{a,b,c\}^* | \#_a(w) \bmod 3 = 2 \wedge \#_b(w) \bmod 5 = 0\} \end{aligned}$$

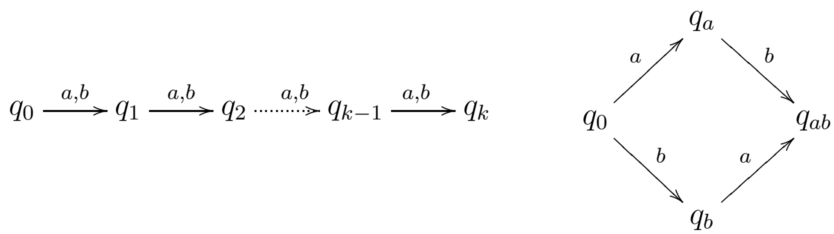
It is easy to see that minimal automata for both L_1 and L_2 have 15 states and they only differ in the definition of the accepting states (each automaton has two accepting states). Their ASB-undecomposability can again be easily verified using Theorem 3.3: each automaton has only two nontrivial S.P. partitions (one represents counting of symbols a modulo 3, the other counting of symbols b modulo 5), we shall denote them π_1 and π_2. It holds $\pi_1 \cdot \pi_2 = 0$ but they do not separate the final states of any of the automata.

The minimal automaton for the union $L_1 \cup L_2$ has again 15 states with the same transition function, but it has four accepting states. The partitions π_1 and π_2 still satisfy the condition $\pi_1 \cdot \pi_2 = 0$, but they now also separate the final states of the automaton in the sense of Definition 3.7. Hence, using Theorem 3.3, the language $L_1 \cup L_2$ has an ASB-decomposition.

To prove the same statement about $\mathcal{U}_{AI}$, $\mathcal{U}_{SI}$ and $\mathcal{U}_{wAI}$, we shall consider the languages $L_1 = \{a^i | i \geq k\}$ and $L_2 = \{b^i | i \geq k\}$ for some given constant $k \geq 1$. According to Lemma 5.3, minimal automata for both languages have $k+1$ states and are wAI-undecomposable (and hence also AI- and SI-undecomposable). For $L_1 \cup L_2$ we have the minimal automaton A having $2k+2$ states with the transition function δ defined by the following transition diagram



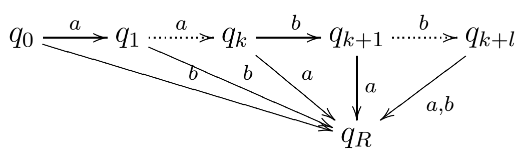
with accepting states p_k and q_k. However, this automaton can be AI-decomposed into A_1 and A_2 consisting of $k+1$ and 4 states, having the corresponding transition functions defined by the transition diagrams



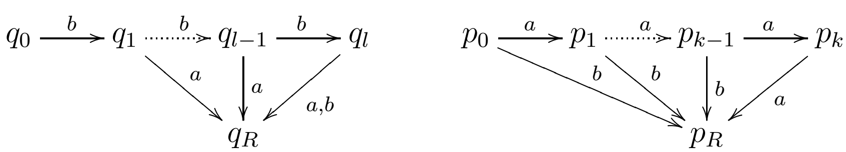
with accepting states q_k, q_a and q_b. Now, $L(A_1) = \{w \in \{a,b\}^* |\ |w| \geq k\}$ and $L(A_2) = \{a\}^* \cup \{b\}^*$, thus it holds $L(A_1) \cap L(A_2) = L_1 \cup L_2$. Therefore, (A_1, A_2) is an AI-decomposition (and also a wAI-decomposition and by Theorem 4.2 also an SI-decomposition) of A. □

Theorem 5.9. *The classes $\mathcal{U}_{AI}$, $\mathcal{U}_{SI}$, $\mathcal{U}_{wAI}$ are not closed under concatenation.*

Proof. To prove the statement, we shall consider the one-word languages $L_1 = \{a^k\}$ and $L_2 = \{b^l\}$ for some given constants $k, l \geq 1$. According to Lemma 5.2, minimal automata for both languages are wAI-undecomposable (and hence also AI- and SI-undecomposable). For $L_1 \cdot L_2 = \{a^k b^l\}$ we have the minimal automaton A having $k+l+2$ states with the transition function defined by the following transition diagram



with accepting state q_{k+l}. However, this automaton can be AI-decomposed into A_1 and A_2 consisting of $l+2$ and $k+2$ states, having the corresponding transition functions defined by the transition diagrams



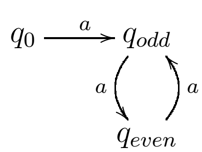
with accepting states q_l and p_k. Now $L(A_1) = \{a\}^* \cdot \{b^l\}$ and $L(A_2) = \{a^k\} \cdot \{b\}^*$, thus it holds $L(A_1) \cap L(A_2) = L_1 \cdot L_2$. Therefore, (A_1, A_2) is an AI-decomposition (and thus also a wAI- and SI-decomposition) of A, which proves the theorem. □

Theorem 5.10. *The classes $\mathcal{U}_{ASB}$, $\mathcal{U}_{SB}$, $\mathcal{U}_{SI}$, $\mathcal{U}_{AI}$ and $\mathcal{U}_{wAI}$ are not closed under iteration.*

Proof. It suffices to consider the one-word language $L = \{a^6\}$, which is according to Lemma 5.2 t-undecomposable for all the mentioned types t. The minimal DFA for language $L^* = \{a^{6k} | k \geq 0\}$ can be easily decomposed into two automata, one counting symbols a modulo 2 and the other modulo 3, therefore L^* is decomposable. Moreover, this is a decomposition of all mentioned types. □

Theorem 5.11. *The classes $\mathcal{U}_{ASB}$, $\mathcal{U}_{SB}$, $\mathcal{U}_{SI}$, $\mathcal{U}_{AI}$ and $\mathcal{U}_{wAI}$ are not closed under non-erasing homomorphism.*

Proof. Consider the language $L = \{a\}^+$. It is obvious that the minimal automaton accepting L has 2 states and therefore cannot be t-decomposed for any type t. However, if we apply to L the non-erasing homomorphism $h\colon \{a\}^* \to \{a\}^*$ defined by $h(a) = aa$, we obtain $h(L) = \{a^{2k} | k \geq 1\}$. We shall show that this language is decomposable. Indeed, the minimal automaton accepting $h(L)$ is $A = (\{q_0, q_{odd}, q_{even}\}, \{a\}, \delta, q_0, \{q_{even}\})$ with the transition function defined by the transition diagram

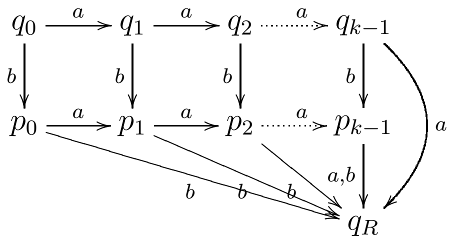


and can be decomposed into two automata consisting of 2 states each, one of them accepting the language $L_1 = \{a\}^+$ and the other accepting the language $L_2 = \{a^{2k} | k \geq 0\}$. This is a decomposition of all desired types, thus the theorem holds. □

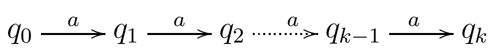
Corollary 5.12. *The classes $\mathcal{U}_{ASB}$, $\mathcal{U}_{SB}$, $\mathcal{U}_{SI}$, $\mathcal{U}_{AI}$ and $\mathcal{U}_{wAI}$ are not closed under homomorphism.*

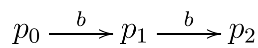
Theorem 5.13. *The classes $\mathcal{U}_{AI}$, $\mathcal{U}_{SI}$ and $\mathcal{U}_{wAI}$ are not closed under inverse homomorphism.*

Proof. To prove this, we shall consider the one-word language $L = \{a^{4k-1}\}$ for some fixed $k \geq 2$. By Lemma 5.2, L is wAI-undecomposable. Now let us consider a homomorphism $h\colon \{a,b\}^* \to \{a\}^*$ defined by the equations $h(a) = a^2$ and $h(b) = a^{2k+1}$. Consider $h^{-1}(L)$. By the definition of inverse homomorphism, $h^{-1}(L) = \{w \in \{a,b\}^* | h(w) \in L\} = \{w \in \{a,b\}^* | h(w) = a^{4k-1}\}$. Obviously such word w cannot consist of letters a only, because $h(a) = a^2$ and $4k-1$ is odd. Also, w cannot contain more than one letter b, because $h(b) = a^{2k+1}$ and $2.(2k+1) > 4k-1$. Now it is easy to see that $h^{-1}(L) = \{w \in \{a,b\}^* | \#_b(w) = 1 \wedge |w| = k\}$. The minimal automaton accepting $h^{-1}(L)$ and working over $\Sigma_{h^{-1}(L)}$ consists of $2k+1$ states, with the transition function defined by the transition diagram



with p_{k-1} being the only accepting state. But this automaton can be AI-decomposed into two automata A_1 and A_2, having $k+1$ and 3 states, with the transition functions defined by the following transition diagrams





with accepting states q_{k-1} and p_1. Now $L(A_1) = \{w \in \{a,b\}^* | \#_a(w) = k-1\}$ and $L(A_2) = \{w \in \{a,b\}^* | \#_b(w) = 1\}$, therefore it holds $L(A_1) \cap L(A_2) = \{w \in \{a,b\}^* | \#_b(w) = 1 \wedge |w| = k\} = h^{-1}(L)$, so the minimal DFA accepting $h^{-1}(L)$ has a nontrivial AI-decomposition (and also wAI- and SI-decomposition). Hence the theorem holds. $\square$

All known closure properties of the classes of undecomposable languages are summarized in Table 1.

	$\cap$	$\cup$	^{C}	$\cdot$	$*$	h	h^{-1}
ASB-undecomposable	no	no	no		no	no	
SB-undecomposable	no		yes		no	no	
AI-undecomposable	no	no		no	no	no	no
SI-undecomposable	no	no	yes	no	no	no	no
wAI-undecomposable	no	no	yes	no	no	no	no

Table 1: Closure properties of the classes of undecomposable languages

6 Degrees of Decomposability

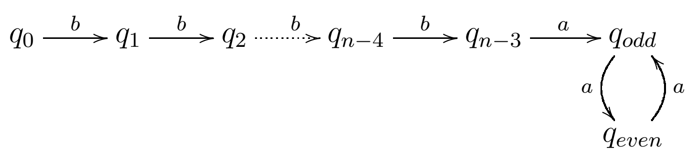
From our previous proofs we know that there exist undecomposable regular languages. But it is interesting to ask, whether there exist "nearly undecomposable" languages, that means languages, that do have a nontrivial decomposition, but it does not help us much, regarding the number of states. This question is for some types of decomposition answered by the following theorem.

Theorem 6.1. *For each $n \in \mathbb{N}, n \geq 3$ there exists a minimal DFA A having n states, such that A is ASB-decomposable, but if (A_1, A_2) is a nontrivial ASB-decomposition (nontrivial SB-decomposition, nontrivial AI-decomposition) of A, then both A_1 and A_2 have $n-1$ states.*

Proof. Let us consider the language

$$L_n = \{w \in \{a,b\}^* | \#_b(w) = n-4\} \,.\, \{b\} \,.\, \{w \in \{a,b\}^* | \#_a(w) = 2k, k \geq 1\}$$

As can be easily shown, the minimal DFA accepting this language is $A = (K, \Sigma, \delta, q_0, \{q_{even}\})$, having n states, with the transition functions defined by the following transition diagram:



We shall prove that this DFA has the desired property.

To find out something about possible SB-decompositions of A, we should inspect its S.P. partitions. Let us assume that π is an S.P. partition on the set of states of A such that q_i and q_j are in the same block of π; $i, j \in \{0, 1, \ldots, n-3\}$. As a consequence of the S.P. property, if $i, j < n-3$ then also q_{i+1} and q_{j+1} are in the same block of π, because $\delta(q_i, b) = q_{i+1}$ and $\delta(q_j, b) = q_{j+1}$. By applying this argument a finite number of times, we can show that there exists some $k \in \{0, 1, \ldots, n-4\}$ such that $q_k \equiv_\pi q_{n-3}$. But since $\delta(q_k, a) = q_k$, it also holds that $q_k \equiv_\pi q_{n-3} \equiv_\pi q_{odd} \equiv_\pi q_{even}$. Thus any partition that makes q_i and q_j equal ($i, j \in \{0, 1, \ldots, n-3\}$) cannot distinguish between the states q_{n-3}, q_{odd} and q_{even}.

Now let us suppose that π is an S.P. partition on the set of states of A such that for some $i \in \{0, 1, \ldots, n-4\}$, $q_i \equiv_\pi q_{odd}$ or $q_i \equiv_\pi q_{even}$. Then it also holds that $q_i \equiv_\pi q_{i+1}$, because $\delta(q_i, b) = q_{i+1}$ and $\delta(q, b) = q$, where $q \in \{q_{odd}, q_{even}\}$. However, we have already shown that $q_i \equiv_\pi q_{i+1}$ implies

$q_{n-3} \equiv_\pi q_{odd} \equiv_\pi q_{even}$, thus this S.P. partition cannot distinguish between the states q_{n-3}, q_{odd} and q_{even}, either.

From these observations it follows that if π is any S.P. partition on the set of states of A such that q_{n-3}, q_{odd} and q_{even} are not all equivalent modulo π, then π must also contain $n-3$ blocks, each of which contains only one state q_i,$i \in \{0, 1, \dots, n-4\}$.

To form an SB-decomposition, there must exist two nontrivial S.P. partitions π_1 and π_2 on the set of states of A, such that $\pi_1 \cdot \pi_2 = 0$. Hence at least one of these partitions has to distinguish between the states q_{n-3}, q_{odd} and q_{even}. If one of these partitions handled this (if it had each of this three states in a different block), from what was said comes that it would need to have a separate block for any other state too, thus it would be the trivial partition 0. Therefore, both π_1 and π_2 have to separate some pair of states from the set $\{q_{n-3}, q_{odd}, q_{even}\}$, and that means that among other blocks they both have $n-3$ blocks, each of which contains exactly one of the states $q_0, q_1, \dots, q_{n-4}$. Now it is easy to see that the only pair of nontrivial S.P. partitions π_1 and π_2 such that $\pi_1 \cdot \pi_2 = 0$ is

$$\begin{aligned}\pi_1 &= \{\{q_0\}, \{q_1\}, \dots, \{q_{n-4}\}, \{q_{n-3}\}, \{q_{odd}, q_{even}\}\} \\ \pi_2 &= \{\{q_0\}, \{q_1\}, \dots, \{q_{n-4}\}, \{q_{n-3}, q_{odd}\}, \{q_{even}\}\}\end{aligned}$$

which corresponds to an SB-decomposition into two automata, each of them having $n-1$ states.

The DFA A has only one final state, hence this decomposition is also a nontrivial ASB-decomposition of A (see Corollary 3.4). Since nontrivial ASB-decomposition is also a nontrivial AI-decomposition, A is AI-decomposable. It remains to prove that all automata that could form a nontrivial AI-decomposition of A must have $n-1$ states. This requires a different approach.

First, let us consider the word $u = b^{n-3}ab^{n-3}a^2$. Since $u \notin L(A)$, at least one of the automata forming an AI-decomposition of A has to reject this word. We show that such automaton has to distinguish between the words $\{\varepsilon, b, b^2, \dots, b^{n-3}, b^{n-3}a\}$, i.e., its computations on any two of them must end in different states. Indeed, suppose that $u \notin L(A_1)$, but $\overline{\delta}_1(q_0, b^i) = \overline{\delta}_1(q_0, b^j)$, where $0 \leq i < j \leq n-3$ and δ_1 is the transition function of A_1. Then, since A_1 is deterministic, it also hold that $\overline{\delta}_1(q_0, b^i.b^{n-3-j}ab^{n-3}a^2) = \overline{\delta}_1(q_0, b^j.b^{n-3-j}ab^{n-3}a^2)$ because we have added the same suffix to both words. Keeping in mind that $u = b^{n-3}ab^{n-3}a^2$, we obtain $\overline{\delta}_1(q_0, u) = \overline{\delta}_1(q_0, b^{n-3-j+i}ab^{n-3}a^2) \in F$, because since $i < j$, $b^{n-3-j+i}ab^{n-3}a^2 \in L(A)$, thus $b^{n-3-j+i}ab^{n-3}a^2$ has to belong to $L(A_1)$, too. Therefore $u \in L(A_1)$, a contradiction. Similarly, let us now suppose that

$\overline{\delta}_1(q_0, b^i) = \overline{\delta}_1(q_0, b^{n-3}a)$. Then, using the same argument as before, also $\overline{\delta}_1(q_0, b^i.b^{n-3}a^2) = \overline{\delta}_1(q_0, b^{n-3}.b^{n-3}a^2) \in F$, because $b^{n-3}.b^{n-3}a^2$ has to belong to $L(A_1)$. But that again implies that also $u \in L(A_1)$, which is again a contradiction. So in order to filter out the word u, the nontrivial AI-decomposition has to contain at least one automaton having $n-1$ states corresponding to words $\{\varepsilon, b, b^2, \dots, b^{n-3}, b^{n-3}a\}$, each to exactly one of them.

Now, let us consider the word $v = b^{n-4}a^2b$ in a similar way. Again, $v \notin L(A)$, so at least one of the automata forming the AI-decomposition has to reject it, say $v \notin A_2$. We show that this automaton has to distinguish between the words $\{\varepsilon, b, b^2, \dots, b^{n-4}, b^{n-4}a^2b, b^{n-4}a^2ba^2\}$, using the same method as before.

- If $\overline{\delta}_2(q_0, b^i) = \overline{\delta}_2(q_0, b^j)$ for some $0 \le i < j \le n-4$, then also $\overline{\delta}_2(q_0, v) = \overline{\delta}_2(q_0, b^i.b^{n-4-i}a^2b) = \overline{\delta}_2(q_0, b^j.b^{n-4-i}a^2b) \in F$. But A_2 cannot accept v, thus has to distinguish between b^i and b^j.

- If $\overline{\delta}_2(q_0, b^i) = \overline{\delta}_2(q_0, b^{n-4}a^2b)$ for some $0 \le i \le n-4$, then also $\overline{\delta}_2(q_0, v) = \overline{\delta}_2(q_0, b^i.b^{n-4-i}a^2b) = \overline{\delta}_2(q_0, b^{n-4}a^2b.b^{n-4-i}a^2b) \in F$. But A_2 cannot accept v, thus has to distinguish between b^i and $b^{n-4}a^2b$.

- If $\overline{\delta}_2(q_0, b^i) = \overline{\delta}_2(q_0, b^{n-4}a^2ba^2)$ for some $0 \le i \le n-4$, then also $\overline{\delta}_2(q_0, v) = \overline{\delta}_2(q_0, b^i.b^{n-4-i}a^2b) = \overline{\delta}_2(q_0, b^{n-4}a^2ba^2.b^{n-4-i}a^2b) \in F$. But A_2 cannot accept v, thus has to distinguish between b^i and $b^{n-4}a^2ba^2$.

- If $\overline{\delta}_2(q_0, b^{n-4}a^2b) = \overline{\delta}_2(q_0, b^{n-4}a^2ba^2)$, then $\overline{\delta}_2(q_0, b^{n-4}a^2ba^2) \in F$ implies $\overline{\delta}_2(q_0, b^{n-4}a^2b) \in F$, again a contradiction. Thus A_2 has to distinguish between $b^{n-4}a^2b$ and $b^{n-4}a^2ba^2$.

So in order to filter out the word v, the nontrivial AI-decomposition has to contain at least one automaton having $n-1$ states corresponding to the words $\{\varepsilon, b, b^2, \dots, b^{n-4}, b^{n-4}a^2b, b^{n-4}a^2ba^2\}$, each to exactly one of them.

Thus if each of the words u, v is filtered out by a different automaton, then they both have $n-1$ states. All we have to prove now is that u and v cannot both be filtered out by the same automaton.

Let us assume the opposite, that both u and v are filtered out by A_1. Then, according to our first reasoning concerning u, the automaton A_1 has to have $n-1$ states, each one of them corresponding to exactly one of the words $\{\varepsilon, b, b^2, \dots, b^{n-3}, b^{n-3}a\}$ (a state corresponds to a word w if the computation of A_1 on w ends in this state). But from our reasoning concerning v, these $n-1$ states also have to correspond to the words $\{\varepsilon, b, b^2, \dots, b^{n-4}, b^{n-4}a^2b, b^{n-4}a^2ba^2\}$. So the first $n-3$ states correspond to the words $\varepsilon, b, b^2, \dots, b^{n-4}$, let's call them $q_0, q_1, \dots, q_{n-4}$ respectively.

Each of the remaining two states corresponds to exactly one of the states from each of the sets $\{b^{n-3}, b^{n-3}a\}$ and $\{b^{n-4}a^2b, b^{n-4}a^2ba^2\}$. Let r denote the state that corresponds to b^{n-3} and s the one that corresponds to $b^{n-3}a$. Suppose that s corresponds to $b^{n-4}a^2b$ and r corresponds to $b^{n-4}a^2ba^2$. That implies that $\overline{\delta}_1(s, a^2) = r$. But since s also corresponds to $b^{n-3}a$, r would have to correspond to $b^{n-3}a^3$ (and also b^{n-3}). But that would mean that $\overline{\delta}_1(q_0, u) = \overline{\delta}_1(q_0, b^{n-3}.ab^{n-3}a^2) = \overline{\delta}_1(q_0, b^{n-3}a^3.ab^{n-3}a^2) \in F$, so such automaton would not be able to filter out u. Thus r corresponds to the words $\{b^{n-3}, b^{n-4}a^2b\}$ and s corresponds to $\{b^{n-3}a, b^{n-4}a^2ba^2\}$. Since $b^{n-4}a^2ba^2 \in L_n$, s has to be an accepting state. Moreover, from the assignment of the words b^{n-3} and $b^{n-3}a$ to the states r and s it follows that $\delta_1(r, a) = s$ and (since then $b^{n-4}a^2ba$ corresponds to s) also $\delta_1(s, a) = s$.

To finish the proof, we show that there is no possible value for $\delta_1(s, b)$.

- Suppose that $\delta_1(s, b) = q_0$, then q_0 corresponds to both ε and $b^{n-4}a^2ba^2b$, thus $\overline{\delta}_1(q_0, v) = \overline{\delta}_1(q_0, \varepsilon.b^{n-4}a^2b) = \overline{\delta}_2(q_0, b^{n-4}a^2ba^2b.b^{n-4}a^2b) \in F$, a contradiction.

- Suppose that $\delta_1(s, b) = q_i$, $1 \leq i \leq n-4$, then the state q_i corresponds to both the words b^i and $b^{n-3}ab$, thus $\overline{\delta}_1(q_0, u) = \overline{\delta}_1(q_0, b^{n-3}ab.b^{n-4}a^2) = \overline{\delta}_2(q_0, b^i.b^{n-4}a^2) \in F$, again a contradiction.

- The same argument (for $i := n-3$) holds for the case $\delta_1(s, b) = r$.

- The last case to consider is that $\delta_1(s, b) = s$. However, this would imply that $\overline{\delta}(q_0, u) = \overline{\delta}(q_0, b^{n-3}ab^{n-3}a^2) = \overline{\delta}(s, b^{n-3}a^2) = s$ which is an accepting state as we have already proved, thus A_1 again cannot filter out u.

This completes the proof that the words u and v cannot be both filtered out by the same automaton. Therefore each of the automata forming any nontrivial AI-decomposition of A has to filter one of them, thus they both have $n-1$ states. □

According to the following theorems, there exists also a lower bound for the number states of the automata in any decomposition, and there exist automata where this lower bound is achieved.

Theorem 6.2. *Let (A_1, A_2) be an SB-decomposition (ASB-decomposition) of an n-state DFA A, consisting of automata having r and s states. Then $r.s \geq n$.*

Proof. If (A_1, A_2) is the given SB-decomposition (ASB-decomposition) of A, then according to Theorem 3.3, there exist S.P. partitions π_1 and π_2 on the set of states of A, such that $\pi_1 \cdot \pi_2 = 0$, π_1 has r blocks and π_2 has s blocks. Thus $\pi_1 \cdot \pi_2$ can have at most $r.s$ blocks and since $\pi_1 \cdot \pi_2 = 0$, A can have at most $r.s$ states. ☐

Definition 6.1. *Let A be a DFA, let (A_1, A_2) be its nontrivial SB- (ASB-) decomposition with the corresponding S.P. partitions π_1 and π_2. We shall call this decomposition* redundant, *if there exist S.P. partitions $\pi_1' \succeq \pi_1$ and $\pi_2' \succeq \pi_2$ such that at least one of these inequalities is strict, but it still holds $\pi_1' \cdot \pi_2' = 0$ (and π_1' and π_2' separate the final states of A).*

Theorem 6.3. *For each $r, s \in \mathbb{N}$, $r, s \geq 2$, there exists a DFA A consisting of $r.s$ states and having only one nontrivial nonredundant SB-decomposition (ASB-decomposition) up to the order of automata, consisting of automata having r and s states.*

Proof. Let us study the the automaton $A_{r,s} = (K, \Sigma, \delta, q_{0,0}, F)$ defined by $K = \{q_{i,j} | i \in \{0, \ldots, r-1\}, j \in \{0, \ldots, s-1\}\}$, $F = \{q_{r-1,s-1}\}$ and the transition function δ defined by the equations

$$\begin{aligned} \delta(q_{i,j}, a) &= q_{i+1,j} \text{ for i} \in \{0, \ldots, \text{r}-2\}, \text{j} \in \{0, \ldots, \text{s}-1\} \\ \delta(q_{r-1,j}, a) &= q_{r-1,j} \text{ for j} \in \{0, \ldots, \text{s}-1\} \\ \delta(q_{i,j}, b) &= q_{i,j+1} \text{ for i} \in \{0, \ldots, \text{r}-1\}, \text{j} \in \{0, \ldots, \text{s}-2\} \\ \delta(q_{i,s-1}, b) &= q_{i,s-1} \text{ for i} \in \{0, \ldots, \text{r}-1\} \end{aligned}$$

To inspect the SB-decompositions of $A_{r,s}$, let us study the S.P. partitions on the set of its states. From the method for generating all S.P. partitions of an automaton that is described in [7], we know that each nontrivial S.P. partition can be obtained as a sum of some partitions $\pi^m_{p,t}$, where $\pi^m_{p,t}$ denotes the minimal S.P. partition such that it does not distinguish between states p and t, i.e., they belong into the same block. Let us determine $\pi^m_{p,t}$ for various states p and t of $A_{r,s}$.

First, let us consider the case of $\pi^m_{p,t}$ such that $p = q_{i,j}$, $t = q_{i',j'}$ and both inequalities $i < i'$ and $j < j'$ hold. Since $q_{i,j} \equiv_\pi q_{i',j'}$, $\overline{\delta}(q_{i,j}, a^{i'-i}b^{j'-j}) = q_{i',j'}$ and $\overline{\delta}(q_{i',j'}, a^{i'-i}b^{j'-j}) = q_{2i'-i,2j'-j}$ (if $2i' - i < r$ and $2j' - j < s$), as a consequence of the substitution property of π, we obtain $q_{i,j} \equiv_\pi q_{2i'-i,2j'-j}$. By applying this argument a finite number of times (keeping in mind the construction of $A_{r,s}$), we obtain $q_{i,j} \equiv_\pi q_{r-1,s-1}$. Now let $k \in \{i, \ldots, r-1\}$ and let $l \in \{i, \ldots, s-1\}$. Then $\overline{\delta}(q_{i,j}, a^{k-i}b^{l-j}) = q_{k,l}$ and $\overline{\delta}(q_{i',j'}, a^{k-i}b^{l-j}) = q_{k+i'-i,l+j'-j}$ (if such states exist), therefore $q_{k,l} \equiv_\pi q_{k+i'-i,l+j'-j}$. Again, we can use the same argument to show that $q_{k,l} \equiv_\pi q_{r-1,s-1}$. Therefore for the type of $\pi = \pi^m_{p,t}$, we have $q_{k,l} \equiv_\pi q_{k',l'}$ for all k, l, k', l' such that $i \leq k, k' < r$ and $j \leq l, l' < s$.

Now let us consider the case of $\pi^m_{p,t}$ such that $p = q_{i,j}$, $t = q_{i',j'}$ and it holds $i > i'$ and $j < j'$. Since $q_{i,j} \equiv_\pi q_{i',j'}$, $\overline{\delta}(q_{i,j}, a^{r-1-i}b^{s-1-j'}) = q_{r-1,s-1-(j'-j)}$ and $\overline{\delta}(q_{i',j'}, a^{r-1-i}b^{s-1-j'}) = q_{r-1-(i-i'),s-1}$, as a consequence of the substitution property of π, we have $q_{r-1,s-1-(j'-j)} \equiv_\pi q_{r-1-(i-i'),s-1}$. By exploiting the substitution property again on this equivalence, using the words $a^{i-i'-1}$, $b^{j'-j-1}$ and $b^{j'-j}$, we obtain $q_{r-2,s-1} \equiv_\pi q_{r-1,s-1} \equiv_\pi q_{r-2,s-2}$. Therefore in this case, no such $\pi^m_{p,t}$ partition can distinguish between states $q_{r-2,s-1}$, $q_{r-1,s-1}$ and $q_{r-2,s-2}$.

The last case to consider is the case of $\pi^m_{p,t}$ such that $p = q_{i,j}$, $t = q_{i',j'}$ and it holds $i = i'$ (the case $j = j'$ is analogous). Without loss of generality we can assume that $j < j'$. Now, using the same arguments as in the first case, we can show that $q_{i,l} \equiv_\pi q_{i,l'}$ for all l, l' such that $j \leq l, l' < s$. Therefore for each given k such that $i \leq k < r$, it holds that $q_{k,l} \equiv_\pi q_{k,l'}$ and all of the states not mentioned in this equivalence form separate blocks of $\pi^m_{p,t}$.

It is easy to verify that one nontrivial ASB-decomposition of $A_{r,s}$ is given by the S.P. partitions

$$\pi_1 = \{\{q_{0,0}, \dots, q_{0,s-1}\}, \{q_{1,0}, \dots, q_{1,s-1}\}, \dots, \{q_{r-1,0}, \dots, q_{r-1,s-1}\}\}$$
$$\pi_2 = \{\{q_{0,0}, \dots, q_{r-1,0}\}, \{q_{0,1}, \dots, q_{r-1,1}\}, \dots, \{q_{0,s-1}, \dots, q_{r-1,s-1}\}\}$$

Now we show that any other SB-decomposition of $A_{r,s}$ is given by S.P. partitions preceding to π_1 and π_2 in the partial order $\preceq$ and therefore is redundant.

Indeed, notice that none of the $\pi^m_{p,t}$ partitions of the first and the second discussed type can distinguish between any of the states $q_{r-2,s-1}$, $q_{r-1,s-1}$ and $q_{r-2,s-2}$, therefore no sum of them can, either. For the partitions of the third type, it holds either $q_{r-2,s-1} \equiv_\pi q_{r-1,s-1}$ or $q_{r-1,s-1} \equiv_\pi q_{r-2,s-2}$, therefore it will take two partitions to distinguish between these three states. Hence any nontrivial SB-decomposition is determined by two S.P. partitions, both of which must be of the third type. But it is easy to see that for any partition π of this type it holds either $\pi \preceq \pi_1$ or $\pi \preceq \pi_2$. $\square$

Now that we know that both the lower and the upper bound for the number of states of the decomposition can be reached by some automaton, it is natural to ask, whether also all the values in this interval can be achieved. This is partly answered by the following result.

Definition 6.2. *Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton, let $K \cap \{p_0, p_1, \dots, p_{k-1}\} = \emptyset$ and let c be a new symbol not included in Σ. We shall define a k-*extension *A' of the automaton A by the following construction: $A' = (K \cup \{p_0, p_1, \dots, p_{k-1}\}, \Sigma \cup \{c\}, \delta', p_0, F)$, where the transition* *function δ' is defined as follows:*

$$
\begin{array}{rrcl}
(\forall q \in K)\ (\forall a \in \Sigma); & \delta'(q,a) & = & \delta(q,a) \\
(\forall q \in K); & \delta'(q,c) & = & q \\
(\forall p \in \{p_0, p_1, \ldots, p_{k-1}\})\ (\forall a \in \Sigma); & \delta'(p,a) & = & p \\
(\forall i \in \{0,1,\ldots,k-2\}); & \delta'(p_i,c) & = & p_{i+1} \\
 & \delta'(p_{k-1},c) & = & q_0
\end{array}
$$

Lemma 6.4. *Let A be a DFA consisting of n states, all of which are reachable. Let A' be its k-extension. Then A has a nontrivial SB-decomposition (nontrivial ASB-decomposition) consisting of automata having r and s states if and only if A' has a nontrivial decomposition of the same type, consisting of automata having $k+r$ and $k+s$ states.*

Proof. Similarly to the proof of Theorem 6.1, we will try to inspect S.P. partitions on the set of states of A'. Let us assume that π is an S.P. partition on the set of states of A such that p_i and p_j are in the same block of π; $i,j \in \{0,1,\ldots,k-1\}$. As a consequence of the S.P. property, if $i,j < k-1$ then also p_{i+1} and p_{j+1} are in the same block of π, because $\delta'(p_i,c) = p_{i+1}$ and $\delta'(p_j,c) = p_{j+1}$. By applying this argument a finite number of times, we can show that there exists some $l \in \{0,1,\ldots,k-2\}$ such that $p_l \equiv_\pi p_{k-1}$, and using the argument once more, we obtain $p_{l+1} \equiv_\pi q_0$. However, it holds $\delta'(p_l,a) = p_l$ for all $a \in \Sigma$, hence $p_l \equiv_\pi \overline{\delta}'(q_0,w)$ for all $w \in \Sigma^*$. Since all of the states of A are reachable, we have $p_l \equiv_\pi q$ for all $q \in K$. Thus such a partition cannot distinguish between the states of automaton A.

Now let us suppose that π is an S.P. partition on the set of states of A such that for some $i \in \{0,1,\ldots,k-1\}$, $p_i \equiv_\pi q$ for some q in K. Then it also holds that $p_i \equiv_\pi p_{i+1}$, because $\delta(p_i,c) = p_{i+1}$ and $\delta(q,c) = q$. But we have already shown that $q_i \equiv_\pi q_{i+1}$ implies that all of the states in K are equivalent modulo π, thus this S.P. partition cannot distinguish between the states of A, either.

From these observations it follows that if π is any S.P. partition on the set of states of A such that the states of A are not all equivalent modulo π, then π must also contain k blocks, each of which contains only one state p_i, where $i \in \{0,1,\ldots,k-1\}$.

Now we can prove the equivalence stated in the theorem.

"$\Rightarrow$": Let A have an SB-decomposition consisting of r and s states. Then there exist S.P. partitions π_1 and π_2 on the set of states of A such that $\pi_1 \cdot \pi_2 = 0$. Let us now construct new partitions π_1' and π_2' on the set of states of A' by the following definition:

$$\begin{aligned}\pi_1' &= \pi_1 \cup \{\{p_0\},\{p_1\},\ldots,\{p_{k-1}\}\}\\ \pi_2' &= \pi_2 \cup \{\{p_0\},\{p_1\},\ldots,\{p_{k-1}\}\}\end{aligned}$$

Obviously π_1' and π_2' are partitions on the set of states of A'. They also have substitution property, because for the states in K this property is inherited from π_1 and π_2, and the new states $p_0, p_1, \ldots, p_{k-1}$ cannot violate this property either, because each of these states belongs to a separate block in π_1' and π_2', making the substitution property hold trivially. Obviously, neither do the new c-moves defined on the states from K violate the substitution property. Finally, it holds that $\pi_1' \cdot \pi_2' = 0$. To see this, note that for a state $q \in K$, it holds $[q]_{\pi_1' \cdot \pi_2'} = [q]_{\pi_1 \cdot \pi_2} = \{q\}$, since $\pi_1 \cdot \pi_2 = 0$. For a state $q \in K' - K$, $[q]_{\pi_i'} = \{q\}$ for $i \in \{1,2\}$ thus $[q]_{\pi_1' \cdot \pi_2'} = \{q\}$, too. Hence each state of A' belongs to a separate block of $\pi_1' \cdot \pi_2'$, which implies $\pi_1' \cdot \pi_2' = 0$. Therefore π_1' and π_2' induce an SB-decomposition of A'. It is also easy to see that if π_1 and π_2 separate the final states of A, then also π_1' and π_2' separate the final states of A', making the induced decomposition an ASB-decomposition.

"$\Leftarrow$": Now let us assume that A' has an SB-decomposition and π_1' and π_2' are the S.P. partitions on K' that induce this decomposition, thus $\pi_1' \cdot \pi_2' = 0$. From the observations made in the beginning of this proof, we know that any S.P. partition that can distinguish between the states in K in any way, must contain each of the states $p_0, p_1 \ldots p_{k-1}$ in a separate block containing only this state. As $\pi_1' \cdot \pi_2' = 0$, for all $q_1, q_2 \in K$, at least one of these partitions must distinguish between these states, i.e., $[q_1]_{\pi_i'} \neq [q_2]_{\pi_i'}$. If one of the partitions distinguished between all such pairs, it would imply that this partition must contain a separate block for each one of the states in K', thus becoming a trivial partition 0, resulting in a trivial decomposition. Therefore both π_1' and π_2' have to distinguish between some pair of states from K, which implies that they both contain a separate block for each of the states $p_0, p_1 \ldots p_{k-1}$ containing no other state. By removing these k blocks from π_1' and π_2', we obtain new partitions on the set K:

$$\begin{aligned}\pi_1 &= \pi_1' - \{\{p_0\},\{p_1\},\ldots,\{p_{k-1}\}\}\\ \pi_2 &= \pi_2' - \{\{p_0\},\{p_1\},\ldots,\{p_{k-1}\}\}\end{aligned}$$

These partitions preserve the substitution property, since $(\forall a \in \Sigma)(\forall q \in K)$: $\delta(q,a) \in K$ and π_1 and π_2 were S.P. partitions. It also holds $\pi_1 \cdot \pi_2 = 0$, as for all $q_1, q_2 \in K$, $q_1 \equiv_{\pi_1 \cdot \pi_2} q_2$ implies $q_1 \equiv_{\pi_1' \cdot \pi_2'} q_2$ and that implies $q_1 = q_2$. So π_1 and π_2 induce an SB-decomposition of A. As π'_1 and π'_2 were nontrivial, so are π_1 and π_2 and the obtained decomposition. It is again easy to see that if π'_1 and π'_2 separate the final states of A', then also π_1 and π_2 must separate the final states of A.

□

We can combine these facts into the following corollary.

Corollary 6.5. *Let $n \in \mathbb{N}$ be such that $n = k + r.s$, where $r, s, k \in \mathbb{N}$, $r, s \geq 2$. Then there exists a DFA A consisting of n states, such that it has only one nontrivial nonredundant SB-decomposition (ASB-decomposition) up to the order of the automata in the decomposition, and this decomposition consists of automata with $k + r$ and $k + s$ states.*

Proof. For the given r and s, by Theorem 6.3 there exists a DFA having $r.s$ states and only one nontrivial nonredundant SB-decomposition (ASB-decomposition), consisting of automata having r and s states. We can construct k-extension of this automaton, which will have $k + r.s$ states and by Lemma 6.4, this extension will have only one nontrivial nonredundant SB-decomposition (ASB-decomposition), consisting of automata having $k + r$ and $k + s$ states (it is easy to see from the proof of Lemma 6.4 that this decomposition will be nonredundant if and only if the original decomposition of the automaton from Theorem 6.3 is nonredundant). □

7 Conclusion

In this thesis, we have studied the possibilities of parallel decomposition of deterministic finite automata. We have defined two types of decomposition similar to the previously studied state behavior decompositions of sequential machines, and derived necessary and sufficient conditions for the existence of these decompositions. Since the decompositions of sequential machines were different from the concept of the advisor introduced in the beginning as our motivation, we have defined three new types of parallel decomposition of deterministic finite automata and established various conditions for the existence of these types of decomposition and relationships between them. We have also inspected the relationships between the decomposition of any deterministic finite automaton and the decomposition of the corresponding minimal automaton. We have introduced perfect decompositions and showed that for these decompositions, some of the decomposition classes coincide. Furthermore, we have studied the classes of undecomposable languages. We have exhibited some examples of undecomposable languages and then studied the closure properties of these classes of languages. We have concluded the thesis by showing that there exist automata for most degrees of decomposability from the interval given by the two mentioned boundaries — the undecomposable and the perfectly decomposable ones.

There are many directions in which this work can be extended. The results presented in this thesis directly imply some questions that have remained open. For example, it would be very useful to find necessary and sufficient conditions for the existence of the decompositions defined in Section 4 that would be easy to verify. It could also be interesting to explore deeper the perfectly decomposable languages, including the closure properties of the corresponding language classes. Another interesting challenge is to prove a statement similar to Corollary 6.5 for all the other types of parallel decomposition.

From a more general point of view, another challenging task is to explore the possibilities for parallel decompositions of nondeterministic finite automata and of all the more powerful computational models, up to the Turing machines, with respect to different complexity measures. One related question that could be worth studying is the generalization of the advisor concept presented in this thesis in the context of more powerful computational models.

References

[1] W. S. McCulloch, W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophysics, 5, 1943, p. 115–133

[2] E. F. Moore, *Gedanken-experiments on sequential machines*, Automata studies, Princeton, New Jersey, Princeton University Press 1956, p. 129–153

[3] G. H. Mealy, *A method for synthesizing sequential circuits*, Bell Systems Technical Journal, volume 34, No. 5 (Sept. 1955), 1045-79

[4] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979

[5] F. Gecseg and I. Peak, *Algebraic Theory of Automata*, Akademiai Kiado, 1972

[6] F. Gecseg and I. Peak, *Products of Automata*, Akademie-Verlag Berlin, 1986

[7] J. Hartmanis and R. E. Stearns, *Algebraic structure theory of sequential machines*, Prentice-Hall, 1966

[8] J. Hopcroft, *An* $n \log n$ *algorithm for minimizing states in a finite automaton*, Theory of Machines and Computation, J. Kohavi, Ed. New York: Academic, 1971, pp. 189–196.

[9] B. W. Watson and J. Daciuk, *An efficient incremental DFA minimization algorithm*, Natural Language Engineering 9 (1): 49-64, Cambridge University Press, 2003

[10] S. MacLane and G. Birkhoff, *Algebra*, 2nd edition, The Macmillan Company, New York, 1968

Abstrakt

Deterministický konečný automat je jednoduchý abstraktný výpočtový model s mnohými praktickými aplikáciami napríklad v oblasti návrhu hardwaru, spracovania prirodzeného jazyka a grafických dát, model checkingu a mnohých ďalších.

Táto diplomová práca sa zaoberá možnosťami rozkladu deteministického konečného automatu na dva jednoduchšie automaty takým spôsobom, aby z výsledkov výpočtov oboch týchto automatov bolo možné určiť aký by bol výsledok výpočtu pôvodného automatu na tomto vstupnom slove. Za mieru zložitosti deterministického konečného automatu sme v celej práci považovali počet jeho stavov. Do úvahy prichádza viacero možností čo považovať za výsledok výpočtu automatu — napríklad stav v ktorom automat skončil alebo jeho rozhodnutie akceptovať alebo zamietnuť vstupné slovo. V závislosti od toho ktorú z týchto možností sme uprednostnili, získali sme rôzne definície rozkladov konečných automatov. Pre takto definované rozklady sme následne študovali podmienky ich existencie a vzájomné vzťahy medzi nimi.

V práci sme sa taktiež venovali vzťahu medzi rozložiteľnosťou daného konečného automatu a rozložiteľnosťou príslušného minimálneho automatu, ktorý je s ním ekvivalentný. Ukázali sme niekoľko tvrdení a príkladov ktoré tento vzťah objasňujú pre každý z uvedených typov rozkladu.

Ďalším predmetom nášho záujmu boli dokonale rozložiteľné automaty (t.j. automaty ktoré je možné rozložiť do najvyššej teoreticky možnej miery) a ich protipól — nerozložiteľné automaty. Ukázali sme, že ak sa zaujímame iba o dokonalé rozklady automatov, niektoré zo skôr definovaných typov rozkladov sú totožné, čo nám umožní lepšie charakterizovať podmienky existencie týchto rozkladov. Pre triedy jazykov ktorých minimálny automat je nerozložiteľný určitým typom rozkladu sme skúmali ich uzáverové vlastnosti vzhľadom na základné operácie na jazykoch. Prácu uzatvárame dôkazom tvrdenia, že v istom zmysle pre každú z hodnôt v intervale určenom spomínanými dvoma extrémami nerozložiteľných a dokonale rozložiteľných automatov existuje deterministický konečný automat s touto mierou rozložiteľnosti.

Kľúčové slová. deterministický konečný automat, paralelný rozklad, rozklad stavového správania