Deterministic Generalized Automata *

Dora Giammarresi** and Rosa Montalbano

Dipartimento di Matematica ed Applicazioni, Università di Palermo
via Archirafi, 34 - 90123 Palermo - ITALY
e-mail: {dora,rosalba}@altair.math.unipa.it

Abstract. A generalized automaton (GA) is a finite automaton where the single transitions are defined on words rather than on single letters. Generalized automata were considered by K. Hashiguchi who proved that the problem of calculating the size of a minimal GA is decidable.
We define the model of deterministic generalized automaton (DGA) and study the problem of its minimization. A DGA has the restriction that, for each state, the sets of words corresponding to the transitions of that state are prefix sets. We solve the problem of calculating the number of states of a minimal DGA for a given language, by giving a procedure that effectively constructs it starting from the minimal (conventional) deterministic automaton.

1 Introduction

Generalized automata (GA) are a model of representation for regular languages that extends the notion of finite automata by allowing the single transitions to be defined on words rather than on single letters. Intuitively, a generalized automaton can be obtained from a conventional one by shrinking long paths of the graph in a unique edge with a "long" label. Therefore, generalized automata are usually more concise than conventional ones representing the same event.

In the past decades, several efforts have been devoted to compute the complexity of representation of a given language inside different models of representation (deterministic, non-deterministic, unambiguous, two-way, alternating, probabilistic, pebbles automata, regular expressions, logical formalisms and so on). The complexity of a language in a given model is generally understood as the size of the minimal representation of the language in that model. For example, a classical measure of the complexity of a finite automaton is its number of states and the complexity of a language in this model is the number of states of a minimal (with respect to the number of states) automaton recognizing it.

In this context, Hashiguchi in 1991 investigated the problem of computing the size of the minimal representation of a given regular language in the model

* Work partially supported by the ESPRIT II Basic Research Actions Program of the EC under Project ASMICS 2 (contract No. 6317) and in part by MURST under project 40% *Algoritmi, Modelli di Calcolo, Strutture Informative.*

** Partially supported by a research fellowship by *(INdAM).*

of generalized automata (see [H91]). In particular, he proved that the problem of calculating the number of states of a minimal GA is actually decidable.

A strictly related problem consists of effectively computing a minimal representation of a given language in a model. In the case of conventional deterministic finite automata, it can be proved that the minimal automaton is unique and an algorithm to calculate it starting from any equivalent deterministic automaton can be obtained using the Myhill-Nerode's theorem (see, for example, [HU79]). For non-deterministic automata there are only partial results stating that there is no unique minimal automaton but there are no constructive procedures for computing it, excepting the one that lists all possible automata. In [JR91] the computational complexity of different problems concerning minimization is studied in a general setting for non-deterministic automata and it is proved that all these problems are computationally hard.

In this paper we introduce the model of *deterministic generalized automata* (DGA) and deal with the minimization problem for this model. In order to preserve all properties implied by the notion of determinism in the case of conventional automata, DGA have the restriction that the sets of words corresponding to the transitions of each state are prefix sets. We solve the problem of computing the number of states of a minimal DGA by giving a procedure to construct a minimal DGA for a given language starting from the minimal (conventional) deterministic one. We introduce two operations that allow one to reduce the number of states of a DGA: the first, called $\mathcal{I}$*-reduction*, contracts states that are "indistinguishable" and the second, called $\mathcal{S}$*-reduction*, suppresses states that are "superfluous". Then we give the conditions under which such operations can be performed. We show that there can be deterministic GA that are irreducible (with respect to the above operations) but not minimal and give necessary and sufficient conditions to reduce a deterministic GA to get a minimal one. Moreover, we show that, differently from the case of conventional deterministic automata, the minimal deterministic GA is not unique.

The size of the minimal representation of a language in a given model (which measures the complexity of the language) plays a primary role also in comparing different models according to their intrinsic succinctness. Much work has been devoted to studying succinctness of representation when transducers are considered (see, for example [WK94]). In the case of finite automata, very recently, Harel et al. studied exponential discrepancies in the succinctness of finite automata when augmented by combinations of various additional mechanisms like alternation (i.e. both universal and existential branching), concurrency, "two-wayness" and pebbles (see [GH94]). We conclude the paper by discussing problems of discrepancy in succinctness between non-deterministic and deterministic versions of generalized automata and give some open problems.

2 Preliminaries

We denote by Σ a finite alphabet and by Σ^* the free monoid generated by Σ. The elements of Σ are called letters, those of Σ^* are called words; the subsets of Σ^* (i.e. the sets of words) are called languages over Σ. The length of a word w is denoted by $|w|$. Given two words v and w, we say that v is a *prefix* of w if there exists a word u such that $w = vu$. Given a set of words X, we say that X is a *prefix set* if no word in X is prefix of some other word in X. Given two sets of words X and Y, the *concatenation of* X *and* Y, indicated as $X \cdot Y$ contains all words xy with $x \in X$ and $y \in Y$.

A *finite (non deterministic) automaton* is a quintuple $\mathcal{A} = (\Sigma, Q, I, F, E)$ where Q is a finite set of *states*, $I, F \subseteq Q$ are the sets of *initial* and of *final states* respectively and $E \subseteq Q \times \Sigma \times Q$ is a set of labeled *edges*. We denote an edge of $\mathcal{A}$ by $e = (r, a, s)$, where $r, s \in Q$ and $a \in \Sigma$ is the label of e. A *path* of *length* n in $\mathcal{A}$ is a sequence of edges $e_i = (r_i, a_i, r_{i+1}) \in E$, for $i = 1, \ldots n$, that we denote by $[r_1, a_1 \ldots a_n, r_{n+1}]$. The word $w = a_1 \ldots a_n$ is called the label of the path $e_1, \ldots, e_n$. If $r_1 \in I$ and $r_{n+1} \in F$ then $e_1, \ldots, e_n$ is called *accepting path* and word w is said *accepted by* (or *recognized by*) $\mathcal{A}$. The language accepted (or recognized) by the automaton $\mathcal{A}$ is the set of all accepted words. A language L over Σ is *recognizable* if L is the language accepted by some finite automaton.

Two finite automata are *equivalent* if they accept the same language. A *minimal* automaton for a language L is an automaton with the minimum number of states among all equivalent automata accepting L. An automaton $\mathcal{A} = (\Sigma, Q, I, F, E)$ is *deterministic* if $|I| = 1$, and for any state $q \in Q$ and any letter $a \in \Sigma$, there exists at most one state $p \in Q$ such that edge $(q, a, p) \in E$. It can be proved (see [HU79]) that for any (non-deterministic) automaton there exists an equivalent deterministic automaton. However, in general, the non-deterministic automaton has a minor number of states than the corresponding deterministic one. The *minimal deterministic automaton* for a given language L is the automaton with minimal number of states among all equivalent deterministic automata accepting L. It can be proved that there is a unique minimal deterministic automaton equivalent to a given deterministic automaton $\mathcal{A} = (\Sigma, Q, i, F, E)$. It can be to obtained as follows (see [HU79] or [P90] for more details). We define an equivalence relation in the set of states Q called *indistinguishability*: two states $p, q \in Q$ are *indistinguishable* if for any word $w \in \Sigma^*$, there exists a path $[p, w, f]$ with $f \in F$ if and only if there exists a path $[q, w, f']$ with $f' \in F$. The minimal deterministic automaton equivalent to $\mathcal{A}$ can be obtained by contracting indistinguishable states of $\mathcal{A}$.

Since an automaton is actually a directed labeled graph, in the following we will need further notations on graphs. Given an edge $e = (p, a, q)$, we call p the beginning and q the end, respectively, of e. An edge e is said *incident* to a state q if it begins or ends in q. An edge $e = (p, a, q)$ is a *self-loop* if $p = q$. A path $e_1, e_2, \ldots, e_n$, where $e_i = (r_i, a_i, r_{i+1})$, is a *cycle* if $r_1 = r_{n+1}$. Notice that, a self-loop is a cycle of length one. A graph is acyclic if it does not contain cycles.

Let $G = (Q, E)$ be a directed graph where Q is the set of vertices and E is the set of edges. If $S \subseteq Q$, the subgraph of G induced by S is the graph $G_S = (S, E')$ such that $E' \subseteq E$ is the set of all edges whose beginnings and ends are in S. We say that S induces a maximal acyclic subgraph if G_S is an acyclic subgraph of G that is not a subgraph of any other acyclic subgraph of G.

3 Generalized automata

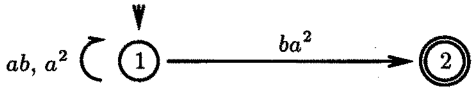
In this section we consider a generalization of the conventional model of automaton described above. Differently from that model, the labels of the edges are words over the alphabet Σ (i.e. not only single letters).

Definition 1. A *generalized (non-deterministic) automaton* (GA) is a quintuple $\mathcal{A} = (\Sigma, Q, I, F, E)$ where Q is a finite set of *states*, $I, F \subseteq Q$ are the sets of *initial* and *final* states and $E \subseteq Q \times \Sigma^* \times Q$ is a *finite* set of labeled *edges*.

Notice that the finiteness of the set E is now necessary to get a finite device: without this restriction we could have as many edges as the words in Σ^*.

The notion of recognizability for a language by generalized automata is the same as by conventional automata, i.e. a word is accepted if it is the label of an accepting path. More precisely, a word $w \in \Sigma^*$ is recognized by a generalized automaton $\mathcal{A}$ if there exist words $w_1, w_2, \ldots, w_n \in \Sigma^*$ and edges $e_1, e_2, \ldots, e_n \in E$ such that w_i is the label of e_i for $i = 1, \ldots, n$, the sequence $e_1, e_2, \ldots, e_n$ is an accepting path and $w_1 w_2 \ldots w_n = w$.

Observe that, in this case, the fact that a word w is accepted by a generalized automaton does not imply that all factors of w are labels of some path in the automaton. Consider, for example, the generalized automaton below recognizing the language $L = (ab + a^2)^* ba^2$ over $\Sigma = \{a, b\}$. Notice that the prefix ab^2 of the accepted word ab^2a^2 does not correspond to any path in the graph.



In general, by allowing the labels of the edges to be words of any length, a generalized automaton gives a representation of a language by means of a graph that is possibly much smaller than the corresponding representation by conventional automaton. For example, any finite language can be represented by a GA with only two states, despite of the length of its words.

Generalized automata were considered by Hashiguchi in [H91]. He studied the problem of calculating the number of states of a minimal generalized automaton for a given language and proved that this problem is decidable.

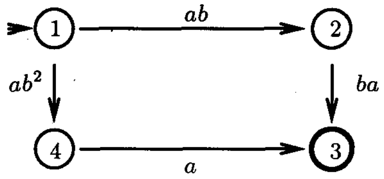
If $\mathcal{A}$ is a GA, denote by $D(\mathcal{A})$ the maximal length of the labels of the edges in $\mathcal{A}$. The decidability is a consequence of the following theorem.

Theorem 2. [K. Hashiguchi 1991] *Let L be a recognizable language and m the cardinality of the syntactic monoid for L. There exists a minimal generalized automaton $\mathcal{A}$ recognizing L such that $D(\mathcal{A}) \leq 2m(m+2)(4m(m+2)+3)$.*

We observe that the number $D(\mathcal{A})$ in the statement of the theorem is actually a very huge number. This because the cardinality of the syntactic monoid of a language is of the order of n^n where n is the number of states of the minimal deterministic (conventional) automaton for L (see [P90]).

4 Deterministic generalized automata

We now define and study the model of generalized automaton in the deterministic case. We start with a remark. In the case of deterministic conventional automata, the "local" condition that, given any state q, for any letter a there is at most one edge beginning in q with label a implies the "global" condition that also for any word w there is at most one path beginning in q with label w. The same does not hold in the case of GA's as shown by following example.



Notice that there exist two paths in $\mathcal{A}$ beginning in state 1 with label ab^2a and this despite $\mathcal{A}$ is deterministic in the "classical" sense (i.e. despite for any state q and any word w there is at most one edge beginning in q with label w).

To capture the "global" property of the conventional notion of determinism we need a stronger condition on the set of edges. Given a GA $\mathcal{A} = (\Sigma, Q, I, F, E)$, for any state $q \in Q$, we define the set $W(q) = \{w \in \Sigma^* \mid (q, w, r) \in E,\, r \in Q\}$. In other words, $W(q)$ contains the labels of all edges beginning in q in $\mathcal{A}$.

Definition 3. Let $\mathcal{A} = (\Sigma, Q, I, F, E)$ be a generalized automaton. We say that $\mathcal{A}$ is *deterministic* if $|I| = 1$ and for any state $q \in Q$, the set $W(q)$ is a prefix set.

Notice that the condition that $W(q)$ is a prefix set effectively guarantees that for any state q and for any word w there is at most one path beginning in q with label w. Moreover conventional deterministic automata satisfy the above definition because the $W(q)$'s are subsets of the alphabet that is a prefix set.

We now focus on the problem of reducing (with respect to the number of states) a given DGA. We will define two operations that transform a DGA into a smaller equivalent one. The first operation contracts indistinguishable states similarly to the minimization operation for conventional deterministic automata. The second operation reduces the number of states by shrinking long paths in a unique edge with a "long" label.

Given a DGA $\mathcal{A} = (\Sigma, Q, i, F, E)$, for any $q \in Q$, we denote by L_{qF} the set of words corresponding to paths from state q to a final state. Two states $p, q \in Q$ are *indistinguishable* if $L_{pF} = L_{qF}$.

Notice that the above definition of indistinguishability among states is an extension to generalized automata of the corresponding definition for conventional automata (cf. [HU79]). The indistinguishability is an equivalence relation over the set of states that we indicate by $\sim$. Therefore we can construct a reduced DGA $\mathcal{A}'$ by contracting all states belonging to the same equivalence class. The edges of $\mathcal{A}'$ can be defined as follows. If $[q]$ denotes the equivalence class of state

q, we define $W([q])$ as the prefix set of the shortest words in $\bigcup_{p\sim q} W(p)$. Observe that, for any $w \in W([q])$ there exists at least a state $p \sim q$ such that $w \in W(p)$. Then, for any (p, w, p') in $\mathcal{A}$ there is edge $([q], w, [p'])$ in $\mathcal{A}'$. We omit the proof that the DGA $\mathcal{A}'$ is equivalent to $\mathcal{A}$. We say that an automaton is $\mathcal{I}$-*irreducible* if it has no indistinguishable states.

Now we give a definition for a transformation $\mathcal{S}$ corresponding to the second operation mentioned above. Let $\mathcal{A}$ be a DGA and r, s be two states of $\mathcal{A}$. We denote by L_{rs} the set of words over Σ that are labels of all paths from r to s in $\mathcal{A}$. We define transformation $\mathcal{S}$ that suppresses states in $\mathcal{A}$ preserving sets L_{rs} for any pair of states r, s not suppressed.

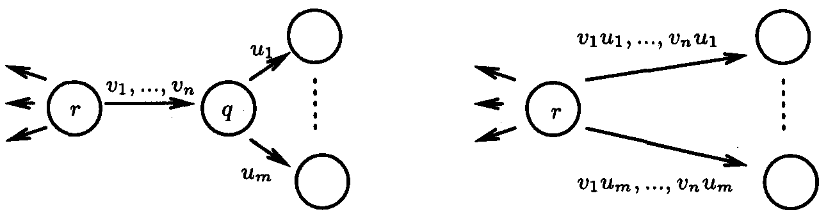
Given a state q of $\mathcal{A}$, transformation $\mathcal{S}$ computes a smaller DGA $\mathcal{S}(\mathcal{A}, q)$ from $\mathcal{A}$ by suppressing the state q and redefining all the edges that were incident in q. More precisely, $\mathcal{S}$ suppresses state q together with all its incident edges and, for any pair of edges (r, u, q) and (q, v, s) that were in $\mathcal{A}$, $\mathcal{S}$ defines a new edge (r, uv, s). Observe that, in order to preserve sets L_{rs} without compromising the finiteness of automaton $\mathcal{S}(\mathcal{A}, q)$, state q must not have self-loops. Since our final goal is to minimize a DGA, we are actually interested in transformations that reduce a DGA preserving the recognized language (i.e. preserving all sets L_{if} where i is the initial state and f is any final state): therefore we do not apply $\mathcal{S}$ both to i and to any final state. We give the following definition.

Definition 4. Given a DGA $\mathcal{A} = (\Sigma, Q, i, F, E)$. A state $q \in Q$ is a *superfluous state* for $\mathcal{A}$ if q is neither an initial nor a final state and it has no self-loops.

The set of all superfluous states for $\mathcal{A}$ will be denoted by *Super*(Q). We now formally define transformation $\mathcal{S}$.

Definition 5. Let $\mathcal{A} = (\Sigma, Q, i, F, E)$ be a DGA and $q \in Q$ be a superfluous state. Then $\mathcal{S}(\mathcal{A}, q) = (\Sigma, Q_q, i, F, E_q)$ is a (generalized) automaton where $Q_q = Q - \{q\}$ and $(r, u, s) \in E_q$ if either $(r, u, s) \in E$ or there exist $(r, u_1, q), (q, u_2, s) \in E$ such that $u_1 u_2 = u$.

For each $r \in Q_q$, the set $W_q(r)$ of words associated to r in the transformed automaton can be calculated starting from the sets $W(r)$ and $W(q)$ as follows. We split the set $W(r)$ in two disjoint subsets $W(r) = X(r, q) \cup \overline{X}(r, q)$ such that $X(r, q)$ contains the words that are labels of edges ending in state q and $\overline{X}(r, q)$ is its complement in $W(r)$. Then, we have: $W_q(r) = X(r, q) \cdot W(q) \cup \overline{X}(r, q)$.



Lemma 6. *Let $\mathcal{A}$ be a DGA and let q be a superfluous state for $\mathcal{A}$. The transformed automaton $\mathcal{S}(\mathcal{A}, q)$ is a DGA equivalent to $\mathcal{A}$.*

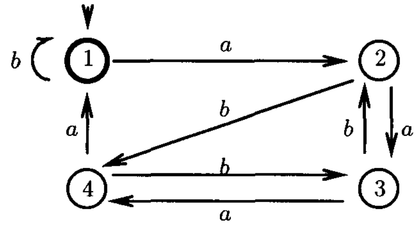
Proof. Let $\mathcal{A}_q = \mathcal{S}(\mathcal{A}, q) = (\Sigma, Q_q, i, F, E_q)$ be as in the Definition 5. The set $W_q(r)$ is a prefix set for any $r \in Q_q$ (see also Proposition 4.1 in [BP85]).

We now have to prove that $\mathcal{A}$ and $\mathcal{A}_q$ recognize the same language. Notice that, by construction, each edge (r, u, s) in $\mathcal{A}_q$ corresponds in $\mathcal{A}$ either to the same edge or to the path $\{(r, u_1, q), (q, u_2, s)\}$ where $u_1 u_2 = u$. Then, for any word $v \in \Sigma^*$, v is the label of an accepting path in $\mathcal{A}_q$ if and only if it is a label of an accepting path in $\mathcal{A}$. □

We say that an automaton without superfluous states is $\mathcal{S}$-*irreducible.*

Definition 7. A DGA is *irreducible* if it is both $\mathcal{I}$-irreducible and $\mathcal{S}$-irreducible.

We remark that irreducibility does not imply minimality. In fact the following DGA over the alphabet $\Sigma = \{a, b\}$ admits two equivalent DGA that are both irreducible but have a different number of states.



By suppressing state 3, we get an irreducible DGA that is is not minimal because a smaller irreducible DGA can be obtained by orderly suppressing states 2 and 4 from the initial automaton.

5 Irreducible DGA

In this section we consider the problem of calculating an irreducible DGA equivalent to a given one. In particular we will find conditions to apply transformation $\mathcal{S}$ to a given DGA in order to "suppress" as many as possible superfluous states.

Notice that, if p and q are both superfluous states of a given DGA $\mathcal{A}$ then p is not necessarily still a superfluous state for the transformed automaton $\mathcal{S}(\mathcal{A}, q)$. In other words, in general, the set of superfluous states of a DGA changes when it is reduced by transformation $\mathcal{S}$. We now establish the conditions under which two superfluous states p and q can both be suppressed. It is not too difficult to prove the following lemma, given here without proof for lack of space.

Lemma 8. *Let $\mathcal{A} = (\Sigma, Q, i, F, E)$ be a DGA and p, q be two superfluous states for $\mathcal{A}$ such that there is no cycle of length two between p and q. Then p and q are superfluous states for $\mathcal{S}(\mathcal{A}, q)$ and $\mathcal{S}(\mathcal{A}, p)$ respectively and $\mathcal{S}(\mathcal{S}(\mathcal{A}, q), p) = \mathcal{S}(\mathcal{S}(\mathcal{A}, p), q)$.*

Lemma 8 allows us to adopt the notation $\mathcal{S}(\mathcal{S}(\mathcal{A},q),p) = \mathcal{S}(\mathcal{S}(\mathcal{A},p),q) = \mathcal{S}(\mathcal{A},\{p,q\})$. We now want to investigate under which conditions this notation can be extended to any set $S = \{s_1, s_2, \ldots, s_h\} \subseteq Q$.

We indicate by $\mathcal{A}_i$ the DGA obtained from $\mathcal{A}$ by suppressing in order states $s_1, s_2, \ldots, s_i$ for $i = 1, \ldots, h$. Notice that the transformation

$$\mathcal{S}((\ldots\mathcal{S}(\mathcal{S}(\mathcal{A},s_1),s_2)\ldots),s_h) \tag{1}$$

can be realized only if, for any $i = 1, \ldots, h-1$, state s_{i+1} is an superfluous state for $\mathcal{A}_i$. We use the notation $\mathcal{S}(\mathcal{A},\{s_1,\ldots,s_h\}) = \mathcal{S}(\mathcal{A},S)$ to refer to expression (1): this will be justified later.

We recall that *Super*(Q) denotes the set of all superfluous states of $\mathcal{A}$. The following lemma characterizes those sets $S \subseteq Q$ for which $\mathcal{S}(\mathcal{A},S)$ can be calculated. The proof is in a short form for lack of space.

Lemma 9. *Let $\mathcal{A} = (\Sigma, Q, i, F, E)$ be a DGA and $S \subseteq Q$. Then $\mathcal{S}(\mathcal{A},S)$ can be calculated if and only if $S \subseteq$ Super(Q) and it induces an acyclic subgraph in $\mathcal{A}$.*

Proof. Observe that, given automata $\mathcal{A}$ and $\mathcal{S}(\mathcal{A},q)$, there is a cycle containing two states $r, s \neq q$ in $\mathcal{A}$ if and only if there is a cycle containing r and s in $\mathcal{S}(\mathcal{A},q)$. Then we can apply induction on the cardinality of S. The base of the induction is given by Lemma 8. □

Lemma 8 guaranties that the computation of DGA $\mathcal{S}(\mathcal{A},S)$ is independent of the order in which the states s_i's are suppressed from $\mathcal{A}$ and justifies this notation. As immediate consequence of Lemma 9 we get the following theorem. We recall that a DGA is irreducible if the set of its superfluous states *Super*$(Q) = \emptyset$. We refer to the subgraph induced by *Super*(Q) as $\mathcal{A}_{Super(Q)}$.

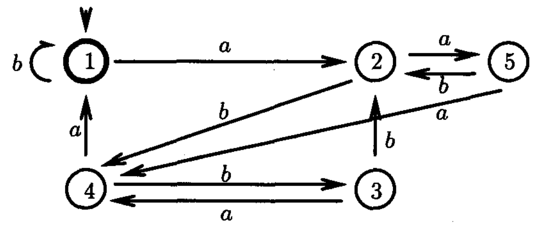
Theorem 10. *Let $\mathcal{A} = (\Sigma, Q, i, F, E)$ be a DGA and $S \subseteq$ Super(Q). The DGA $\mathcal{S}(\mathcal{A},S)$ is $\mathcal{S}$-irreducible if and only if S induces a maximal acyclic subgraph in $\mathcal{A}_{Super(Q)}$.*

Observe that, given an automaton $\mathcal{A}$, finding the set of states S of the above theorem is an NP-complete problem since it is strictly related to the problem "Given a direct graph, find the minimum number of states to be deleted so that resulting subgraph is acyclic" that is NP-complete (see [LY80]).

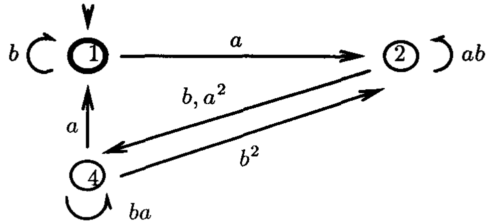
6 Minimal DGA

In this section we study the problem of finding a minimal DGA recognizing a given language. We recall that a minimal DGA does not have either indistinguishable or superfluous states. We start by remarking that, given a DGA, the procedure consisting of an $\mathcal{S}$-reduction followed by an $\mathcal{I}$-reduction in not equivalent to the procedure that inverts these two operations. This can be seen from the following example.

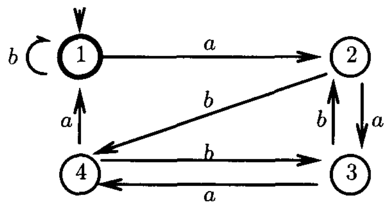
Consider the DGA $\mathcal{A}$ over the alphabet $\Sigma = \{a, b\}$ given below.

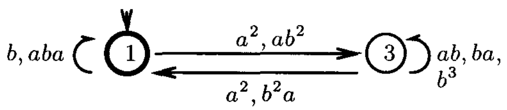


Observe that the subset $S = \{3,5\}$ induces in $\mathcal{A}$ a maximal acyclic subgraph so that $\mathcal{A}_1 = \mathcal{S}(\mathcal{A}, S)$ is an $\mathcal{S}$-irreducible DGA. $\mathcal{A}_1$ is represented below.



Since in $\mathcal{A}_1$ there are no indistinguishable states we conclude that $\mathcal{I}(\mathcal{A}_1) = \mathcal{A}_1$ is both an $\mathcal{S}$-irreducible and an $\mathcal{I}$-irreducible DGA equivalent to $\mathcal{A}$. Now observe, that states 3 and 5 are indistinguishable in $\mathcal{A}$ so that we can contract them in a unique state called 3. We obtain the $\mathcal{I}$-irreducible automaton $\mathcal{A}_2 = \mathcal{I}(\mathcal{A})$ given below on the left. Then the set of states $S' = \{2,4\}$ induces a maximal acyclic graph in $\mathcal{A}_2$ so that $\mathcal{A}_2' = \mathcal{S}(\mathcal{A}_2, S')$ is an $\mathcal{S}$-irreducible DGA. $\mathcal{A}_2'$ is represented by the graph given below on the right.





Notice that the two resulting DGA , $\mathcal{A}_1$ and $\mathcal{A}_2'$, are both $\mathcal{S}$- and $\mathcal{I}$-irreducible but they have a different number of states.

Theorem 11. *Let L be a regular language and let $\mathcal{N}$ be a minimal DGA recognizing L. If $\mathcal{M}$ is the equivalent minimal conventional deterministic automaton then there exists a set S of states of $\mathcal{M}$ such that $\mathcal{N} = \mathcal{S}(\mathcal{M}, S)$.*

Proof. Let $\mathcal{N} = (Q_\mathcal{N}, i_\mathcal{N}, F_\mathcal{N}, E_\mathcal{N})$ and $\mathcal{M} = (Q_\mathcal{M}, i_\mathcal{M}, F_\mathcal{M}, E_\mathcal{M})$. We prove that there exists a set $S \subseteq Q_\mathcal{M}$ such that $\mathcal{N} = \mathcal{S}(\mathcal{M}, S)$ by defining a mapping $\varphi : Q_\mathcal{N} \to Q_\mathcal{M}$ and showing that states in $Q_\mathcal{M}$ that have no counterimage in $Q_\mathcal{N}$ are superfluous states for $\mathcal{M}$.

The mapping φ is defined as follows: $i_{\mathcal{M}} = \varphi(i_{\mathcal{N}})$ and if $q_{\mathcal{N}} \in Q_{\mathcal{N}}$ and $q_{\mathcal{M}} \in Q_{\mathcal{M}}$ then $\varphi(q_{\mathcal{N}}) = q_{\mathcal{M}}$ if and only if there exists a word $w \in L_{i_{\mathcal{N}}q_{\mathcal{N}}} \cap L_{i_{\mathcal{M}}q_{\mathcal{M}}}$.

We first show that φ is actually a function over $Q_{\mathcal{N}}$. Since automata $\mathcal{N}$ and $\mathcal{M}$ are equivalent then, given $q_{\mathcal{N}} \in Q_{\mathcal{N}}$ there exists $q_{\mathcal{M}} \in Q_{\mathcal{M}}$ such that $\varphi(q_{\mathcal{N}}) = q_{\mathcal{M}}$. Such state $q_{\mathcal{M}}$ is unique. Suppose that there exists also $p_{\mathcal{M}} \in Q_{\mathcal{M}}$ such that $\varphi(q_{\mathcal{N}}) = p_{\mathcal{M}}$: then, by the definition of φ, there exist two words u, v such that paths $[i_{\mathcal{N}}, u, q_{\mathcal{N}}]$ and $[i_{\mathcal{N}}, v, q_{\mathcal{N}}]$ are in $\mathcal{N}$ and paths $[i_{\mathcal{M}}, u, q_{\mathcal{M}}]$ and $[i_{\mathcal{M}}, v, p_{\mathcal{M}}]$ are in $\mathcal{M}$. But the equivalence of $\mathcal{N}$ and $\mathcal{M}$ implies that $q_{\mathcal{M}}$ and $p_{\mathcal{M}}$ are indistinguishable and this contradicts the hypothesis that $\mathcal{M}$ is minimal.

Then we can define a set $S = Q_{\mathcal{M}} - \varphi(Q_{\mathcal{N}})$. Notice that S contains all states of $\mathcal{M}$ that do not correspond to any state of $\mathcal{N}$. We now prove that $\mathcal{M}' = \mathcal{S}(\mathcal{M}, S)$ is defined, that is it satisfies the conditions of Lemma 9, and that $\mathcal{N} = \mathcal{M}'$. Let us first observe that $F_{\mathcal{M}} \subseteq \varphi(Q_{\mathcal{N}})$: therefore the set S does not contains both the initial state $i_{\mathcal{M}}$ and the set of final states of $\mathcal{M}$. We now show that S induces an acyclic subgraph in $\mathcal{M}$.

Suppose that in $\mathcal{M}$ there is a cycle $[s, u, s]$ whose states are all in S. Let v, w be two words such that the paths $[i_{\mathcal{M}}, v, s], [s, w, f_{\mathcal{M}}]$ are in $\mathcal{M}$ where $f_{\mathcal{M}} \in F_{\mathcal{M}}$. The words $vu^n w \in L$ for all integers $n \geq 0$: therefore in $\mathcal{N}$ for any n there exists a path $[i_{\mathcal{N}}, vu^n w, f_{\mathcal{N}}]$, where $f_{\mathcal{N}} \in F_{\mathcal{N}}$. Since $Q_{\mathcal{N}}$ is a finite set, there exist infinite values of n for which path $[i_{\mathcal{N}}, uv^n w, f_{\mathcal{N}}]$ in $\mathcal{N}$ contains a cycle and it can be split as paths $[i_{\mathcal{N}}, x, r], [r, y, r], [r, y, r], \ldots, [r, y, r], [r, z, f_{\mathcal{N}}]$ that is $vu^n w = xy^k z$ for a suitable value of k.

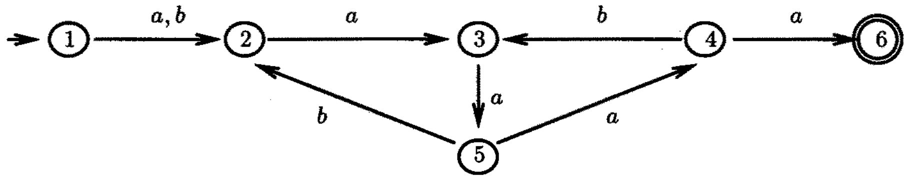
Therefore we can choose k, h in a way that $k \geq h$, $|xy^h| \geq |v|$ and$|y^{k-h}z| \geq |w|$ while $xy^k z = vu^n w$. We observe that $|xy^h| \leq |vu^n|$ otherwise $|xy^k z| = |xy^h| + |y^{k-h}z| > |vu^n w|$ contradicting the hypothesis. Since the word xy^h is a prefix of $vu^n w$ (that belongs to L) then there exists a state s' in $\mathcal{M}$ such that the path $[i_{\mathcal{M}}, xy^h, s']$ is in $\mathcal{M}$. From the definition of φ we have: $s' = \varphi(r)$ that is $s' \in \varphi(Q_{\mathcal{N}})$ (and therefore s' does not belongs to S). Moreover, notice that since $|v| \leq |xy^h| \leq |vu^n|$, then the state s' is a state in the cycle $[s, u, s]$ in $\mathcal{M}$. But this implies that $s' \in S$ contradicting what we stated before.

It remains to show that $\mathcal{N} = \mathcal{M}'$. We already know that $\mathcal{N}$ and $\mathcal{M}'$ are equivalent, $i_{\mathcal{M}'} = i_{\mathcal{M}} = \varphi(i_{\mathcal{N}})$ and that map φ is defined onto the set of states $Q_{\mathcal{M}'}$ of $\mathcal{M}'$ that is $Q_{\mathcal{M}'} = Q_{\mathcal{M}} - S = \varphi(Q_{\mathcal{N}})$. Mapping φ is actually a bijection from $Q_{\mathcal{N}}$ in $Q_{\mathcal{M}'}$. In fact if there exist two states $p, q \in Q_{\mathcal{N}}$ such that $\varphi(p) = \varphi(q)$ then $|Q_{\mathcal{M}'}| < |Q_{\mathcal{N}}|$ and this contradicts the hypothesis of $\mathcal{N}$ minimal DGA. □

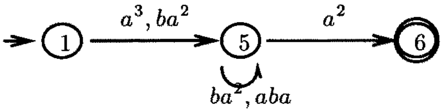
From this theorem we get a procedure to compute the size n of a minimal DGA recognizing a given language L: we calculate the minimal conventional deterministic automaton $\mathcal{M}$ for L and then calculate a maximal set of states S of $\mathcal{M}$ that induces a maximal acyclic subgraph. Then $n = |Q_{\mathcal{M}}| - |S|$. This solves the corresponding problem studied by Hashiguchi in [H91] for the deterministic setting. Notice that, the maximal length of the labels in the edges of a minimal DGA $\mathcal{N}$ (referred as $D(\mathcal{N})$ by Hashigushi in [H91]), is equal, at most, to the number of states suppressed in $\mathcal{M}$ plus 1, that is at most n, where n is the number of states of the minimal deterministic automaton recognizing L. Differently from the case of conventional deterministic automata, it holds the following.

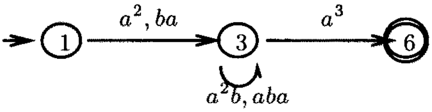
Theorem 12. *Given a language L, there is not a unique minimal DGA recognizing L.*

The proof is given by the following example. Let L be the language recognized by the minimal deterministic automaton $\mathcal{A}$ represented below.



In $\mathcal{A}$ there are two maximal superfluous sets, $S_1 = \{2,3,4\}$ and $S_2 = \{2,4,5\}$. By suppressing S_1 in $\mathcal{A}$ we obtain the minimal DGA for L given below on the left. In the same way, by suppressing S_2 in $\mathcal{A}$ we obtain another minimal DGA for L different from the previous one that is given below on the right.





As immediate consequence of Theorems 10 and 11, we obtain a procedure to find all minimal DGA equivalent to a given deterministic automaton $\mathcal{A}$. We take the minimal (conventional) deterministic automaton $\mathcal{M}$ equivalent to $\mathcal{A}$ and compute all maximal sets among all superfluous sets that induce maximal acyclic subgraphs in $\mathcal{M}_{Super(Q_{\mathcal{M}})}$. All minimal DGA equivalent to $\mathcal{A}$ can be computed by applying an $\mathcal{S}$-reduction to $\mathcal{M}$ with respect to such sets.

We finish the section by remarking that the inverse operation of transformation $\mathcal{S}$ (that is breaking edges with "long" labels and create a sequence of edges with "shorter" labels) is easy to define. Given the edge $(p, w_1 w_2 \dots w_n, q)$ we can insert states $r_1, r_2, \dots, r_{n-1}$ and edges $(p, w_1, r_1), (r_1, w_2, r_2), \dots, (r_{n-1}, w_n, q)$. Therefore, to solve the problem of minimizing a given DGA $\mathcal{A}$, we apply this inverse operation to $\mathcal{A}$ until we obtain a conventional deterministic automaton $\mathcal{A}'$ and then we minimize $\mathcal{A}'$. Finally we apply Theorem 11.

7 Final discussions and further work

It is well known that, in the case of conventional automata there is an exponential gap in the complexity of representation between the non-deterministic and deterministic versions. In fact, consider the language $L_n = (a+b)^* a (a+b)^{n-1}$, for any integer n: the minimal deterministic automaton for L_n has exactly 2^n states while the corresponding non-deterministic one has $n+1$ states. We notice that such discrepancy in succinctness between non-deterministic and deterministic versions still holds inside the model of GA. In fact, the minimal (conventional) deterministic automaton for L_n has exactly 2^{n-1} final states (therefore not superfluous) that will be necessarily also in any minimal DGA. On the other hand the minimal (non-deterministic) GA has only two states for any n.

The above example suggests that, if the minimal conventional deterministic automaton has "too many" final states, then the corresponding GA cannot be reduced too much. A further direction for this work is then to investigate about the succinctness in the case of automata with only one final state. This is related with the decomposition of a regular language in unitary components ([E74]).

As final observation, notice that the transformation S can be defined as well for non-deterministic GA. It still gives an equivalent GA but in general we do not know whether there exists a procedure that compute a minimal non-deterministic (generalized) automaton.

Acknowledgments

We are deeply grateful to Antonio Restivo for several discussions that inspired this paper and for his helpful comments and hearty encouragements. We also thank Pino Italiano and Sergio Salemi for some useful suggestions and an anonymous referee for his careful reading.

References

[AHU] A. Aho, J. E. Hopcroft and J. D. Ullman. *The Design and the Analysis of Computer Algorithm.* Addison-Wesley, Reading, MA 1974.

[BP85] J. Berstel and D. Perrin *Theory of Codes* Academic Press, 1985.

[E74] S. Eilenberg. *Automata, Languages and Machines* Vol.A, Academic Press,1974.

[GH94] N. Globerman and D. Harel. Complexity results for multi-pebble automata and their logics. In Proc. *(ICALP'94).* S. Abiteboul and E. Shamir Eds. Springer-Verlag, n.820, pp. 73–82 (1994).

[H91] K. Hashigushi. Algorithms for determining the smallest number of nonterminals (states) sufficient for generating (accepting) a regular language. In Proc. *(ICALP'91).* J.Leach Albert *et al.* Eds. Springer-Verlag, n.510, pp. 641–648 (1991).

[HU79] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, MA 1979.

[JR91] Tao Jiang and B. Ravikumar. Minimal NFA problems are hard. In Proc. *(ICALP'91).* J.Leach Albert *et al.* Eds. Springer-Verlag, n.510, pp. 629–640 (1991).

[LY80] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Comp. and System Science*, **20**, pp. 219–230 (1980).

[P90] D. Perrin. Finite Automata. In *Handbook of Theoretical Computer Science.* J. Van Leeuwen ed. Elsevier, Vol.B, pp. 1–57, 1990.

[WK94] A. Weber and R. Klemm. Economy of description for single-valued transducers. In Proc. *(STACS'94).* P. Enjalbert *et al.* Eds. Springer-Verlag, n.775, pp. 607–618 (1994).