



**HAL**  
open science

## Note: Regular languages with no conflicts (forbidden pairs) are regular but have exponential size DFA

Alexis Cornet, Christian Laforest

### ► To cite this version:

Alexis Cornet, Christian Laforest. Note: Regular languages with no conflicts (forbidden pairs) are regular but have exponential size DFA. [Research Report] LIMOS (UMR CNRS 6158), université Clermont Auvergne, France 2016. hal-01402160

**HAL Id: hal-01402160**

**<https://hal.science/hal-01402160v1>**

Submitted on 24 Nov 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Note: Regular languages with no conflicts (forbidden pairs) are regular but have exponential size DFA

Alexis Cornet\*

Christian Laforest<sup>†</sup>

November 24, 2016

## Abstract

Many graph problems (Dominating sets, Steiner tree, etc.) are hard to optimize but finding a solution, regardless of its size, is in general trivial and polynomial. In recent papers, several authors introduced *conflicts*, that are pairs of edges or vertices that cannot be both in a solution. These new constraints drastically improve the hardness: They proved that in most cases, deciding if there exists a solution is now NP-complete. In this short note we transport this problematic of conflicts in language theory. Despite the negative results obtained in the field of discrete optimization, we show that a language  $\mathcal{L}^c$  composed by the words of any regular language  $\mathcal{L}$  that do not contain pairs of conflicting symbols is still regular. However, we show that the DFA accepting  $\mathcal{L}^c$  that we construct has a non polynomial number of states. Nevertheless, we prove that this drawback cannot be avoided in general, even if a symbol is in conflict with at most another one, and seems to represent the price to pay for dealing with conflicts in regular languages.

## 1 Introduction and notations

We start by reminding some classical concepts in formal languages (see [10] for details and complements).  $\Sigma$ , a finite set of *symbols*, is called an *alphabet*. A *word* is a finite sequence of symbols. The *empty word* is denoted by  $\lambda$ . A *language*  $\mathcal{L}$  is a (finite or infinite) set of words. A *DFA* (Deterministic Finite Automaton)  $M$  is composed of a *finite* set  $Q$  of *states* including the (unique) *initial* state. The set of *final* (or *accepting*) states is noted  $F$  ( $F \subseteq Q$ ). The *transition function*  $\delta$  is said *deterministic* because for any  $p \in Q$  and any  $c \in \Sigma$  there exists a *unique* state  $q \in Q$  ( $p$  and  $q$  can be equal) such that  $\delta(p, c) = q$ . If  $w = a_1 \dots a_k$  is a word and  $p$  a state, we note  $\delta^*(p, w)$  the (unique) state in which  $M$  is after treating the sequence of symbols in the order of  $w$ , starting from state  $p$ . A word  $w$  is *accepted* by  $M$  of initial state  $q_0$  if  $\delta^*(w, q_0) \in F$ . The set of words accepted by  $M$  is called the language *accepted* by  $M$ :  $\tau(M) = \{w : \delta^*(w, q_0) \in F\}$ . A classical result states that there exists a DFA  $M$  accepting  $\mathcal{L}$  if and only if  $\mathcal{L}$  is *regular*.

**Introducing conflicts.** Let  $M$  be a DFA,  $\Sigma$  its alphabet and  $\mathcal{L}$  its accepted regular language. In this paper we add new constraints, named *conflict graph*  $G$ .

In Section 2,  $G$  is a *graph* on symbols of  $\Sigma$ . A *conflict* (edge of  $G$ ) denotes here a *forbidden pair of symbols*, that cannot be both present into a same word. A word contains a conflict if it contains two symbols linked by an edge of  $G$ . We show that the language  $\mathcal{L}^c$  composed of words of  $\mathcal{L}$  with no conflict is regular. To prove this result we transform  $M$  into a new DFA  $\mathcal{M}^c$  accepting  $\mathcal{L}^c$ . However the size of  $\mathcal{M}^c$  is exponential in the size of the initial instance ( $M$  and  $G$ ). A natural question is then: Is it possible to construct a polynomial size DFA  $\mathcal{M}^c$  accepting  $\mathcal{L}^c$ ? In Section 3 we prove that it is not the case by analyzing a particular instance,  $\mathcal{L}$  and  $G(\Sigma)$ , and proving that any DFA accepting  $\mathcal{L}^c$  must contain an exponential number of states.

\*LIMOS, Université Clermont Auvergne, France

<sup>†</sup>LIMOS, ISIMA, Université Clermont Auvergne, France. Corresponding author. christian.laforest@isima.fr

A motivation for this work is theoretical. In recent papers, several authors studied classical problems on graph with the additional constraints given by pairs of vertices or edges simultaneously forbidden in a solution. For example, in [2, 6, 7, 3, 4, 5, 11], the authors try to find paths between two vertices (or spanning trees), containing no pairs of conflicting edges. In [8, 9] the existence of hamiltonian paths or cycles containing no conflicting pairs of edges is questioned. In [1] the authors try to find a connected vertex cover or a Steiner tree or a dominating set with no pair of conflicting vertices. Despite a few constructions in very particular cases, most of results are NP-completeness theorems to decide whether there is a solution (a path, a tree, a connected vertex cover, etc.), regardless the size of this solution. Hence, in graphs many of these problems with conflicts are very hard. This motivates us to export this notion of conflicts outside the graph theory. In the field of formal language theory, our results show good news: Even with conflicts on symbols, a regular language remains regular (Section 2). Moreover our proof is constructive. However, the bad news is that the DFA we propose does not have a polynomial size. We show in Section 3 that this cannot be avoided in general. Hence, the exponential increasing of the size of the DFA seems to be the price to pay for introducing conflicts in regular languages.

Regular languages have received a lot of attention. It is a remarkable family of languages, stable over operations like: Union, intersection, concatenation, etc. A list of operations that transform a regular language into another regular language can be found in Chapter 4 (some of them are in exercises) of [10]. The transformation we propose in this note is original (to our knowledge) since it is *conditional*: if a symbol is in a word *then* another symbol (in conflict with the first) cannot be in it. Our work can lead to the design of solutions to solve practical problems like: In a text, how to find words containing no given pairs of symbols? DFA are good tools for exploring a text. This is also why we concentrate on DFA and its size. Moreover, the formalism we use (conflict graph) allows to express these constraints in a compact and readable form.

## 2 Construction of exponential size DFA accepting regular languages with no conflicts between symbols

Let  $\mathcal{L}$  be any regular language on alphabet  $\Sigma$ . Let  $M$  be a DFA accepting  $\mathcal{L}$  ( $\tau(M) = \mathcal{L}$ ). In this Section we introduce  $G(\Sigma)$  that we call a *conflict graph*, a non oriented graph whose vertices are symbols of  $\Sigma$ . Each edge  $uv$  of  $G(\Sigma)$  is called a *conflict* between symbols  $u$  and  $v$  of  $\Sigma$ . Given  $\mathcal{L}$  and  $G(\Sigma)$  we note  $\mathcal{L}^c$  the language composed of all the words of  $\mathcal{L}$  that do *not* contain two symbols that are in conflict in  $G(\Sigma)$ :  $\mathcal{L}^c = \{w = a_1 \dots a_k : w \in \mathcal{L} \text{ and } a_i a_j \notin G(\Sigma) (\forall i, j)\}$ . A first question here is: Is  $\mathcal{L}^c$  a regular language? The input of our problem in this Section is a DFA  $M$  on an alphabet  $\Sigma$ , accepting a regular language  $\mathcal{L}$ , and a conflict graph  $G(\Sigma)$ . From this instance, we are going to construct a DFA  $\mathcal{M}^c$  accepting  $\mathcal{L}^c$ , hence proving that  $\mathcal{L}^c$  is regular. We will then show that its size is exponential compared to the one of  $M$ .

**Construction of  $\mathcal{M}^c$ .** Let  $\mathcal{STAB}$  be the family of *stables* of  $G(\Sigma)$ , i.e. sets of symbols with no conflict between them. Create a new initial state  $q'_0$  for  $\mathcal{M}^c$ , that is also final if and only if  $q_0$  is final in  $M$ . Now, for each stable  $S$  of  $\mathcal{STAB}$ , create a copy noted  $M_S$  of  $M$ . Each copy of the initial state  $q_0$  in each  $M_S$  is now non initial (only  $q'_0$  is initial in  $\mathcal{M}^c$ ). Each final state of  $M_S$  remains final in the construction. We also add a new non final state TRASH. We modify now transitions of these copies to get  $\delta^c$ , the final transition function of  $\mathcal{M}^c$ . Obviously, for each  $c \in \Sigma$ ,  $\delta^c(\text{TRASH}, c) = \text{TRASH}$  (i.e. create loops on state TRASH without possibilities to output). Concerning the new initial state  $q'_0$  we have: For each  $c \in \Sigma$ ,  $\delta^c(q'_0, c) = p$  where  $p$  is the copy of the state  $\delta(q_0, c)$  in the copy  $M_{\{c\}}$  (set  $\{c\} \in \mathcal{STAB}$  is a particular stable with only one element).

Now, the global and informal idea of the transformation is that the treatment of a word  $w$  by  $\mathcal{M}^c$  will be in a state of  $M_S$  when the set of symbols already treated on  $w$  is  $S$ . Thus, copy  $M_S$  is used as “memory” of the already treated symbols. This mechanism permits to “output” to state TRASH when a conflicting symbol is encountered. Otherwise, the treatment of  $w$  continues in copies  $M_S$  of  $\mathcal{M}^c$  with transitions similar to the ones of  $M$ . Let us give details.

$\mathcal{M}^c$  “remains” in the copy  $M_S$  while the current symbol treated is in  $S$ : For any  $c \in S$  and any state  $p$

of  $M_S$ ,  $\delta^c(p, c) = q$  where  $q$  is the copy of state  $\delta(p, c)$  in  $M_S$ .

If the automaton is in state  $p$  and the current read symbol  $c$  is in conflict with a symbol of  $S$  then the automaton must go to state TRASH (this represents the detection of a conflict of  $c$  with a previously treated symbol): For all  $c$  in conflict with any symbol of  $S$  and for any state  $p$  of  $M_S$ ,  $\delta^c(p, c) = \text{TRASH}$ .

Now, the last case to take into account is when the current symbol read  $c$  is not in  $S$  and is not in conflict with the elements of  $S$ . In this case,  $S' = S \cup \{c\}$  is a stable ( $S' \in \mathcal{STAB}$ ). Hence, for any state  $p$  of  $M_S$ , and for any symbol  $c$  with  $S' = S \cup \{c\} \in \mathcal{STAB}$  the transition is  $\delta^c(p, c) = q$  where  $q$  is the copy of state  $\delta(p, c)$  (of  $M$ ) in  $M_{S'}$ . This ends the construction of  $\mathcal{M}^c$  that is a DFA (indeed  $\delta^c(p, c)$  exists and is unique for each state of  $\mathcal{M}^c$  and each symbol  $c \in \Sigma$ ). Moreover,  $\mathcal{M}^c$  has one initial state  $q'_0$ . We must prove now that  $\mathcal{M}^c$  accepts  $\mathcal{L}^c$ .

**$\mathcal{M}^c$  accepts  $\mathcal{L}^c$ .** If  $\lambda \in \mathcal{L}^c$ , then  $\lambda \in \mathcal{L}$  (as  $\mathcal{L}^c \subseteq \mathcal{L}$ ). Thus,  $\lambda$  is accepted by  $M$ , i.e.  $q_0$  is a final state of  $M$  and by construction  $q'_0$  is also final in  $\mathcal{M}^c$ . Hence  $\lambda$  is also accepted by  $\mathcal{M}^c$ .

Let  $w = a_1 \dots a_k \neq \lambda$  be any word on alphabet  $\Sigma$  and  $S$  its set of symbols. We note  $R_i$  the set of symbols of the prefix of length  $i$  of  $w$ :  $a_1 \dots a_i$ .

If  $w \in \mathcal{L}^c$  then, by definition,  $S$  is a stable of the conflict graph. Hence, by construction, the treatment of  $w$  by  $\mathcal{M}^c$  ends into a state of the copy  $M_S$  of  $\mathcal{M}^c$ : Indeed, after the treatment of  $a_1$  the current state  $p_1$  is one of  $M_{R_1} = M_{\{a_1\}}, \dots$ , after the treatment of  $a_i$  the current state  $p_i$  is one of  $M_{R_i}, \dots$ , after the treatment of the last symbol  $a_k$  of  $w$ , the current state  $p_k$  is one of  $M_{R_k} = M_S$ . As there is no conflicting symbols in  $w$ , the automaton never goes into TRASH state. Now, let us note  $q_i$  the state of  $M$  corresponding to the state  $p_i$  in the copy  $M_{R_i}$ . If  $w$  is treated by  $M$  then  $\delta^*(q_0, w) \in F$  since  $w \in \mathcal{L}^c \subseteq \mathcal{L}$ . During this treatment of  $w$ , the sequence of states is  $q_1, \dots, q_k$ , hence  $q_k$  is final in  $M$  and, by construction, its copy  $p_k$  into  $M_S$  is final in  $\mathcal{M}^c$ . Word  $w$  is then accepted by  $\mathcal{M}^c$ .

Let us consider now the reverse situation. Suppose that  $w$  is accepted by  $\mathcal{M}^c$ . This means that  $S$  is a stable of the conflict graph (otherwise  $\delta^{c^*}(q'_0, w) = \text{TRASH}$  that is non final). Hence, by construction, the final state  $\delta^{c^*}(q'_0, w)$  is in  $M_S$ . Using again the notations of the previous paragraph, and by construction of  $\mathcal{M}^c$ , if  $w$  is treated by  $M$ , the sequence of states of the treatment is  $q_1, \dots, q_k$ . As  $p_k = \delta^{c^*}(q'_0, w)$  is final and is the copy of  $q_k$  into  $M_S$ ,  $q_k$  is final into  $M$ , thus  $w$  is accepted by  $M$  and then  $w \in \mathcal{L}$ . Moreover as there is no conflicting pair of symbols,  $w \in \mathcal{L}^c$ .

**$\mathcal{M}^c$  has an exponential size.** Let us compare the size of  $\mathcal{M}^c$  to the size of the instance ( $M$  and  $G(\Sigma)$ ).  $\mathcal{M}^c$  is composed of two new states  $q'_0$ , TRASH and  $|\mathcal{STAB}|$  copies of  $M$ , where some of the transitions are redirected between copies. The size is then  $O(|\mathcal{STAB}| \cdot |M|)$ . But as each element of  $\mathcal{STAB}$  is a subset of  $\Sigma$ ,  $|\mathcal{STAB}| \leq 2^{|\Sigma|}$ . The size of  $\mathcal{M}^c$  is then  $O(|M| \cdot 2^{|\Sigma|})$ . This generic construction leads to an exponential size automaton.

This construction can be refined and can lead to a smaller (but still non polynomial in general) size automaton in certain situations. For that, just take into account the stables on the set of symbols of  $\Sigma$  that are in at least a conflict (the others are "compatible" with any symbol of  $\Sigma$ ) to make the copies of  $M$  (just care about the symbols that can lead to a conflict; The others are neutral).

### 3 Exponential size DFA cannot be avoided

In Section 2 we proved that  $\mathcal{L}^c$  is regular by constructing a DFA accepting it. But this automaton has an exponential size in general. In this Section we prove that this drawback cannot be avoided: For some instances, i.e.  $\mathcal{L}$  and  $G(\Sigma)$ , the smaller DFA  $\mathcal{M}^c$  for  $\mathcal{L}^c$  necessarily has an exponential size, even if each symbol is in a conflict with at most another one.

Let  $\Sigma_k$  be an alphabet on  $2k$  symbols noted as integers to simplify:  $\Sigma_k = \{1, 2, \dots, 2k\}$ . The special conflict graph  $G(\Sigma_k)$  that we consider here has  $\Sigma_k$  as set of vertices and edges (i.e. conflicts) constitute a *perfect matching*: An edge between symbol  $2i - 1$  and  $2i$  for all  $i = 1, \dots, k$ , i.e. a conflict between 1 and 2, a conflict between 3 and 4, etc. a conflict between  $2k - 1$  and  $2k$ . The particular language  $\mathcal{L}_k$  considered here is the set of *all possible words* on alphabet  $\Sigma_k$ .  $\mathcal{L}_k$  is regular since it is accepted by a simple DFA  $M_k$

composed of a unique state  $q_0$  (initial and final) with:  $\delta(q_0, c) = q_0$  for all  $c \in \Sigma_k$ . We note  $\mathcal{L}^c$  the resulting language from  $\mathcal{L}_k$  and  $G(\Sigma_k)$ .

We describe a DFA  $\mathcal{M}^c$  for  $\mathcal{L}^c$  whose transition function is  $\delta^c$ . Create a state  $q_S$  for each stable  $S \in \mathcal{STAB}$ ; This state  $q_S$  is final. Create an initial state  $q_0$  that is also final (since  $\lambda \in \mathcal{L}_k$ ). This means that all the states of  $\mathcal{M}^c$  are final, except a new one TRASH with:  $\delta^c(\text{TRASH}, c) = \text{TRASH}$  for all  $c \in \Sigma_k$ .

The transitions follow the idea of the generic construction of Section 2. Let  $q_S$  be any state.

- For any  $c \in S$ ,  $\delta^c(q_S, c) = q_S$ .
- For any  $c \notin S$  and  $c$  in conflict with a symbol of  $S$ ,  $\delta^c(q_S, c) = \text{TRASH}$ .
- For any  $c \notin S$  and  $c$  in conflict with no symbol of  $S$ ,  $\delta^c(q_S, c) = q_{S \cup \{c\}}$  ( $S \cup \{c\}$  is a stable).

It is not difficult to see that  $\mathcal{M}^c$  is a DFA accepting  $\mathcal{L}^c$ . Indeed if a word  $w$  is in  $\mathcal{L}^c$  then it contains no conflicting pairs of symbols and its treatment by  $\mathcal{M}^c$  will not end in TRASH, hence  $w$  is accepted. Now, conversely, if  $w$  is accepted by  $\mathcal{M}^c$ , this means that its treatment ends in any state, except TRASH, hence, does not contain any conflicting pairs of symbols, i.e. is in  $\mathcal{L}^c$ . Moreover  $\mathcal{M}^c$  contains more that  $|\mathcal{STAB}|$  states. But  $|\mathcal{STAB}|$  contains at least the  $2^k$  sets, namely the ones that consist in selecting one extremity of each of the  $k$  edges of  $G(\Sigma_k)$ . Hence, the size of  $\mathcal{M}^c$  is not polynomial. However, one could argue that a smaller (and potentially polynomial) size automaton accepting  $\mathcal{L}^c$  could exist. We show now that it cannot be the case by proving that  $\mathcal{M}^c$  is minimal, i.e. is a minimal size DFA accepting  $\mathcal{L}^c$ .

For that we need to remind a few classical notions. Two states  $p$  and  $q$  are *distinguishable* in  $\mathcal{M}^c$  if there exists a word  $w$  on  $\Sigma_k$  such that  $\delta^{c*}(p, w)$  is final and  $\delta^{c*}(q, w)$  is *not* final, or vice versa. Following classical results (see the textbook [10] for example),  $\mathcal{M}^c$  is minimal if each pair of states are distinguishable. In the following we note  $F^c$  the set of final states of  $\mathcal{M}^c$ , i.e. all the states except TRASH.

First, TRASH is distinguishable from all the other states that are all final. Indeed  $\delta^c(q_0, 1) = q_{\{1\}} \in F^c$  and  $\delta^c(\text{TRASH}, 1) = \text{TRASH} \notin F^c$ . For any other state  $q_S$ , let  $c$  be any symbol in  $S$ ; We have:  $\delta^c(q_S, c) = q_S \in F^c$  and  $\delta^c(\text{TRASH}, c) = \text{TRASH} \notin F^c$ .

Now, let us consider the particular case of  $q_0$ . Let  $q_S$  be any other state. Let  $c$  be any symbol in conflict with a symbol of  $S$ . Then  $\delta^c(q_0, c) = q_{\{c\}} \in F^c$  and  $\delta^c(q_S, c) = \text{TRASH} \notin F^c$ .

Let  $q_S$  be any state (different of  $q_0$  and TRASH). Previous results shown that  $q_S$  is distinguishable of  $q_0$  and TRASH. Let  $q_R$  be any other state. First case: If  $S$  contains a symbol  $c$  and  $R$  contains a symbol in conflict with  $c$  then  $\delta^c(q_S, c) = q_S \in F^c$  and  $\delta^c(q_R, c) = \text{TRASH} \notin F^c$ . Second case: Symbols of  $S$  are not in conflict with the ones of  $R$ . As  $S \neq R$ , suppose that  $|R| < |S|$  (the proof for the case  $|S| < |R|$  is similar) and let  $c$  be any symbol of  $S$  not in  $R$ . Let  $d$  be the (unique) symbol in conflict with  $c$ . This means that  $d \notin R$ , hence  $R \cup \{d\}$  is a stable and  $\delta^c(q_R, d) = q_{R \cup \{d\}} \in F^c$  and  $\delta^c(q_S, d) = \text{TRASH} \notin F^c$ .

We proved that  $\mathcal{M}^c$  cannot be minimized since each pair of states are distinguishable. Hence, by a classical result (see [10]), there is no smaller DFA accepting  $\mathcal{L}^c$ .

## References

- [1] DELBOT, F., LAFOREST, C., AND PHAN, R. Hardness Results and Approximation Algorithms for Discrete Optimization Problems with Conditional and Unconditional Forbidden Vertices. Research report, HAL, <https://hal.archives-ouvertes.fr/hal-01257820>, Jan. 2016.
- [2] GABOW, H. N., MAHESHWARI, S. N., AND OSTERWEIL, L. J. On two problems in the generation of program test paths. *Software Engineering, IEEE Transactions on*, 3 (1976), 227–231.
- [3] KANTÉ, M. M., LAFOREST, C., AND MOMÈGE, B. An exact algorithm to check the existence of (elementary) paths and a generalisation of the cut problem in graphs with forbidden transitions. In *SOFSEM 2013* (2013), pp. 257–267.
- [4] KANTÉ, M. M., LAFOREST, C., AND MOMÈGE, B. Trees in graphs with conflict edges or forbidden transitions. In *TAMC 2013, Hong Kong* (2013), pp. 343–354.

- [5] KANTÉ, M. M., MOATAZ, F. Z., MOMÈGE, B., AND NISSE, N. Finding paths in grids with forbidden transitions. In *WG 2015* (2015), pp. 154–168.
- [6] KOLMAN, P., AND PANGRÁC, O. On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics* 157, 13 (2009), 2871–2876.
- [7] KOVÁČ, J. Complexity of the path avoiding forbidden pairs problem revisited. *Discrete Applied Mathematics* 161, 10 (2013), 1506–1512.
- [8] LAFOREST, C., AND MOMÈGE, B. Some hamiltonian properties of one-conflict graphs. In *IWOCA 2014* (2014), pp. 262–273.
- [9] LAFOREST, C., AND MOMÈGE, B. Nash-williams-type and chvátal-type conditions in one-conflict graphs. In *SOFSEM 2015* (2015), pp. 327–338.
- [10] LINZ, P. *An introduction to formal languages and automata*. Jones and Barlett, 2006 (fourth edition).
- [11] YINNONE, H. On paths avoiding forbidden pairs of vertices in a graph. *Discrete applied mathematics* 74, 1 (1997), 85–92.