Fixed Point Approach to Commutation of Languages*

Karel Culik II

Department of Computer Science and Engineering
University of South Carolina
Columbia, 29008 S.C., USA
kculik@sc.rr.com

Juhani Karhumäki

Department of Mathematics and
Turku Centre for Computer Science
University of Turku
20014 Turku, FINLAND
karhumak@cs.utu.fi

Petri Salmela

Department of Mathematics and
Turku Centre for Computer Science
University of Turku
20014 Turku, FINLAND
pesasa@utu.fi

Turku Centre for Computer Science
TUCS Technical Report No 582
December 2003

ISBN 952-12-1286-1
ISSN 1239-1891

Abstract

We show that the maximal set commuting with a given regular set – its centralizer – can be defined as the maximal fixed point of a certain language operator. However, an infinite number of iterations might be needed even in the case of finite languages.

Keywords: Commutation, language, fixed point, combinatorics of words

TUCS Laboratory
Discret Mathematics for Information Technology

*Supported by the Academy of Finland under grant 44087

1 Introduction

The commutation of two elements in an algebra is among the most natural operations. In the case of free semigroups, i.e., words, it is easily and completely understood: two words commute if and only if they are powers of a common word, see, e.g., [11]. For the monoid of languages, even for the finite languages the situation changes drastically. Many natural problems are poorly understood and are likely to be very difficult. For further details we refer in general to [3], [9] or [6] and in connection to complexity issues to [10] and [4].

Commutation of languages X and Y means that the equality $XY = YX$ holds. It is an equality on sets, however, to verify it one typically has to go to the level of words. More precisely, for each $x \in X$ and $y \in Y$ one has to find $x' \in X$ and $y' \in Y$ such that $xy = y'x'$. In a very simple setting this can lead to nontrivial considerations. An illustrative (and simple) example is a proof that for a two-element set $X = \{x, y\}$ with $xy \neq yx$, the maximal set commuting with X is X^+, see [1].

One can also use the above setting to define a computation. Given languages X and Y, for a word $x \in X$ define the rewriting rule

$$x \Rightarrow_C x' \text{ if there exists } x' \in X, y, y' \in Y \text{ such that } xy = y'x'.$$

Let $\Rightarrow_C^*$ be the transitive and reflexive closure of $\Rightarrow_C$. What can we say about this relation? Very little seems to be known. Naturally unanswered (according to our knowledge) questions are: when is the closure of a word $x \in X$, i.e., its *orbit*, finite or is it always recursive for given regular languages X and Y?

We do not claim that the above operation is biologically motived. However, it seems to resemble some of the natural operations on DNA-sequences, see [13]: the result is obtained by matching two words and then factorizing the result differently. Consequently, it provides a further illustration of the computational complexity of the operations based on matching of words.

Our goal is to consider a particular question on commutation of languages without any biological or other motivation. More precisely, we want to introduce an algebraic approach, so-called *fixed point approach*, to study *Conway's Problem.* The problem asks whether or not the maximal language commuting with a given regular language X is regular as well. The maximal such set is called the *centralizer of* X. An affirmative answer is known only in very special cases, see, e.g., [15], [3], [14], [7] and [8]. In general, the problem seems to be very poorly understood – it is not even known whether the centralizer of a finite language X is recursive!

We show that the centralizer of any language is the largest fixed point of a very natural language operator. Consequently, it is obtained as the limit of simple recursion. When started from a regular set X all the intermediate approximations are regular, as well. However, as we show by an example, an infinite number of iterations might be needed and hence Conway's Problem remains unanswered.

One consequence of our results is that if Conway's Problem has an affirmative answer, even nonconstructively, then the membership problem for the centralizer of a regular language is actually decidable, i.e., the centralizer is recursive.

2 Preliminaries

We shall need only very basic notations of words and languages; for words see [12] or [2] and for languages [17] or [5].

Mainly to fix the terminology we specify the following. The *free semigroup* generated by a finite *alphabet* A is denoted by A^+. Elements of A^+ are called *words* and subsets of A^+ are called *languages*. These are denoted by lower case letters $x, y, \ldots$ and capital letters $X, Y, \ldots$, respectively. Besides standard operations on words and languages we especially need the operations of the quotients. We say that a word v is a *left quotient* of a word w if there exists a word u such that $w = uv$, and we write $v = u^{-1}w$. Consequently, the operation $(u, v) \to u^{-1}v$ is a partial mapping. Similarly, we define *right quotients*, and extend both of these to languages in a standard way: $X^{-1}Y = \{x^{-1}y \mid x \in X, y \in Y\}$.

We say that two languages X and Y *commute* if they satisfy the equality $XY = YX$. Given an $X \subseteq A^+$ it is straightforward to see that there exists the unique maximal set $\mathcal{C}(X)$ commuting with X. Indeed, $\mathcal{C}(X)$ is the union of all sets commuting with X. It is also easy to see that $\mathcal{C}(X)$ is a subsemigroup of A^+. Moreover, we have simple approximations, see [3]:

Lemma 1. *For any $X \subseteq A^+$ we have*

$$X^+ \subseteq \mathcal{C}(X) \subseteq \mathrm{Pref}(X^+) \cap \mathrm{Suf}(X^+).$$

Here $\mathrm{Pref}(X^+)$ (resp. $\mathrm{Suf}(X^+)$) stands for all nonempty prefixes (resp. suffixes) of X^+.

Now we can state:

Conway's Problem. Is the centralizer of a regular X regular as well?

Although the answer is believed to be affirmative, it is known only in the very special cases, namely when X is a prefix set, binary or ternary, see [15], [3] or [7], respectively. This fact together with the fact that we do not know whether the centralizer of a finite set is even recursive, can be viewed as an evidence of amazingly intriguing nature of the problem of commutation of languages.

Example 1. *(from [3]) Consider* $X = \{a, ab, ba, bb\}$*. Then, as can be readily seen, the centralizer* $\mathcal{C}(X)$ *equals to* $X^+ \setminus \{b\} = (X \cup \{bab, bbb\})^+$*. Hence, the centralizer is finitely generated but doesn't equal either to* X^+ *or* $\{a, b\}^+$*.*

Finally, we note that in the above the centralizers were defined with respect to the semigroup A^+. Similar theory can be developed over the free monoid A^*.

3 Fixed Point Approach

As discussed extensively in [14] and [8], there has been a number of different approaches to solve the Conway's Problem. Here we introduce one more, namely so-called fixed point approach. It is mathematically quite elegant, although at the moment it does not yield into breakthrough results. However, it can be seen as another evidence of the challenging nature of the problem.

Let $X \subseteq A^+$ be an arbitrary language. We define recursively

$$X_0 = \mathrm{Pref}(X^+) \cap Suf(X^+)$$

and

$$X_{i+1} = X_i \setminus [X^{-1}(XX_i \Delta X_i X) \cup (XX_i \Delta X_i X)X^{-1}], \text{ for } i \geq 0, \tag{1}$$

where Δ denotes the symmetric difference of languages. Finally we set

$$Z_0 = \bigcap_{i \geq 0} X_i. \tag{2}$$

We shall prove

Theorem 1. *Z_0 is the centralizer of X, i.e., $Z_0 = \mathcal{C}(X)$.*

Proof. The result follows directly from the following three facts:

(i) $X_{i+1} \subseteq X_i$ for all $i \geq 0$,

(ii) $\mathcal{C}(X) \subseteq X_i$ for all $i \geq 0$, and

(iii) $Z_0 X = X Z_0$.

Indeed, (iii) implies that $Z_0 \subseteq \mathcal{C}(X)$, while (ii) together with (2) implies that $\mathcal{C}(X) \subseteq Z_0$.

Claims (i)–(iii) are proved as follows. Claim (i) is obvious. Claim (ii) is proved by induction on i. The case $i = 0$ is clear, by Lemma 1. Let $z \in \mathcal{C}(X)$ and assume that $\mathcal{C}(X) \subseteq X_i$. Assume that $z \notin X_{i+1}$. Then

$$z \in X^{-1}(XX_i \Delta X_i X) \cup (XX_i \Delta X_i X)X^{-1}.$$

Consequently, there exists an $x \in X$ such that

$$xz \text{ or } zx \in (XX_i \Delta X_i X).$$

This, however, is impossible since $z \in \mathcal{C}(X) \subseteq X_i$ and $\mathcal{C}(X)X = X\mathcal{C}(X)$. For example, xz is clearly in XX_i, but also in X_iX due to the identity $xz = z'x'$ with $z' \in \mathcal{C}(X)$, $x' \in X$. So z must be in X_{i+1}, and hence (ii) is proved.

It remains to prove the condition (iii). If Z_0X and XZ_0 were unequal, then there would exist a word $w \in Z_0$, such that either $wX \not\subseteq XZ_0$ or $Xw \not\subseteq Z_0X$. By symmetry we may assume the previous case. By the definition of Z_0 and (i) this would mean that beginning from some index k we would have $wX \not\subseteq XX_i$, when $i \geq k$. However, $w \in Z_0 \subseteq X_i$ for every $i \geq 0$, especially for k, and hence $X_kX \neq XX_k$. This would mean that $w \in (XX_k \Delta X_k X)X^{-1}$ and hence $w \notin X_{k+1}$, and consequently $w \notin Z_0$, a contradiction. □

Theorem 1 deserves a few remarks.

First, we define the language operator φ by the formula

$$\varphi : Y \mapsto Y \setminus [X^{-1}(XY \Delta YX) \cup (XY \Delta YX)X^{-1}],$$

where X is a fixed language. Then obviously all languages commuting with X are fixed points of φ, and the centralizer is the maximal one. Second, in the construction of Theorem 1 it is not important to start from the chosen X_0. Any superset of $\mathcal{C}(X)$ would work, in particular A^+. Third, as can be seen by analyzing the proof, in formula (1) we could drop one of the members of the union. However, the presented symmetric variant looks more natural.

In the next section we give an example showing that for some languages an infinite number of iterations are needed in order to get the centralizer. In the final concluding section we draw some consequences of this result.

4 An Example

As an example of the case in which the fixed point approach leads to an infinite iteration we discuss the language $X = \{a, bb, aba, bab, bbb\}$. First, we prove that the centralizer of this language is X^+. To do this we start by proving the following two lemmata. We consider the *prefix order* of A^*, and say that two words are *incomparable* if they are so with respect to this order. Given language X, we say that v is an incomparable element of X if v is incomparable with every other word in X.

Lemma 2. *Let X be a rational language including a word v incomparable in X. If $w \in \mathcal{C}(X)$, then for some integer $n \in \{0, 1, 2, \ldots\}$ there exist words $t \in X^n$ and $u \in \mathrm{Suf}(X)$ such that $w = ut$ and $uX^nX^* \subseteq \mathcal{C}(X)$.*

Proof. If $w \in \mathcal{C}(X)$ and v is an incomparable element in X, then equation $X\mathcal{C}(X) = \mathcal{C}(X)X$ implies that $vw \in \mathcal{C}(X)X$ and therefore $vwv_1^{-1} \in \mathcal{C}(X)$ for some element $v_1 \in X$. Repeating the argument n times we obtain

$$v^n w(v_n \cdots v_2 v_1)^{-1} \in \mathcal{C}(X), \qquad v_i \in X,$$

where $t = v_n \cdots v_2 v_1$ and $w = ut$. Then $v^n u \in \mathcal{C}(X)$ for some integer $n \in \{0, 1, 2, \ldots\}$ and word $u \in \mathrm{Suf}(X) \cap \mathrm{Pref}(w)$. Since v is incomparable, we conclude that for every $s \in X^n$

$$v^n us \in \mathcal{C}(X)X^n = X^n\mathcal{C}(X),$$

and hence

$$us \in \mathcal{C}(X).$$

In other words, $uX^n \subseteq \mathcal{C}(X)$. Since $\mathcal{C}(X)$ is a semigroup, we have also the inclusion $uX^nX^* \subseteq \mathcal{C}(X)$. □

For every proper suffix $u_i \in \mathrm{Suf}(X)$, including the empty word 1, there either exists a minimal integer n_i, for which $u_iX^{n_i} \subseteq \mathcal{C}(X)$, or $u_iX^n \nsubseteq \mathcal{C}(X)$ for every integer $n \geq 0$. Since Lemma 2 excludes the latter case, we can associate with every word $w \in \mathcal{C}(X)$ a word $u_i \in \mathrm{Suf}(X)$ and the minimal n_i such that $w \in u_iX^{n_i}X^*$.

Lemma 3. *If the finite language X contains an incomparable word, it has a rational centralizer. Moreover, the centralizer is finitely generated.*

Proof. If the language X is finite, then the set of proper suffixes of X is also finite. With the above terminology we can write

$$\begin{aligned} \mathcal{C}(X) &= \bigcup_{i \in I} u_iX^{n_i}X^* \\ &= \underbrace{(\bigcup_{i \in I} u_iX^{n_i})}_{=G} X^* \\ &= GX^*, \end{aligned}$$

where I is an index set defining suffixes u_i above. Here the language G is finite and $X \subseteq G$. Indeed, if $u_0 = 1$, then $n_0 = 1$, and hence $u_0X^{n_0} = 1 \cdot X = X \subseteq G$.

Since $\mathcal{C}(X)$ is a semigroup and X is included in G, we obtain

$$\mathcal{C}(X) = \mathcal{C}(X)^+ = (GX^*)^+ = (X + G)^+ = G^+.$$

□

Now we can prove that the centralizer of our language $X = \{a, bb, aba, bab, bbb\}$ is X^+. The word bab is incomparable. The set of proper suffixes of X is $\{1, a, b, ab, ba, bb\}$. We will consider all of these words separately:

$u_0 = 1 :\ 1 \cdot X \subseteq \mathcal{C}(X)$ so that $n_0 = 1$.

$u_1 = a :\ a \in X \subseteq \mathcal{C}(X)$ so that $n_1 = 0$.

$u_2 = b :\ b \cdot a^n \cdot a \notin X\mathcal{C}(X) = \mathcal{C}(X)X$ and therefore $b \cdot a^n \notin \mathcal{C}(X)$ for all $n \in \mathbb{N}$.
This means that the number n_2 does not exist.

$u_3 = ab :\ a \cdot ab \cdot (bab)^n \notin \mathrm{Suf}(X^+)$ implies $aab(bab)^n \notin \mathcal{C}(X)$ so that $aab(bab)^n \notin X\mathcal{C}(X)$ and therefore $ab(bab)^n \notin \mathcal{C}(X)$ for all $n \in \mathbb{N}$.
Hence the number n_3 does not exist.

$u_4 = ba :\ ba \cdot a^n \cdot a \notin X\mathcal{C}(X)$ and therefore $ba \cdot a^n \notin \mathcal{C}(X)$ for all $n \in \mathbb{N}$, and hence the number n_4 does not exist.

$u_5 = bb :\ bb \in X \subseteq \mathcal{C}(X)$ so that $n_5 = 0$.

As a conclusion $I = \{0, 1, 5\}$, and $G = \bigcup_{i \in I} u_i X^{n_i} = 1 \cdot X + a + bb = X$. This gives us the centralizer

$$\mathcal{C}(X) = GX^* = XX^* = X^+,$$

in other words we have established:

Fact 1. $\mathcal{C}(\{a, bb, aba, bab, bbb\}) = \{a, bb, aba, bab, bbb\}^+$.

Next we prove that the fixed point approach applied to the language X leads to an infinite loop of iterations. We prove this by showing that there exist words in $\mathcal{C}(X) \setminus X_i$ for every X_i of the iteration (1). To do this we take a closer look on the language $L = (bab)^* ab (bab)^*$. Clearly, $L \subseteq X_0 = \mathrm{Pref}(X^+) \cap \mathrm{Suf}(X^+)$ and $L \cap X^+ = \emptyset$.

By the definition of the fixed point approach, word $w \in X_i$ is in X_{i+1} if and only if $Xw \subseteq X_i X$ and $wX \subseteq XX_i$. We will check this condition for an arbitrary word $(bab)^k ab (bab)^n \in L$ with $k, n \geq 1$. The first condition $Xw \subseteq X_i X$ leads to the cases:

$$\left.\begin{array}{rcl} a \cdot (bab)^k ab (bab)^n & = & (aba)(bb \cdot a)^{k-1} (bab)^{n+1} \in X^+X \subseteq X_i X, \\ bb \cdot (bab)^k ab (bab)^n & = & (bbb) a (bb \cdot a)^{k-1} (bab)^{n+1} \in X^+X \subseteq X_i X, \\ aba \cdot (bab)^k ab (bab)^n & = & a(bab) a (bb \cdot a)^{k-1} (bab)^{n+1} \in X^+X \subseteq X_i X, \\ bbb \cdot (bab)^k ab (bab)^n & = & (bb)^2 a (bb \cdot a)^{k-1} (bab)^{n+1} \in X^+X \subseteq X_i X \end{array}\right\} \quad (3)$$

and

$$bab \cdot (bab)^k ab (bab)^n = (bab)^{k+1} ab (bab)^{n-1} \cdot bab \in X_i X.$$

However, the last one holds if and only if

$$(bab)^{k+1} ab (bab)^{n-1} \in X_i. \quad (4)$$

Similarly, the second condition $wX \subseteq XX_i$ yields us:

$$\left.\begin{array}{rcl} (bab)^k ab(bab)^n \cdot a & = & bab(bab)^{k-1}(a\cdot bb)^n aba \in XX^+ \subseteq XX_i, \\ (bab)^k ab(bab)^n \cdot bb & = & bab(bab)^{k-1}(a\cdot bb)^n a \cdot bbb \in XX^+ \subseteq XX_i, \\ (bab)^k ab(bab)^n \cdot aba & = & bab(bab)^{k-1}(a\cdot bb)^n a\cdot a\cdot bab\cdot a \in XX^+ \subseteq XX_i, \\ (bab)^k ab(bab)^n \cdot bbb & = & bab(bab)^{k-1}(a\cdot bb)^n a\cdot a\cdot bb\cdot bb \in XX^+ \subseteq XX_i \end{array}\right\} \tag{5}$$

and

$$(bab)^k ab(bab)^n \cdot bab = bab \cdot (bab)^{k-1} ab(bab)^{n+1} \in XX_i.$$

Here the last one holds if and only if

$$(bab)^{k-1} ab(bab)^{n+1} \in X_i. \tag{6}$$

From (4) and (6) we obtain the equivalence

$$(bab)^k ab(bab)^n \in X_{i+1} \iff (bab)^{k+1} ab(bab)^{n-1}, (bab)^{k-1} ab(bab)^{n+1} \in X_i \tag{7}$$

Now, the result follows by induction, when we cover the cases $k = 0$ or $n = 0$.

In the case $k = 0$ and $n \geq 0$ we have that $ab(bab)^n \in X_0$, but $ab(bab)^n \notin X_1$, since $a \cdot ab(bab)^n \notin X_0 X$. The same applies also for $(bab)^n ba$ by symmetry.

In the case $n = 0$ and $k \geq 1$ we first note that $X(bab)^k ab \subseteq X_0 X$, due to the equations in (3) and the fact that $bab \cdot (bab)^k ab = (bab)^k ba \cdot bab \in X_0 X$. Similarly, $(bab)^k abX \subseteq XX_0$, due to the equations in (5) and the fact $(bab)^k ab \cdot bab = bab \cdot (bab)^{k-1} ab(bab) \in XX_0$. These together imply that $(bab)^k ab \in X_1$. On the other hand, since $(bab)^k ba \notin X_1$, then $bab \cdot (bab)^k ab \notin X_1 X$, and hence $(bab)^k ab \notin X_2$.

Over all we obtain the following result: if $i = \min\{k, n+1\}$, then

$$(bab)^k ab(bab)^n \in X_i \quad \text{but} \quad (bab)^k ab(bab)^n \notin X_{i+1}.$$

The above can be illustrated as follows. If the language $(bab)^* ab(bab)^*$ is written in the form of a downwards infinite pyramid as shown in Figure 1, then Figure 2 shows how the fixed point approach deletes parts of this language during the iterations. In the first step $X_0 \to X_1$ only the words in $ab(bab)^*$ are deleted as drawn on the leftmost figure. The step $X_1 \to X_2$ deletes words in $(bab)ab(bab)^*$ and $(bab)^* ab$, and so on. On the step $X_i \to X_{i+1}$ the operation always deletes the remaining words in$(bab)^i ab(bab)^*$ and $(bab)^* ab(bab)^{i-1}$, but it never manages to delete the whole language $(bab)^* ab(bab)^*$. This leads to an infinite chain of steps as shown in the following

Fact 2. $X_0 \supset X_1 \supset \cdots X_i \supset \cdots \mathcal{C}(X)$.

$$ab$$

$$ab(bab) \qquad (bab)ab$$

$$ab(bab)^2 \qquad (bab)ab(bab) \qquad (bab)^2ab$$

$$ab(bab)^3 \qquad (bab)ab(bab)^2 \qquad (bab)^2ab(bab) \qquad (bab)^3ab$$

$$ab(bab)^4 \qquad (bab)ab(bab)^3 \qquad (bab)^2ab(bab)^2 \qquad (bab)^3ab(bab) \qquad (bab)^4ab$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

Figure 1: Language $(bab)^*ab(bab)^*$ written as a pyramid.

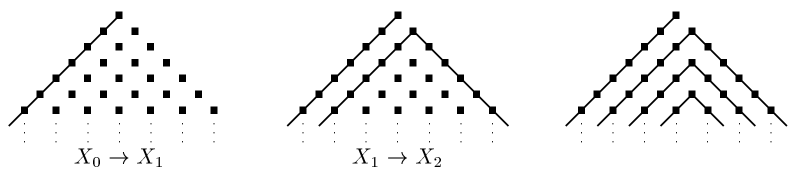

Figure 2: Deleting words in $(bab)^*ab(bab)^*$ from languages X_i during the iteration.

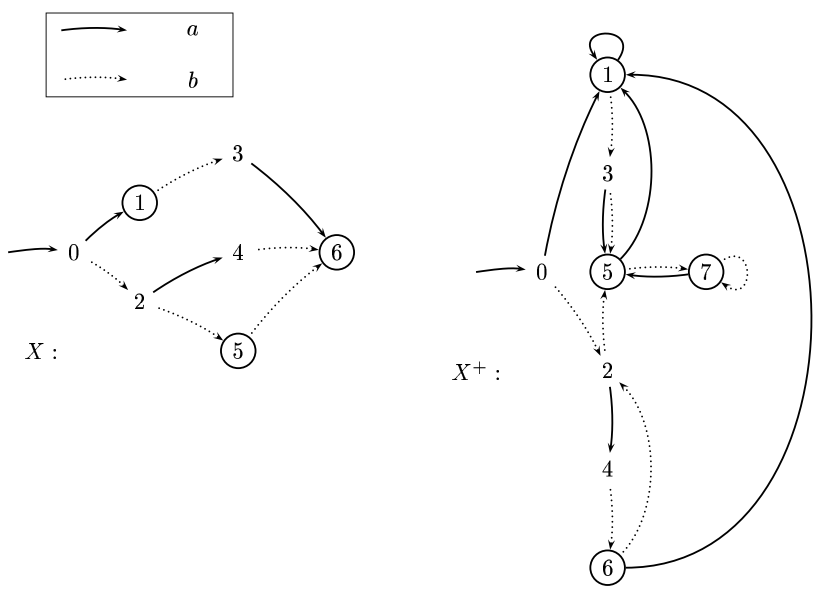

Figure 3: Finite automata recognizing languages X and X^+

When computing the approximations X_i the computer and available software packages are essential. We used *Grail+*, see [18]. For languages X and X^+ their minimal automata are shown in Figure 3.

Let us consider the minimal automata we obtain in the iteration steps of the procedure, and try to find some common patterns in those. The automaton recognizing the starting language X_0 is given in Figure 4.

The numbers of states, finals states and transitions for a few first steps of the iteration are given in Table 1. From this table we can see that after a few steps the growth becomes constant. Every step adds six more states, three of those being final, and eleven transitions.

When we draw the automata corresponding to subsequent steps from X_i to X_{i+1}, as in Figures 5 and 6, we see a clear pattern in the growth. See that automata representing languages X_5 and X_6 are built from two sequence of sets of three states. In the automaton of X_6 both sequences have got an additional set of three states. So there are totally six new states, including three new final ones, and eleven new transitions. In every iteration step the automata seem to use the same pattern to grow. When the number of iteration steps goes to infinity, the lengths of both sequences go also to infinity. Then the corresponding states can be merged together and the result will be the automaton recognizing the language X^+. This seems to be a general phenomena in the cases where an infinite number of iterations are needed. Intuitively that would solve the Conway's Problem. However, we do not know how to prove it.

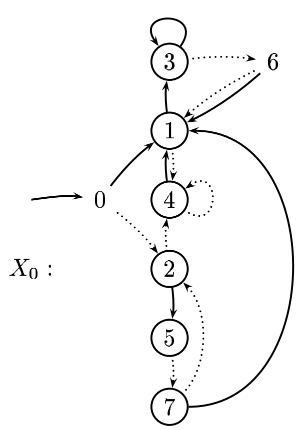

Figure 4: Finite automaton recognizing the language X_0

	final states	states	transitions
X_0	6	8	15
X_1	5	9	17
X_2	6	13	24
X_3	9	19	35
X_4	12	25	46
X_5	15	31	57
X_6	18	37	68
X_7	21	43	79
X_8	24	49	90
X_9	27	55	101
X_{10}	30	61	112
X_{11}	33	67	123
X_{12}	36	73	134
X_{13}	39	79	145
X_{14}	42	85	156
X_{15}	45	91	167
X_{16}	48	97	178
X_{17}	51	103	189
X_{18}	54	109	200
X_{19}	57	115	211
	final states	states	transitions
X_{20}	60	121	222
X_{21}	63	127	233
X_{22}	66	133	244
X_{23}	69	139	255
X_{24}	72	145	266
X_{25}	75	151	277
X_{26}	78	157	288
X_{27}	81	163	299
X_{28}	84	169	310
X_{29}	87	175	321
X_{30}	90	181	332
X_{31}	93	187	343
X_{32}	96	193	354
X_{33}	99	199	365
X_{34}	102	205	376
X_{35}	105	211	387
X_{36}	108	217	398
X_{37}	111	223	409
X_{38}	114	229	420
X_{39}	117	235	431

Table 1: The numbers of states, final states and transitions of automata corresponding to the iteration steps for the language X.

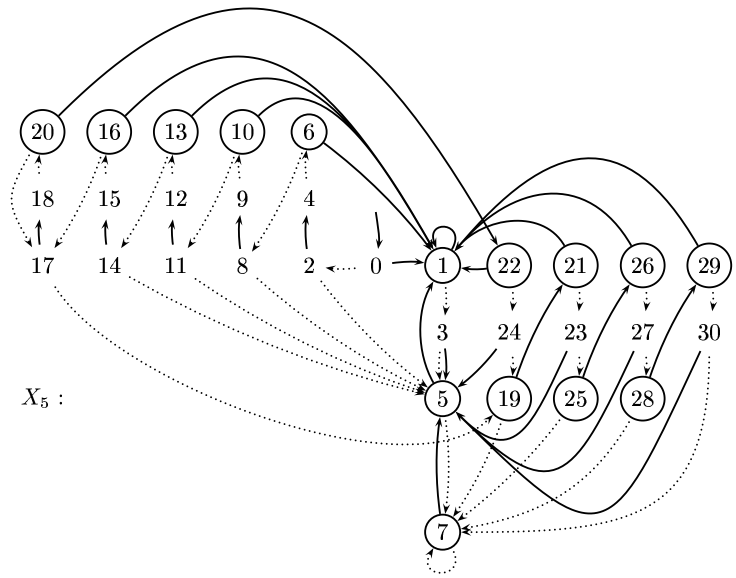

Figure 5: The finite automaton recognizing X_5

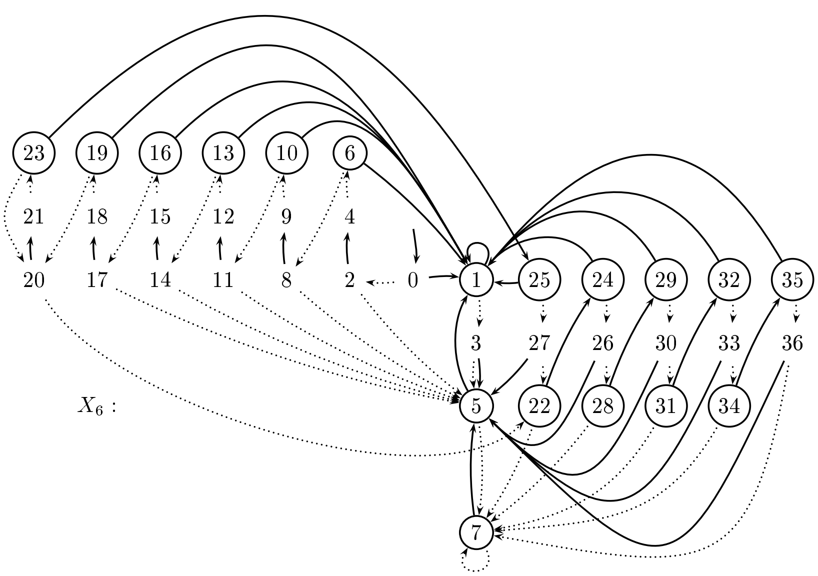

Figure 6: The finite automaton recognizing X_6

5 Conclusions

Our main theorem has a few consequences. As theoretical ones we state the following two. These are based on the fact that formula (1) in the recursion is very simple. Indeed, if X is regular so are all the approximations X_i. Similarly, if X is recursive so are all these approximations. Of course, this does not imply that also the limit, that is the centralizer, should be regular or recursive.

What we can conclude is the following, much weaker result, first noticed in [8]:

Theorem 2. *If X is recursive, then $\mathcal{C}(X)$ is in co-RE, that is its complement is recursively enumerable.*

Proof. As we noticed above, all approximations X_i are recursive, and moreover effectively findable. Now the result follows from the identity

$$\overline{\mathcal{C}(X)} = \bigcup_{i \geq 0} \overline{X_i},$$

where bar is used for the complementation. Indeed, a method to algorithmically list all the elements of $\overline{\mathcal{C}(X)}$ is as follows: Enumerate all words $w_1, w_2, w_3, \ldots$ and test for all i and j whether $w_i \in \overline{X_j}$. Whenever a positive answer is obtained output w_i. □

For regular languages X we have the following result.

Theorem 3. *Let $X \in A^+$ be regular. If $\mathcal{C}(X)$ is regular, even noneffectively, then $\mathcal{C}(X)$ is recursive.*

Proof. We assume that X is regular, and effectively given, while $\mathcal{C}(X)$ is regular however not necessarily constructively. We have to show how to decide the membership problem for $\mathcal{C}(X)$. This is obtained as a combination of two semialgorithms.

A semialgorithm for the question "$X \in \overline{\mathcal{C}(X)}$?" is obtained, as in the proof of Theorem 2. A semialgorithm for the complementary question is as follows: Given x, enumerate all regular languages $L_1, L_2, \ldots$ and test whether or not

$$(i) \quad L_i X = X L_i$$

and

$$(ii) \qquad x \in L_i.$$

Whenever the answers to both of the questions are affirmative output the input x. The correctness of the procedure follows since $\mathcal{C}(X)$ is assumed to be regular. Note also that the tests in (i) and (ii) can be done since L_i's and X are effectively known regular languages. □

Theorem 3 is a bit amusing since it gives a meaningful example of the case where the regularity implies the recursiveness. Note also that a weaker assumption that $\mathcal{C}(X)$ is only context-free would not allow the proof, since the question in (i) is undecidble for contex-free languages, see [4].

We conclude with more practical comments. Although we have not been able to use our fixed point approach to answer Conway's Problem even in some new special cases, we can use it for concrete examples. Indeed, as shown by experiments, in most cases the iteration terminates in a finite number of steps. Typically in these cases the centralizer is of one of the following forms:

(i) A^+,

(ii) X^+, or

(iii) $\{w \in A^+ \mid wX, Xw \subseteq XX^+\}$.

However, in some cases, as shown in Section 4, an infinite number of iterations are needed – and consequently some ad hoc methods are required to compute the centralizer.

The four element set of example 1 is an instance where the centralizer is not of any of the forms (i)–(iii). This as well as some other such examples, can be verified by the fixed point approach using a computer, see [16].

Concluding remark: The fixed point approach was independently noticed by M. Hirvensalo and Z. Esik.

References

[1] Berstel, J., Karhumäki, J.: Combinatorics on words – A tutorial, *Bull. EATCS* 79 (2003), 178–229.

[2] Choffrut, C., Karhumäki, J.: Combinatorics of Words. In Rozenberg, G., Salomaa, A. (eds.), *Handbook of Formal Languages*, Vol. 1, Springer-Verlag (1997) 329–438.

[3] Choffrut, C., Karhumäki, J., Ollinger, N.: The commutation of finite sets: a challenging problem, *Theoret. Comput. Sci.*, 273 (1–2) (2002) 69–79.

[4] Harju, T., Ibarra, O., Karhumäki, J., Salomaa, A., Decision questions concerning semilinearity, morphisms and commutation of languages, *J. Comput. System Sci.* 65 (2002) 278–294.

[5] Hopcroft, J., Ullman, J., *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (1979).

[6] Karhumäki, J.: Challenges of commutation: an advertisement, in *Proc of FCT 2001*, LNCS 2138, Springer (2001) 15–23

[7] Karhumäki, J., Latteux, M., Petre, I.: The commutation with codes and ternary sets of words, in *Proc. of STACS 2003*, LNCS 2607, Springer (2003), 74–84; final version to appear.

[8] Karhumäki, J. and Petre, I.: Conway's problem for three-word sets, *Theoret. Comput. Sci.*, 289/1 (2002) 705–725.

[9] Karhumäki, J. and Petre, I., Two problems on commutation of languages, in: G. Păun, G. Rozenberg and A. Salomaa (eds.), *Current Trends in Theoretical Computer Science*, World Scientific, to appear.

[10] Karhumäki, J., Plandowski, W., Rytter, W.: On the complexity of decidable cases of the commutation problem of languages, LNCS 2138, Springer (2001) 193–203; final version to appear.

[11] Lothaire, M.: *Combinatorics on Words* (Addison-Wesley, Reading, MA.), (1983).

[12] Lothaire, M.: *Algebraic Combinatorics on Words* (Cambridge University Press), (2002).

[13] Păun, G., Rozenberg, G. and Salomaa, A., *DNA Computing. New Computing Paradigms*, Texts in Theoretical Computer Science. An EATCS series. Springer (1998)

[14] Petre, I.: *Commutation Problems on Sets of Words and Formal Power Series*, PhD Thesis, University of Turku (2002)

[15] Ratoandromanana, B.: Codes et motifs, *RAIRO Inform. Theor.*, 23(4) (1989) 425–444.

[16] Salmela, P., *Rationaalisen kielen sentralisaattorista ja sen määrittämisestä kiintopistemetodilla*, Master's theses, University of Turku (2002)

[17] Salomaa, A., *Formal Languages*, Academic Press (1973).

[18] Grail+ 3.0 – software package, Department of Computer Science, University of Western Ontario, Canada, `http://www.csd.uwo.ca/research/grail/ grail.html`.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.fi

University of Turku
- **Department of Information Technology**
- **Department of Mathematics**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**