Scientific Annals of Computer Science

RNA-editing with combined insertion and deletion preserves regularity

Erik P. de Vink,[1] Hans Zantema,[2] & Dragan Bošnački[3]

Abstract

We consider two elementary forms of string rewriting called *guided insertion/deletion* and *guided rewriting*. The original strings are modified depending on the match with a given set of auxiliary strings, called guides. Guided insertion/deletion considers matching of a string and a guide with respect to a specific correspondence of strings. Guided rewriting considers matching of a string and a guide with respect to an equivalence relation on the alphabet. Guided insertion/deletion is inspired by RNA-editing, a biological process by which the original genetic information stored in DNA is modified before its final expression. The formalism here allows for simultaneous insertion and deletion of string elements. Guided rewriting, based on a letter-to-letter relation, is technically more appealing than guided insertion/deletion. We prove that guided rewriting preserves regularity: for every regular language its closure under guided rewriting is regular too. In the proof we will rely on the auxiliary notion of a slice sequence. We establish a correspondence of slice sequences and guide rewrite sequences. Because of their left-to-right nature, slice sequences are more convenient to deal with than guided rewrite sequences in the construction of the finite automata that we encounter in the proofs of regularity. Based on the result for guided rewriting we establish that guided insertion/deletion preserves regularity as well.

Keywords: RNA editing, string rewriting, guided insertion/deletion, guided rewriting, regular languages

[1]Department of Mathematics and Computer Science, Technische Universiteit Eindhoven and Centrum Wiskunde en Informatica, Amsterdam

[2]Department of Mathematics and Computer Science, Technische Universiteit Eindhoven and Institute for Computing and Information Sciences, Radboud University Nijmegen

[3]Department of Mathematics and Computer Science and Department of Biomedical Engineering, Technische Universiteit Eindhoven

1 Introduction

RNA editing is a biological mechanism that modifies the original "text" of the genetic information of a living organism after it is copied (transcribed) from the DNA. In this paper, we investigate two elementary formalisms of string transformation which are inspired by RNA editing. We consider *guided insertion/deletion*, which is close to an editing mechanism as encountered in the living cell, and guided rewriting, based on an *adjustment relation*, which lends itself more easily to formal analysis. In both forms of string rewriting a substring of the original string is adapted when it matches a string from a specific set, called the set of *guides*. The set G of guides is fixed and finite. In guided insertion/deletion the guide and the part of the string that is rewritten are not required to be of the same length, but they need to be equal up to occurrences of a distinguished dummy symbol. In guided rewriting the guide and substring are equivalent symbol-by-symbol according to the adjustment relation, a chosen and fixed equivalence relation.

Both flavors of rewriting preserve the finiteness of the initial set of strings. Assuming a finite set of guides G, in both cases only a finite set of strings can be obtained by repeatedly rewriting a given string. In this work we show that also regularity of the initial string set is preserved for both cases. Starting from a language L, we consider the extension $L_{i/d}$ of the language with all the rewrites obtained by guided insertion/deletion and the extension L_G of the language obtained by adding all the adjustment-based guided rewrites. The main results of the paper state that regularity of L implies regularity of $L_{i/d}$ and regularity of L_G.

The motivation of this work is rooted in one of the basic processes of life which concerns the flow of genetic information. Initially, the original information stored in DNA molecules is faithfully copied to RNA by the process of transcription. In eukaryote cells, i.e., cells that have a nucleus, the RNA which is finally translated to proteins, does not carry an exact copy of the original information stored in the DNA part. Instead, the RNA string, which transmits the genetic information further on the chain, is a modification obtained by post-processing. On an abstract level an RNA molecule can be regarded as string over the alphabet $\{C, G, A, U\}$. The modification consists of insertion and deletion of these letters, also called nucleotides, on multiple locations in the original string. The class of the underlying adaptation mechanisms is collectively called *RNA-editing*.

The computational power of insertion-deletion systems for RNA-editing is studied in [20]: After abstracting away the biological details, an insertion step is the replacement of a string uv by the string $u\alpha v$ taken from a particular finite set of triples u, α, v. Similarly, a deletion step replaces $u\alpha v$ by uv for another finite set of triples u, α, v. In [14] the restriction is considered where u and v are both empty. This mechanism claims full computational power, that is, all recursively enumerable languages can be generated in this way.

Inspired by DNA recombination, Head proposes in [9] the notion of *splicing*. The DNA molecules (strings) are modified by so-called splicing rules. Each splicing rule is a tuple $r = (u_1, v_1; u_2, v_2)$. Given two words $w_1 = x_1u_1v_1y_1$ and $w_2 = x_2u_2v_2y_2$ the rule r produces the word $w = x_1u_1v_2y_2$. So, the word w_1 is split in between u_1 and v_1, the word w_2 in between u_2 and v_2 and the resulting subwords x_1u_1 and v_2y_2 are recombined into the word w. For splicing a closure result, reminiscent to the one for guided insertion/deletion and guided rewriting considered in this paper, has been established. Casted in our terminology, if L is a regular language and S is a finite set of splicing rules, then L_S is regular too, cf. [12, 15]. Here, L_S is the least language containing L and closed under the splicing rules of S.

Compared to the above described formal systems, natural RNA-editing mechanisms are very often quite limited. In most of the natural RNA-editing instances only the symbol U is inserted and deleted, instead of arbitrary strings α, see e.g. [1]. Motivated by this observation, we investigate guided insertion/deletion focusing on the special role of a distinguished symbol 0, a formal analog of the RNA letter U. A similar scheme, but which prohibited simultaneous insertion and deletion of the special symbol, we considered in [21]. To prove that under the present scheme regularity is preserved we need the stepping stone of guided rewriting based on an abstract adjustment relation. In particular, we prove the regularity preservation theorem for guided insertion/deletion by using the analogous result for guided rewriting based on adjustment.

The regularity result for the adjustment-based rewriting is proved by constructing a finite automaton that accepts the language L_G. The construction procedure takes as input the set of guides G and a given finite automaton accepting the language L. A crucial point in the proof is the translation of the guided rewrite sequences into so-called slice sequences. The point is that, since guides may overlap, each guided rewrite step adds a 'layer' on top of the previous string. In this sense guided rewriting is vertically oriented. E.g., Figure 2 in Section 5 shows six rewrite steps of the string *ebcfa* yielding the string *fbcfb* involving eight layers in total. However, in reasoning about recognition by a finite automaton a horizontal orientation is more natural. One would like to sweep from left to right, so to speak. Again referring to Figure 2, five slices can be distinguished, viz. a slice for each symbol of the string *ebcfa*. The technical machinery developed in this paper allows for a transition between the two orientations.

In order to obtain a regularity result for guided insertion/deletion we apply a string transformation: for the language L and finite set of guides G over the alphabet $\Sigma \cup \{0\}$ let N be a bound on the number of consecutive 0's in G. We adapt the alphabet $\Sigma \cup \{0\}$ to $\Sigma \cup \Theta$ by introducing $N+1$ new symbols representing strings of 0's up to length N and a new symbol representing all strings of 0's larger than N. The transformation we consider replaces in a string u over the alphabet $\Sigma \cup \{0\}$ all its maximal substrings of 0's by the corresponding symbol of Θ, obtaining a string $\overline{u}$ over the alphabet $\Sigma \cup \Theta$. In this way we obtain the transformed language $\overline{L}$ and guide set $\overline{G}$ over $\Sigma \cup \Theta$. We establish that the closure of a language L over $\Sigma \cup \{0\}$ under guided insertion/deletion with respect to the set of guides G is regular iff $\overline{L}$ under guided rewriting with respect to $\overline{G}$ is regular.

Paper layout. Section 2 provides the biological background of RNA-editing. The theorem on the preservation of regularity for guided insertion-deletion is presented in Section 3. The notion of guided rewriting based on an adjustment relation is introduced in Section 4; a corresponding theorem on the preservation of regularity for guided rewriting is formulated here too. To pave the way for the proof of the latter theorem, Section 5 introduces the notions of a rewrite sequence and of a slice sequence and establishes their relationship. Rewrite sequences record the subsequent guided rewrites that take place, slice sequences represent the cumulative effect of all rewrites at a particular position of the string being adjusted. Section 6 describes a construction of a finite automaton accepting the extended language L_G for a fixed set of guides G and a finite automaton accepting the language L. In Section 7 the proof is given that regularity of L implies the regularity of $L_{i/d}$. Section 8 wraps up with related work and concluding remarks.

2 Biological Motivation

In this section we briefly describe the biological aspects of the RNA-editing mechanisms and provide the corresponding abstractions.

In the living cell there are different kinds of RNA editing that vary in the type of edited RNA and the set of editing operations. In this paper we focus on an editing which is quite extensively studied from a biological point of view and which involves simultaneous insertion and deletion of uracil in messenger RNA (mRNA) [3]. (Some other types of RNA editing involve also letter substitution, cf. e.g. [17].) Uracil is represented by the letter U. The three other types of nucleotides for RNA, viz. adenine, guanine and cytosine, are represented by the letters A, G and C, respectively.

The type of U-insertion/deletion editing we are dealing with occurs in the mitochondrial genes of kinetoplastid protozoa [19]. Kinetoplastids are single cell organisms that include parasites like *Trypanosoma brucei* and *Crithidia fasciculata* and that can cause serious diseases in humans and/or animals. Although the mitochondrial genes contain a relatively small amount of information, they are of utmost importance for the organism as a whole [5]. Apart from being interesting from a fundamental point of view, understanding of the RNA-editing mechanisms can be crucial in developing medicines for the corresponding diseases.

Modifications of kinetoplastid mRNA are usually made within the coding regions. These are the parts that are translated into proteins, which are the building blocks of the cells. The coded information of the original gene can be altered and therefore expressed, i.e. translated into proteins, in a varying number of ways, depending on the environment in the cell. This provides additional flexibility as well as potential specialization of different parts of the organisms for particular functions.

In the sequel we describe an idealized version of the mechanism for the insertion and deletion of U. More details can be found, for instance, in [19, 1, 6, 18]. For simplicity we assume that only identical letters match with one another. In reality, the matching is based on complementarity, usually assuming the so-called Crick-Watson pairs: A matches with U and G matches with C.

A single step in the mRNA editing involves two strands of RNA, a strand of (pre-edited) mRNA and a strand of guide RNA (gRNA), the latter typically referred to as the *guide*. We explain the mechanism for the insertion of uracil on the example given in Figure 1. We consider the mRNA fragment $u = N_1N_2N_3N_4N_5$ and the guide $g = N_2N_3UUUN_4$, where N_i can be an arbitrary nucleotide A, G or C, but not U. Obviously, there is some match between u and g involving the letters N_2, N_3, and N_4, which is partially 'spoiled' by the UUU sequence. Guide g attaches to u at positions where the letters match. The matching substrings N_2N_3 and N_4 serve as anchors (Fig. 1a).

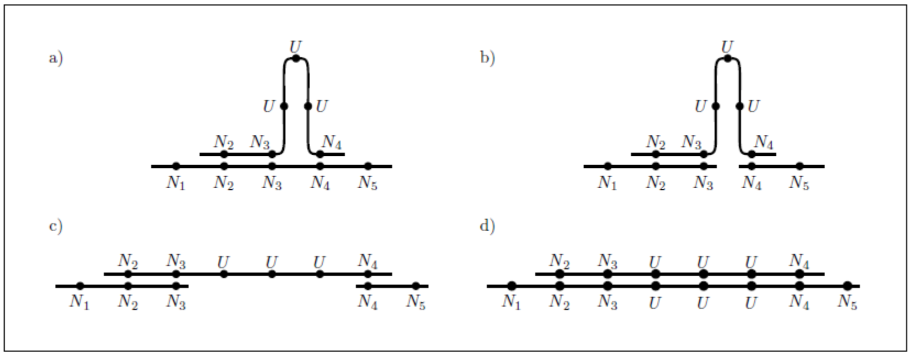


Figure 1: Various stages of guided U-insertion

By means of enzyme machinery, i.e., a special complex of proteins-enzymes called editosome [2], u is split open between N_3 and N_4 (Fig. 1b and 1c). Then the editosome fills the gap between the anchors using the guide as a template. (Actually, different enzymes of the editosome complex are responsible for cutting the mRNA strand at the first mismatch position and adding the Us, however here we can safely disregard these details.) For each letter U in the guide the editosome adds a U in the gap. As a result the mRNA string u is transformed into $N_1N_2N_3UUUN_4N_5$ (Fig. 1d). In general, one can have more than two anchors (involving only non-U letters) in which the guide and the mRNA strand match. In that case the mRNA is opened between each pair of anchors and all gaps between these anchors are filled with U such that the number of Us in the guide is matched.

The deletion of Us from a strand of mRNA is implemented by a symmetrical biochemical mechanism. We illustrate the deletion process too on an example. Assume the mRNA strand $u = N_1N_2N_3UUN_4N_5$ and the guide $g = N_2N_3N_4$. Like in the insertion case, g initiates the editing by attaching itself to u at the matching positions N_2,N_3, and N_4. Only now the enzymatic complex removes the mismatching UU substring between N_3 and N_4 to ensure a perfect match between the substring and the guide. As a result the edited string $N_1N_2N_3N_4N_5$ is obtained. In general, we can have several anchoring positions on the same string. In that case, all Us between each two matching positions are removed from the mRNA.

Simultaneous insertions and deletions of U are also possible. For instance the guide $N_2N_3UUUN_4$ can induce parallel editing of the string $N_1UN_2UN_3UN_4UN_5UN_6$ which results in the string

$N_1UN_2N_3UUUN_4UN_5UN_6$, where the U between N_2 and N_3 has been deleted and two U's between N_3 and N_4 have been inserted. This is done by the same biochemical mechanisms that are involved in separate insertions and deletions. Like in the other cases described above, we can have multiple insertions and deletions induced by the same guide on the original pre-edited sequence.

Abstracting from the biochemical details, for all three cases considered above it is common that a strand $u = xyz$, such that y equals g up to occurrences of U, is modified by the insertion and deletion mechanism and becomes a string $v = xgz$. The rewriting system that we describe in the sequel also applies to another case with the same effect. For example, consider a guide $g = N_2N_3UUUN_4$ and a pre-edited mRNA $u = N_1N_2N_3UUN_4N_5N_6$. Now, to obtain the match of the guide g and a substring y of u, a U is inserted in u, resulting in the string $v = N_1N_2N_3UUUN_4N_5N_6$. If the U subsequence in y was longer though, like in the case for $u' = N_1N_2N_3UUUN_4N_5N_6$ and $g' = N_2N_3UUN_4$, then we have that the extra U in u' is removed resulting in $v' = N_1N_2N_3UUN_4N_5N_6$.

For our purposes, the mRNA editing mechanism underlying U-insertion and deletion boils down to symbolic manipulations of strings. The common denominator of the above described editing mechanisms is that in a single step some substring y is replaced by a guide g for which y and g match modulo occurrences of the symbol U. In the rest of the paper the analog of the nucleotide U will be denoted by 0.

3 Guided insertion / deletion

Inspired by the biological scheme of editing of mRNA as discussed in the previous section, we study more abstract notions of guided insertion and deletion and guided rewriting based on an adjustment relation in the remainder of this paper. In this section we address guided insertion and deletion, turning to guided rewriting in Section 4.

More precisely, fix an alphabet Σ and distinguish $0 \notin \Sigma$. Put $\Sigma_0 = \Sigma \cup \{0\}$. Choose a finite set $G \subseteq \Sigma_0^*$, with elements g also referred to as guides. Reflecting the biological mechanism, we assume that each $g \in G$ is not equal to the empty string ε and that the first and last letter of each $g \in G$ is not equal to 0. Hence, $G \subseteq \Sigma \cup \Sigma{\cdot}\Sigma_0^*{\cdot}\Sigma$, or, more particularly, $G \subseteq \Sigma{\cdot}(\,0^*{\cdot}\Sigma\,)^*$. Now a guided insertion/deletion step $\Rightarrow_{i/d}$ with respect to G is given by

$$u \Rightarrow_{i/d} v \iff u = xyz \wedge v = xgz \wedge g \in G \wedge \pi(y) = \pi(g)$$

where $y \in \Sigma{\cdot}\Sigma_0^*{\cdot}\Sigma$, and $\pi(y)$ and $\pi(g)$ are obtained from y and g, respectively, by removing their 0s. Thus, $\pi : \Sigma_0^* \to \Sigma^*$ is the homomorphism such that $\pi(\varepsilon) = \varepsilon$, $\pi(0) = \varepsilon$ and $\pi(a) = a$ for $a \in \Sigma$. So, intuitively, g is anchored on the substring y of u and sequences of 0s are adjusted as prescribed by the guide g, in effect replacing the substring y by the guide g while maintaining the prefix x and suffix z.

As a simple example of a single guided insertion/deletion step, for $G = \{g\}$ with $g = bcb000ab0c$ and $u = a00bc00babcc00a00b$, we have $u \Rightarrow_{i/d} v$ for $v = a00bcb000ab0cc00a00b$. Here we have $u = a00 \cdot bc00babc \cdot c00a00b$, $\pi(bc00babc) = bcbabc = \pi(bcb000ab0c)$ and $v = a00 \cdot bcb000ab0c \cdot c00a00b$. Note, for the string v, being the result of a rewrite with guide g itself with only one possible anchoring, only trivial steps can be taken further. So, the operation of guided insertion/deletion with the same guide g at the same position in a string is idempotent. However, anchoring may overlap. Consider the set of guides $G = \{\, aa0a,\, a0aa \,\}$, for example. Then the string aaa yields an infinite rewrite sequence

$$aaa \Rightarrow_{i/d} aa0a \Rightarrow_{i/d} a0aa \Rightarrow_{i/d} aa0a \Rightarrow_{i/d} a0aa \;\cdots$$

Still, from aaa only finitely many different rewrites can be obtained by insertion/deletion steps guided by this G, viz. $\{\, aaa,\, aa0a,\, a0aa \,\}$.

The restrictions put on G exclude arbitrary deletions (possible if ε would be allowed as guide) and infinite pumping (if guides need not be delimited by symbols from Σ). As an illustration of the latter case, starting from the string abc and 'guide' $0ab$, the infinite sequence $abc \Rightarrow_{i/d} 0abc \Rightarrow_{i/d} 00abc \Rightarrow_{i/d} 000abc\ldots$ would be obtained. The restriction on the substring y prevents to make changes outside the scope of the guide g and forbids $a0b000c \Rightarrow_{i/d} ab0c$ by way of the guide ab.

As a first observation we show that the set $L^u_{i/d} = \{\, v \in \Sigma_0^* \mid u \Rightarrow^*_{i/d} v \,\}$, for any finite set of guides G and any string u, is finite. Write $u = a_0 0^{i_1} a_1 \ldots a_{n-1} 0^{i_n} a_n$ where $a_k \in \Sigma$, $k = 0, \ldots, n$, and $i_k \geqslant 0$, $k = 1, \ldots, n$, for some $n \geqslant 0$. In effect, a guided insertion/deletion step only modifies the substrings 0^{i_k} or leaves them as is. Therefore, after one or more guided insertion/deletion steps the substrings 0^{i_k} are strings taken from the set

$$Z^u_{i/d} = \{\, 0^{i_k} \mid 1 \leqslant k \leqslant n \,\} \cup \{\, 0^\ell \mid xa \cdot 0^\ell bz \in G,\ x, z \in \Sigma_0^*,\, a, b \in \Sigma,\ \ell \geqslant 0 \,\}$$

Thus, if $u \Rightarrow_{i/d}^{*} v$ then $v \in \hat{L}_{i/d}^{u} = \{\, a_0 z_1 a_1 \ldots a_{n-1} z_n a_n \mid z_k \in Z_{i/d}^{u},\ 1 \leqslant k \leqslant n \,\}$, i.e. $L_{i/d}^{u} \subseteq \hat{L}_{i/d}^{u}$. Since the set of guides G is finite, it follows that $Z_{i/d}^{u}$ is finite, that $\hat{L}_{i/d}^{u}$ is finite and that $L_{i/d}^{u}$ is finite as well.

More generally, given a set of guides G, we define the extension by insertion/deletion $L_{i/d}$ of a language L over Σ_0 by putting $L_{i/d} = \{\, v \in \Sigma_0^* \mid \exists u \in L \colon u \Rightarrow_{i/d}^{*} v \,\}$. Casted to the biological setting of Section 2, L are the strands of messenger RNA, G are strands of guide RNA. Next, we consider the question whether regularity of the language L is inherited by the induced language $L_{i/d}$. Note, despite the finiteness of the insertion/deletion scheme for a single string, it is not obvious that such a statement would hold.

With the machinery of rewrite sequences and slice sequences developed in the sequel of the paper, we will be able to prove the following for guided insertion/deletion.

Theorem 1. *If L is a regular language, then the language $L_{i/d}$ is regular too.*

We will prove Theorem 1 by applying a more general result on guided rewriting, viz. Theorem 2 formulated in the next section and ultimately proven in Section 6. As in the notion of guided rewriting as developed in the sequel, symbols are only replaced by single symbols by which lengths of strings are always preserved, a transformation is required to be able to apply Theorem 2.

Before moving to guided rewriting we relate our results to those of [21]. There a relation similar to $\Rightarrow_{i/d}$ was introduced, with the only difference that in a single step either 0's are deleted or inserted, but not both at the some time. The consequence of this small difference is significant: the main conclusion of [21] is that in that setting regularity is *not* preserved, which is the opposite of Theorem 1 in the present setting.

4 Guided rewriting

The idea of guided rewriting is that a symbol is replaced by an equivalent symbol, equivalence taken with respect to some *adjustment relation* $\sim$. The resulting one-one correspondence of the symbols of the string u and its guided rewrite v, enjoyed by this notion of reduction, will turn out technically convenient in the sequel. Intuitively, the equivalent symbols abstract from sequences of 0's.

Let Σ be a finite alphabet and $\sim$ an equivalence relation on Σ, called the *adjustment relation*. If $a \sim b$ we say that a can be adjusted to b. For a string $u \in \Sigma^*$ we write $\#u$ for its length, use $u[i]$ to denote its i-th element, $i = 1, \ldots, \#u$, and let $u[p,q]$ stand for the substring $u[p]\, u[p{+}1] \cdots u[q]$. The relation $\sim$ is lifted to Σ^* by putting

$$u \sim v \quad \textit{iff} \quad \#u = \#v \;\wedge\; \forall i = 1, \ldots, \#u \colon u[i] \sim v[i]$$

Next we define the notion of guided rewriting that involves an adjustment relation.

Definition 1. *We fix a finite subset $G \subseteq \Sigma^*$, called the set of guides.*

(a) For $u, v \in \Sigma^*$, $g \in G$, $p \geqslant 0$, we define $u \Rightarrow_{g,p} v$, stating that v is the rewrite of u with guide g at position p, by

$$u \Rightarrow_{g,p} v \quad \textit{iff} \quad \exists x, y, z \in \Sigma^* \colon u = x\,y\,z \;\wedge\; \#x = p \;\wedge\; y {\sim} g \;\wedge\; v = x\,g\,z$$

(b) We write $u \Rightarrow v$ if $u \Rightarrow_{g,p} v$ for some $g \in G$ and $p \geqslant 0$. We use $\Rightarrow^*$ to denote the reflexive transitive closure of $\Rightarrow$. A sequence $u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_n$ is called a reduction.

(c) For a language L over Σ and a set of guides G we write

$$L_G = \{\, v \in \Sigma^* \mid \exists u \in L \colon u \Rightarrow^* v \,\}$$

So, a $\Rightarrow$-step adjusts a substring to a guide in G element-wise, and L_G consists of all strings that can be obtained from a string from L by any number of such adjustments. Clearly, if $u \Rightarrow v$ then also $u \sim v$.

As an example, if $\Sigma = \{a, b, c\}$, $G = \{bb\}$ and $a \sim b$ but not $a \sim c$, then by a $\Rightarrow$-step two consecutive symbols not equal to c are replaced by two consecutive b's. In particular, $aaacaa \rightarrow_{bb,1} abbcaa$ and $abbcaa \rightarrow_{bb,0} bbbcaa$. We have

$$\begin{aligned} \{aaacaa\}_G \;\; = \;\; \{\, & aaacaa,\ bbacaa,\ abbcaa, \\ & aaacbb,\ bbbcaa,\ abbcbb,\ bbacbb, bbbcbb \,\} \end{aligned}$$

Next, we state the main result of this paper regarding guided rewriting.

Theorem 2. *Let an equivalence relation $\sim$ on Σ and a finite set of guides G be given. Suppose L is a regular language. Then L_G is regular too.*

Before going to the proof, we first show that both finiteness of G and the requirement of $\sim$ being an equivalence relation are essential.

To see that finiteness of G is essential for Theorem 2 to hold, let $G = \{ c\,a^k c\,b^k c \mid k \geqslant 1 \}$ and $L = \mathcal{L}(c\,a^* c\,a^* c)$. Let $\sim$ satisfy $a \sim b$ but not $a \sim c$. Then all elements of L on which an adjustment is applicable are of the shape $ca^k ca^k c$, where the result of the adjustment is $ca^k cb^k c$, which can not be changed by any further adjustment because of the presence of b. So

$$L_G \,\cap\, \mathcal{L}(c\,a^* c\,b^* c) \;=\; \{ c\,a^k c\,b^k c \mid k \geqslant 1 \}$$

is not regular. Since regularity is closed under intersection we conclude that L_G cannot be regular itself. However, note that in this example the set of guides is not finite, but not regular either. (We revisit this issue in Section 8.)

Also equivalence properties of $\sim$ are essential for Theorem 2. For $G = \{ ab \}$ and $\sim\, = \{ (a,b), (b,a) \}$ the only possible $\Rightarrow$-steps are replacing the pattern ba by ab. Note that here $\sim$ is neither reflexive nor transitive. Since ba may be replaced by ab, bubble sort on a's and b's can be mimicked by $\Rightarrow^*$, while on the other hand $\Rightarrow^*$ preserves both the number of a's and the number of b's. Hence

$$\mathcal{L}((ab)^*)_G \,\cap\, \mathcal{L}(a^* b^*) \;=\; \{ a^k b^k \mid k \geqslant 0 \}$$

which proves that $\mathcal{L}((ab)^*)_G$ is not regular, again since regularity is closed under intersection.

5 Rewrite sequences and slice sequences

This section introduces an auxiliary notion, viz. the notion of a slice sequence, that can be considered as a 'vertical' version of the 'horizontal' notion of a rewrite sequence. We will establish a correspondence between these notions, which provides the basis of our proof of Theorem 2 in Section 6.

Fix an alphabet Σ, an adjustment relation $\sim$, and a set of guides G.

Definition 2. *A sequence $\varrho = (g_k, p_k)_{k=1}^{r}$ of guide-position pairs is called a guided rewrite sequence for a string $u \in \Sigma^*$ if it holds that (i) $g_k \in G$, (ii) $0 \leqslant p_k \leqslant \#u - \#g_k$, and (iii) $u[p_k{+}1, p_k{+}\#g_k] \sim g_k$, for all $k = 1, \ldots, r$.*

A guide-position pair (g, p) indicates an intended guided rewrite with g of the string u at position p. For the rewrite to fit we must have $p + \#g \leqslant \#u$. The first p symbols of u, i.e. the substring $u[1, p]$, are not affected by the rewrite, as are the last $\#u - p + \#g$ symbols of u, i.e. the substring $u[p{+}\#g{+}1, \#u]$.

The sequence ϱ induces a sequence of strings $(u_k)_{k=0}^r$ by putting $u_0 = u$ and u_k such that $u_{k-1} \Rightarrow_{g_k,p_k} u_k$ for $k = 1, \ldots, r$. To conclude that $u_{k-1} \Rightarrow_{g_k,p_k} u_k$ is indeed a proper guided rewrite step, in particular that we have $u_{k-1}[p_k{+}1, p_k{+}\#g_k] \sim g_k$, we use the assumption $u[p_k{+}1, p_k{+}\#g_k] \sim g_k$ and the fact that if $u \Rightarrow_{g,p} v$ then $u[p + 1, p + \#g] \sim v[p + 1, p + \#g]$ and $u \sim v$. Therefore, by induction $u \Rightarrow^* u_{k-1}$ and $u[p_k{+}1, p_k{+}\#g_k] \sim u_{k-1}[p_k{+}1, p_k{+}\#g_k]$

The final string u_r of the guided rewrite sequence is referred to as the yield of ϱ for u, notation $\mathit{yield}(\varrho)$. Conversely, every specific reduction from u to v gives rise to a corresponding guided rewrite sequence for u.

A guided rewrite sequence $\varrho = (g_k, p_k)_{k=1}^r$ is said to be repetition-free all its guide-position pairs are different, i.e. for $1 \leqslant k_1, k_2 \leqslant r$, $g_{k_1} = g_{k_2} \wedge p_{k_1} = p_{k_2}$ implies $k_1 = k_2$.

Definition 3. *Let $a \in \Sigma$. A sequence $s\ell = (g_i, q_i)_{i \in I}$ of guide-offset pairs, for $I \subseteq \mathbb{N}$ a finite index set, is called a slice for a and G if it holds that (i) $g_i \in G$, (ii) $1 \leqslant q_i \leqslant \#g_i$, and (iii) $a \sim g_i[q_i]$, for all $i \in I$. The slice $s\ell$ is called a slice for a string $u \in \Sigma^*$ at position n, $1 \leqslant n \leqslant \#u$, if it is a slice of $u[n]$.*

A position p refers to the symbol $u[p]$ of a string u. In contrast, in a guide-offset pair (g, q) of a slice sequence, the offset q is relative to the guide g. Since we require $1 \leqslant q \leqslant \#g$ for such a pair, the symbol $g[q]$ is well-defined. We will reserve the use of q for offsets, indices within a guide, and the use of p for positions after which a rewrite may take place, i.e. for lengths of proper substrings of a given string.

The goal of the notion of slice is to summarize the effect of a number of guided rewrites local to a specific position within a string. The symbol generated by the last rewrite that affected the position, i.e. the particular symbol of the last element of the slice sequence, is part of the overall outcome of the total rewrite. This symbol is called the *yield* of the slice. More precisely, if $I \neq \emptyset$, the yield of a slice $s\ell$ for a symbol a is defined as $\mathit{yield}(s\ell) = g_{i_{\max}}[q_{i_{\max}}]$ where $i_{\max} = \max(I)$. In case $I = \emptyset$, we put $\mathit{yield}(s\ell) = a$. Occasionally we write $a \sim s\ell$, as for a slice $s\ell$ for a symbol a it always holds that $a \sim \mathit{yield}(s\ell)$.

A slice $s\ell = (g_i, q_i)_{i \in I}$ is said to be repetition-free if, for $i_1, i_2 \in I$, $g_{i_1} = g_{i_2} \wedge q_{i_1} = q_{i_2}$ implies $i_1 = i_2$. If we have $I = \emptyset$, the slice $s\ell$ is called the empty slice.

Next we consider sequences of slices, and investigate the relationship between slices on two consecutive positions in a guided rewrite sequence.

Definition 4. *A sequence $\sigma = (s\ell_n)_{n=1}^{\#u}$ is called a slice sequence for a string u if the following holds:*

- *$s\ell_n$ is a slice for u at position n, for $n = 1, \dots, \#u$;*
- *for $n = 1, \dots, \#u{-}1$, putting $s\ell_n = (g_i, q_i)_{i \in I}$ and $s\ell_{n+1} = (g'_i, q'_i)_{i \in J}$, there exists a monotone partial injection $\gamma_n : I \to J$ such that, for all $i \in I$ and $j \in J$,*

$$\begin{array}{lrcl} (i) & i \notin dom(\gamma_n) & \Longrightarrow & q_i = \#g_i \\ (ii) & \gamma_n(i) = j & \Longleftrightarrow & g_i = g'_j \wedge q_i + 1 = q'_j \\ (iii) & j \notin rng(\gamma_n) & \Longrightarrow & q'_j = 1 \end{array}$$

- *the slices $s\ell_1$ and $s\ell_{\#u}$, say $s\ell_1 = (g_i, q_i)_{i \in I}$ and $s\ell_{\#u} = (g'_j, q'_j)_{j \in J}$, satisfy $q_i = 1$, for all $i \in I$, and $q'_j = \#g'_j$, for all $j \in J$, respectively.*

For the slices $s\ell_n$ and $s\ell_{n+1}$ the mapping $\gamma_n : I \to J$ is called the cut for $s\ell_n$ and $s\ell_{n+1}$. It witnesses that $s\ell_n$ and $s\ell_{n+1}$ match in the sense that a rewrite *(i)* may end at position n, *(ii)* may continue for its next offset at position $n{+}1$, and *(iii)* may start at position $n{+}1$. Note, for arbitrary pairs of slices the cut may not exist. In fact, the requirements of Definition 4 completely determine the cut between two slices. Since a cut γ is an order-preserving bijection from $dom(\gamma)$ to $rng(\gamma)$, and $dom(\gamma)$ and $rng(\gamma)$ are finite, it follows that for two slices $s\ell, s\ell'$ the cut for $s\ell$ and $s\ell'$ is unique. We write $s\ell \rightsquigarrow s\ell'$. A slice $s\ell = (g_i, q_i)_{i \in I}$ is called a start slice if $q_i = 1$ for all $i \in I$. Similarly, $s\ell$ is called an end slice if $q_i = \#g_i$ for all $i \in I$. A start slice is generally, but not necessarily, associated with the first position of the string that is rewritten, an end slice with the last position. Note, a start slice as well as an end slice are allowed to be empty. The yield of the slice sequence σ is the sequence of the yield of its slices, i.e. we define $yield(\sigma) = yield(s\ell_1) \cdots yield(s\ell_{\#u})$.

Example 1. *Let $\sim$ be the adjustment relation with equivalence classes $\{a, b\}, \{c, d\}, \{e, f\}$ and let the set of guides G be given by $G = \{\, g_1,\, g_2,\, g_3 \,\}$*

	I_n	$(g_i, q_i)_{i \in I_n}$
$s\ell_1$	$2, 4$	$2 \mapsto (g_1, 1),\ 4 \mapsto (g_1, 1)$
$s\ell_2$	$2, 3, 4$	$2 \mapsto (g_1, 2),\ 3 \mapsto (g_2, 1),\ 4 \mapsto (g_1, 2)$
$s\ell_3$	$1, 3$	$1 \mapsto (g_3, 1),\ 3 \mapsto (g_2, 2)$
$s\ell_4$	$3, 5, 6$	$3 \mapsto (g_2, 3),\ 5 \mapsto (g_1, 1),\ 6 \mapsto (g_1, 1)$
$s\ell_5$	$5, 6$	$5 \mapsto (g_1, 2),\ 6 \mapsto (g_1, 2)$

Table 1: slice sequence of Example 1

where $g_1 = fb$, $g_2 = ace$ and $g_3 = d$. For the string $u = ebcfa$ we consider the guided rewrite sequence $\varrho = (\,(g_3, 2),\ (g_1, 0),\ (g_2, 1),\ (g_1, 0),\ (g_1, 3),\ (g_1, 3)\,)$. The associated reduction looks like

$$\begin{aligned} ebcfa \ \Rightarrow_{g_3,2}\ & ebdfa \ \Rightarrow_{g_1,0}\ fbdfa \ \Rightarrow_{g_2,1} \\ & facea \ \Rightarrow_{g_1,0}\ fbcea \ \Rightarrow_{g_1,3}\ fbcfb \ \Rightarrow_{g_1,3}\ fbcfb \end{aligned} \tag{1}$$

Recording what happens at all of the five positions of the string u yields, for this example, the slice sequence $\sigma = (s\ell_n)_{n=1}^{5}$ given in Table 1. The slice sequence is visualized in Figure 2.

For the particular choice of $I_1, \ldots, I_5$, the monotone partial injection γ_n, $n = 1 \ldots 4$, maps every number to itself. It is easily checked that all requirements of a slice sequence hold. The ovals covering guide-offset pairs reflect the cuts as mappings between to adjacent slices. However, they also comprise complete guides across a varying number of slices. Note, $s\ell_1$ is a start slice, $s\ell_5$ is an end slice. We have for the slice sequence $\sigma = (s\ell_n)_{i=1}^{5}$ that $yield(\sigma) = yield(s\ell_1) \cdot \cdots \cdot yield(s\ell_5) = fbcfb$. Indeed, this coincides with the yield of the guided rewrite sequence ϱ of (1).

The rest of this section is devoted to proving that the above holds in general: Given a string and a set of guides, for every guided rewrite sequence there exists a slice sequence and for every slice sequence there exists a guided rewrite sequence. Moreover, the yield of the guided rewrite sequence and slice sequence are the same.

Theorem 3. *Let $\varrho = (g_k, p_k)_{k=1}^{r}$ be a guided rewrite sequence for a string u. Then there exists a slice sequence $\sigma = (s\ell_n)_{n=1}^{\#u}$ for u such that $yield(\sigma) = yield(\varrho)$.*

Proof. Induction on r. If ϱ is the empty rewrite sequence, we take for σ the slice sequence of n empty slices. Then we have $yield(\varrho) = u$ and $yield(\sigma) = u$.

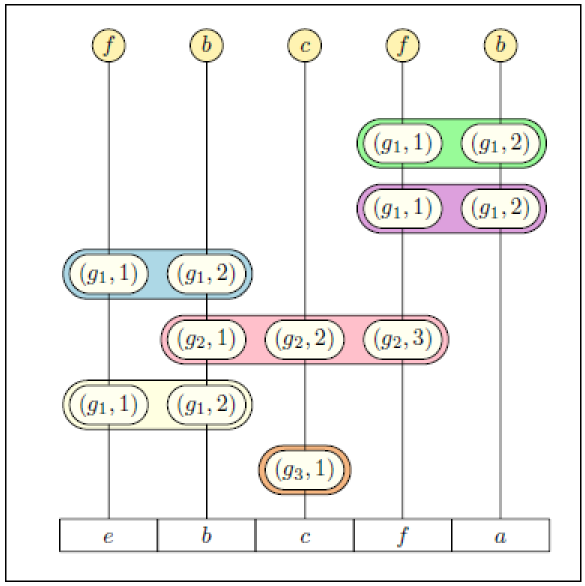


Figure 2: slice sequence of Example 1

Suppose ϱ is non-empty. Let $(u_k)_{k=0}^r$ be the sequence of strings induced by ϱ. By induction hypothesis there exists a slice sequence σ' for the first $r-1$ steps of ϱ. Suppose $u_{r-1} \Rightarrow_{g_r,p_r} u_r$. The slice sequence σ is obtained by extending the slices of σ' from position p_r+1 to $p_r+\#g_r$ with the pairs $(g_r, n-p_r)$. Then,

$$\begin{aligned} \mathit{yield}(\sigma) &= \mathit{yield}(\sigma'[1,p_r]) \cdot g_r[1,\#g_r] \cdot \mathit{yield}(\sigma'[p_r+\#g_r+1,\#u]) \\ &= u_{r-1}[1,p_r] \cdot g_r \cdot u_{r-1}[p_r+\#g_r+1,\#u_{r-1}] \;=\; u_r \;=\; \mathit{yield}(\varrho) \end{aligned}$$

Verification of σ being a slice sequence for u requires transitivity of $\sim$. $\square$

In order to show the reverse of Theorem 3 we proceed in a number of stages. First we need to relate individual guide-offset pairs in neighboring slices. For this purpose we introduce the ordering $\preccurlyeq$ on so-called chunks.

Definition 5. *Let $\sigma = (s\ell_n)_{n=1}^{\#u}$ be a slice sequence for u. Assume we have $s\ell_n = (\,g_{n,i}, q_{n,i}\,)_{i \in I_n}$, for $n = 1, \ldots, \#u$. Let $\gamma_n \colon I_n \to I_{n+1}$ be the cut for $s\ell_n$ and $s\ell_{n+1}$, $1 \leqslant n < \#u$. Define $\mathcal{X} = \{\,(\,g_{n,i}, q_{n,i}, i, n\,) \mid 1 \leqslant n \leqslant \#u,\ i \in I_n\,\}$ to be the set of chunks of σ and define the ordering $\preccurlyeq$ on $\mathcal{X}$ by putting $(\,g, q, i, n\,) \preccurlyeq (\,g', q', i', n'\,)$ iff*

- *either* $n' \geqslant n$ *and there exist indexes* $\ell_0, h_0, \ldots, \ell_{n'-n}, h_{n'-n}$ *such that*
  - $\ell_k, h_k \in I_{n+k}$ *and* $\ell_k \leqslant h_k$, $0 \leqslant k \leqslant n'-n$
  - $h_k \in dom(\gamma_{n+k})$ *and* $\gamma_{n+k}(h_k) = \ell_{k+1}$, $0 \leqslant k < n'-n$
  - $\ell_0 = i$ *and* $h_{n'-n} = i'$
- *or* $n' \leqslant n$ *and there exist indexes* $\ell_0, h_0, \ldots, \ell_{n-n'}, h_{n-n'}$ *such that*
  - $\ell_k, h_k \in I_{n'+k}$ *and* $\ell_k \leqslant h_k$, $0 \leqslant k \leqslant n-n'$
  - $\ell_k \in dom(\gamma_{n'+k})$ *and* $\gamma_{n'+k}(\ell_k) = h_{k+1}$, $0 \leqslant k < n-n'$
  - $h_0 = i'$ *and* $\ell_{n-n'} = i$

Note, for indices $\ell_k, h_k \in I_{n+k}$ as above, we have $\ell_k \leqslant h_k$, so ℓ_k is the lower index, h_k is the higher index. In the above setting with $n' \geqslant n$, we say that the sequence $\ell_0, h_0, \ell_1, h_1, \ldots, \ell_{n'-n}, h_{n'-n}$ is leading from $i \in I_n$ up to $i' \in I_{n'}$. Likewise for the case where $n' \leqslant n$.

For example, for the slice sequence $(s\ell_i)_{i=1}^r$ of Figure 2, to identify the guide belonging to the guide-offset pair $(g_2, 1)$ of slice $s\ell_2$, the pair is more precisely represented by the chunk $(g_2, 1, 3, 2)$, for the pair is associated with index $3 \in I_2$ of slice $s\ell_2$. Since for the cuts $\gamma_2 : I_2 \to I_3$ and $\gamma_3 : I_3 \to I_4$ we have $\gamma_2(3)$ and $\gamma_3(3) = 3$, we have $(g_2, 1, 3, 2) \preccurlyeq (g_2, 2, 3, 3) \preccurlyeq (g_2, 3, 3, 4)$ via the sequence $3, 3, 3, 3$ connects $(g_2, 1)$ and $(g_2, 2)$, and $3, 3, 3, 3$ connecting $(g_2, 2)$ and $(g_2, 3)$. (Hence the combination of these sequences is $3, 3, 3, 3, 3, 3$ which connects $(g_2, 1)$ and $(g_2, 3)$ directly.) As no jumps from a low index ℓ to a high index h need to be taken, we also have $(g_2, 1, 3, 2) \succcurlyeq (g_2, 2, 3, 3) \succcurlyeq (g_2, 3, 3, 4)$. Thus $(g_2, 1, 3, 2) \equiv (g_2, 2, 3, 3) \equiv (g_2, 3, 3, 4)\,\}$. In fact, $\{\, (g_2, 1, 3, 2), (g_2, 2, 3, 3), (g_2, 3, 3, 4)\,\}$ is an equivalence class for $\mathcal{X}$ corresponding to the guide g_2 (cf. Lemma 1). Differently, we have $(g_2, 1, 3, 2) \preccurlyeq (g_1, 2, 6, 5)$ relating g_2 to the fourth occurrence of g_1 via the sequence $3, 3, 3, 3, 3, 5, 5, 5$, for example. Since there is a jump here from $\ell_2 = 3$ to $h_2 = 5$, we do not have $(g_2, 1, 3, 2) \succcurlyeq (g_1, 2, 6, 5)$. The ordering $(g_2, 1, 3, 2) \preccurlyeq (g_1, 2, 6, 5)$ reflects that apparently the rewrite with this occurrence of g_1 is on top of part of the rewrite using g_2 as guide.

Given a slice sequence σ, the ordering $\preccurlyeq$ on the chunks of σ in $\mathcal{X}$ gives rise to a partial ordering on the set $\mathcal{X}/{\equiv}$ of equivalence classes of chunks. As we will argue, the equivalence classes correspond to guides and their ordering corresponds to the relative order in which the guides occur in a rewrite sequence ϱ having the same yield as the slice sequence σ.

Lemma 1. *(a) The relation $\preccurlyeq$ on $\mathcal{X}$ is reflexive and transitive.*

(b) The relation $\equiv$ on $\mathcal{X}$ such that $x \equiv y \iff x \preccurlyeq y \wedge y \preccurlyeq x$ is an equivalence relation.

(c) The ordering $\preccurlyeq$ on $\mathcal{X}/{\equiv}$ induced by $\preccurlyeq$ on X by $[x] \preccurlyeq [y] \iff \exists x' \in [x]\, \exists y' \in [y] \colon x' \preccurlyeq y'$, makes $\mathcal{X}/{\equiv}$ a partial order.

Proof. We only prove part (a); parts (b) and (c) are straightforward. As to verify reflexivity of $\preccurlyeq$, let $(g, q, i, n) \in \mathcal{X}$. Choose $\ell_0 = i$ and $h_0 = i$. Then $\ell_0, h_0 \in I_n$, $\ell_0 \leqslant h_0$, and, obviously, $\ell_0 = i$ and $h_0 = i$. So, $(g, q, i, n) \preccurlyeq (g, q, i, n)$.

As to verify transitivity of $\preccurlyeq$, assume $(g_1, q_1, i_1, n_1) \preccurlyeq (g_2, q_2, i_2, n_2)$ and $(g_2, q_2, i_2, n_2) \preccurlyeq (g_3, q_3, i_3, n_3)$. We check that $(g_1, q_1, i_1, n_1) \preccurlyeq (g_3, q_3, i_3, n_3)$ for the case $n_3 \leqslant n_1 \leqslant n_2$, leaving the other cases, which are similar or easier, to the reader. Pick $\ell_k, h_k \in I_{n_1+k}$, for $0 \leqslant k \leqslant n_2{-}n_1$, meeting the first set of requirements of Definition 5, and pick $\ell'_j, h'_j \in I_{n_3+j}$, for $0 \leqslant j \leqslant n_2{-}n_3$ meeting the second set of requirements. Consider the sequence of indices ℓ'_0, h'_0, …, ℓ_0, $h'_{n_1-n_3}$ which is the initial part of the sequence from $i_2 \in I_{n_2}$ up to $i_3 \in I_{n_3}$, viz. the first $n_1{-}n_3$ out of $n_2{-}n_3$ pairs of indices, except that $\ell'_{n_1-n_3}$ has been replaced by ℓ_0. We check that the second set of requirements of Definition 5 holds for this sequence, making it a sequence leading from $i_1 \in I_{n_1}$ up to $i_3 \in I_{n_3}$. It is straightforward to check that the requirements are being met, except for $\ell_0 \leqslant h'_{n_1-n_3}$. This follows from the fact that $\ell_0 = i_1$ is related to $h_{n_2-n_1} = i_2$ by a sequence of indices respecting the ordering on the index sets I_{n_1+k} or related by an order-preserving mapping γ_{n_1+k}, and $h'_{n_3-n_1} \in I_{n_1}$ is related to $h'_{n_2-n_3} = i_2$ by a sequence of indices respecting the ordering on the index sets I_{n_3+j} or related by an order-preserving mapping γ_{n_3+j} too, $n_1{-}n_3 \leqslant j \leqslant n_2{-}n_3$. Therefore, we have ℓ_0 '$\leqslant$' $h_{n_2-n_1} = i_2 = \ell'_{n_3-n_2}$ '$\leqslant$' $h'_{n_1-n_3}$. (A more precise and detailed statement can be proven by induction on $n_2{-}n_1$, but is omitted here.) □

The next lemma describes the form of the equivalence class holding a chunk $x = (g, q, i, n)$. Using the cuts, equivalent chunks can be found backwards up to position $n{-}q{+}1$ and forward up to position $n{-}q{+}\#g$. These chunks together, $(g, 1, i_{n-q+1}, n{-}q{+}1)$, …, (g, q, i_n, n), …, $(g, \#g, i_{n-q+\#g}, n-q+\#g)$ span the guide g that is to be applied, in the rewrite sequence to be constructed.

Lemma 2. *Let* $\sigma = (s\ell_n)_{n=1}^{\#u}$ *be a slice sequence for a string* u. *Let* $\mathcal{X} = \{ \, (g_{n,i}, q_{n,i}, i, n) \mid 1 \leqslant n \leqslant \#u, \; i \in I_n \, \}$ *be the set of chunks and choose* $x \in \mathcal{X}$, *say* $x = (g, q, i, n)$. *Put* $p = n{-}q$. *Then there exist* $j_1 \in I_{p+1}, \ldots, j_{\#g} \in I_{p+\#g}$ *such that* $[x] = \{ \, (g, s, j_s, p+s) \mid 1 \leqslant s \leqslant \#g \, \}$.

Proof. It holds that $(g, q, i, n{-}1) \equiv (g', q', i', n)$ iff $g = g'$, $q = q'{-}1$, and $i = \gamma_{n-1}^{-1}(i')$ where $\gamma_{n-1} : I_{n-1} \to I_n$ is the cut for $s\ell_{n-1}$ and $s\ell_n$, while $(g, q, i, n) \equiv (g', q', i', n{+}1)$ iff $g = g'$, $q{+}1 = q'$, and $\gamma_n(i) = i'$, where $\gamma_n : I_n \to I_{n+1}$ is the cut for $s\ell_n$ and $s\ell_{n+1}$. So, choose $j_s = (\, \gamma_{n-q+s}^{-1} \circ \cdots \circ \gamma_{n-1}^{-1} \,)(i)$ for $1 \leqslant s \leqslant q$, and $j_s = (\, \gamma_{n+s} \circ \cdots \circ \gamma_n \,)(i)$ for $q \leqslant s \leqslant \#g$. □

We are now in a position to prove the reverse of Theorem 3.

Theorem 4. *Let* σ *be a slice sequence for a string* u. *Then there exists a guided rewrite sequence* ϱ *for* u *such that* $\mathit{yield}(\varrho) = \mathit{yield}(\sigma)$.

Proof. Suppose $\sigma = (\, s\ell_n \,)_{n=1}^{\#u}$, $s\ell_n = (\, g_{i,n}, q_{i,n} \,)_{i \in I_n}$, for $n = 1, \ldots, \#u$, and let $\mathcal{X} = \{ \, (\, g_{n,i}, q_{n,i}, i, n \,) \mid 1 \leqslant n \leqslant \#u, \; i \in I_n \, \}$ be the corresponding set of chunks. We proceed by induction on $\#\mathcal{X}$. Basis, $\#\mathcal{X} = 0$: In this case every slice is empty and $\mathit{yield}(\sigma) = \mathit{yield}(\, s\ell_1 \,) \cdots \mathit{yield}(\, s\ell_{\#u} \,) = u[1] \cdots u[\#u] = u$ and the empty guided rewrite sequence for u has also yield u.

Induction step, $\#\mathcal{X} > 0$: Clearly, $\mathcal{X}/{\equiv}$ is finite and therefore we can choose, by Lemma 1, $x \in \mathcal{X}$ such that $[x]$ is maximal in $\mathcal{X}/{\equiv}$. By Lemma 2 we can assume $[x] = \{ \, (g, s, i_s, p{+}s) \mid 1 \leqslant s \leqslant \#g \, \}$ for suitable p and indexes $i_s \in I_{p+s}$, for $s = 1, \ldots, \#g$. Note, by maximality of $[x]$, the indexes i_s must be the maximum of I_{p+s}. In particular, $\mathit{yield}(\, \sigma \,)[p{+}s] = \mathit{yield}(\, s\ell_{p+s} \,) = g[s]$, for $s = 1, \ldots, \#g$.

Now, consider the slice sequence $\sigma' = (\, s\ell'_n \,)_{n=1}^{\#u}$ where

$$s\ell'_n = \begin{cases} s\ell_n & \text{for } n = 1, \ldots, p \text{ and } n = p{+}\#g{+}1, \ldots, \#u \\ (\, g_{i,n}, q_{i,n} \,)_{i \in I_n \setminus \{ i_{n-p} \}} & \text{for } n = p{+}1, \ldots, p{+}\#g \end{cases}$$

So, the slice sequence σ' is obtained from the slice sequence σ by leaving out the guide-offset pairs related to the particular occurrence of g.

Let $\mathcal{X}'$ be the set of chunks of σ'. Then $\#\mathcal{X}' < \#\mathcal{X}$. By induction hypothesis we can find a guided rewrite sequence $\varrho' = (\, g'_k, p'_k \,)_{k=1}^{r}$ for u such that $\mathit{yield}(\, \varrho' \,) = \mathit{yield}(\, \sigma' \,)$. Define the guided rewrite sequence $\rho = (\, g_k, p_k \,)_{k=1}^{r+1}$ by $g_k = g'_k$, $p_k = p'_k$ for $k = 1, \ldots, r$ and $g_{r+1} = g$, $p_{r+1} = p$. We have $0 \leqslant p \leqslant \#u{-}\#g$ and $u[p{+}1, p{+}\#g] \sim g$ since $s\ell_{p+1}, \ldots, s\ell_{p+\#g}$ are slices for $u[p{+}1], \dots, u[p{+}\#g]$, respectively. So, ϱ is a well-defined guided rewrite sequence for u.

It holds that $\mathit{yield}(\,\varrho'\,) \Rightarrow_{g,p} \mathit{yield}(\,\varrho\,)$ as ϱ extends ϱ' with the pair (g, p). Therefore,

$$\mathit{yield}(\varrho)[n] = \begin{cases} \mathit{yield}(\varrho')[n] & \text{for } n = 1, \dots, p \text{ and } n = p{+}\#g{+}1, \dots, p{+}\#g \\ g[n{-}p] & \text{for } n = p{+}1, \dots, p{+}\#g \end{cases}$$

From this it follows, for any index n, $1 \leqslant n \leqslant p$ or $p{+}\#g{+}1 \leqslant n \leqslant \#u$, that $\mathit{yield}(\,\varrho\,)[n] = \mathit{yield}(\,\varrho'\,)[n] = \mathit{yield}(\,\sigma'\,)[n] = \mathit{yield}(\,\sigma\,)[n]$, and for any index n, $p{+}1 \leqslant n \leqslant p{+}\#g$, that $\mathit{yield}(\,\varrho\,)[n] = g[n{-}p] = \mathit{yield}(\,\sigma\,)[n]$. As $\#\mathit{yield}(\,\varrho\,) = \#\mathit{yield}(\,\sigma\,) = \#u$, we obtain $\mathit{yield}(\,\varrho\,) = \mathit{yield}(\,\sigma\,)$, as was to be shown. $\square$

For the slice sequence $(s\ell_i)_{i=1}^5$ of Figure 2 we have the following equivalence classes of chunks:

$$\begin{array}{lll} G_3 = \{\,(g_3,1,1,3)\,\} & & G_2 = \{\,(g_2,1,3,2),\ (g_2,2,3,3),\ (g_2,3,3,4)\,\} \\ G_1^1 = \{\,(g_1,1,2,1),\ (g_1,2,2,2)\,\} & & G_1^3 = \{\,(g_1,1,5,4),\ (g_1,2,5,5)\,\} \\ G_1^2 = \{\,(g_1,1,4,1),\ (g_1,2,4,2)\,\} & & G_1^4 = \{\,(g_1,1,6,4),\ (g_1,2,6,5)\,\} \end{array}$$

Moreover, $G_3 \preccurlyeq G_1^1 \preccurlyeq G_2$, $G_2 \preccurlyeq G_1^2$ and $G_2 \preccurlyeq G_1^3 \preccurlyeq G_1^4$. A possible linearization is $G_3 \preccurlyeq G_1^1 \preccurlyeq G_2 \preccurlyeq G_1^3 \preccurlyeq G_1^4 \preccurlyeq G_1^2$. This corresponds to the rewrite sequence

$$\mathit{ebcfa} \Rightarrow_{g_3,2} \mathit{ebdfa} \Rightarrow_{g_1,0} \mathit{fbdfa} \Rightarrow_{g_2,1} \mathit{facea} \Rightarrow_{g_1,3} \mathit{facfb} \Rightarrow_{g_1,3} \mathit{facfb} \Rightarrow_{g_1,0} \mathit{fbcfb}$$

Note that the yield *fbcfb* of this rewrite sequence is the same as the yield of the sequence (1) of Example 1. However, here the second rewrite with g_1 of (1) has been moved to the end now. This does not effect the end result as the particular rewrites do not overlap.

6 Guided rewriting preserves regularity

Given a language L and a set of guides G, according to Definition 1, the language L_G is given as the set $\{\, v \in \Sigma^* \mid \exists u \in L \colon u \Rightarrow^* v \,\}$. Theorem 2 formulated in Section 4, states that if L is regular than L_G is regular too. We will prove the theorem by constructing a non-deterministic finite automaton accepting L_G from a deterministic finite automaton accepting L. The proof exploits the correspondence of rewrite sequences and slice sequences, as captured by Theorem 3 and Theorem 4. First we need an auxiliary result to assure finiteness of the automaton for L_G.

Lemma 3. *Let G be a finite set of guides. Let $Z = \{ s\ell \mid \exists a \in \Sigma\colon s\ell$ repetition-free slice for a with respect to $G \}$. Then Z is finite. Moreover, for every string u and every rewrite sequence ϱ for u, there exists a slice sequence σ for u consisting of slices from Z only such that $\mathit{yield}(\sigma) = \mathit{yield}(\varrho)$.*

Proof. Recall, a slice $s\ell = (g_i, q_i)_{i \in I}$ is repetition-free if, for $i_1, i_2 \in I$, $g_{i_1} = g_{i_2} \wedge q_{i_1} = q_{i_2}$ implies $i_1 = i_2$. Therefore, finiteness of Z is immediate: there are finitely many guide-offset pairs (g, q), hence finitely many repetition-free finite sequences of them. Thus, there are only finitely many repetition-free slices.

Now, let ϱ be a rewrite sequence for a string u. By Theorem 3 we can choose a slice sequence σ' such that $\mathit{yield}(\sigma') = \mathit{yield}(\varrho)$. Suppose $\sigma' = (s\ell_n)_{n=1}^{\#u}$ and $s\ell_n = (g_{i,n}, q_{i,n})_{i \in I_n}$ for $n = 1, \ldots, \#u$. By Lemma 2 it follows that given a repeated guide-offset pair (g, q), say $(g, q) = (g_{i,n}, q_{i,n})$ and $(g, q) = (g_{j,n}, q_{j,n})$ for indexes $i < j$ in I_n, we can delete the complete equivalence class of (g_i, q_i, i, n) from slices $s\ell_{n-q+1}$ to $s\ell_{n-q+\#g}$, while retaining a slice sequence. (In fact, we are removing the 'lower' occurrence of the guide g.) Moreover, the resulting slice sequence has the same yield as for all slices the topmost guide-offset pair remains untouched. The existence of a repetition-free slice sequence σ such that $\mathit{yield}(\sigma) = \mathit{yield}(\sigma')$, hence $\mathit{yield}(\sigma) = \mathit{yield}(\varrho)$, then follows by induction on the number of repetitions. $\square$

As a corollary we obtain that every rewrite sequence ϱ has a repetition-free equivalent ϱ'.

We are now prepared to prove that guided rewriting preserves regularity.

Proof of Theorem 2. Without loss of generality $\varepsilon \notin L$. Let $M = (\Sigma, Q, \rightarrow, q_0, F)$ be a DFA accepting L. We define the NFA $M' = (\Sigma, Q', \rightarrow', q_0, F')$ as follows: Let q_F be a fresh state. Put $Q' = Q \cup (Q \times Z) \cup \{q_F\}$ with Z as given by Lemma 3, $F' = \{q_F\}$ and

$$
\begin{array}{rll}
q_0 \xrightarrow{\varepsilon}{}' q_0 \times \zeta & \text{if } \zeta \text{ is a start slice} \\
q \times \zeta \xrightarrow{b}{}' q' \times \zeta' & \text{if } q \xrightarrow{a} q',\ a \sim \zeta,\ \mathit{yield}(\zeta) = b,\ \zeta \rightsquigarrow \zeta' \\
q \times \zeta \xrightarrow{b}{}' q_F & \text{if } \exists q'\colon q \xrightarrow{a} q' \in F,\ a \sim \zeta,\ \mathit{yield}(\zeta) = b,\ \zeta \text{ is an end slice}
\end{array}
$$

Note, by Lemma 3, Q' is a finite set of states. The automaton M' has only one final state, viz. q_F. In the second type of transition, say with $\zeta = (g_i, q_i)_{i \in I}$ and $\zeta' = (g'_j, q'_j)_{j \in J}$, the requirement $\zeta \rightsquigarrow \zeta'$ implies the existence of a cut $\gamma : I \to J$ in the sense of Definition 4. Thus in a way, the slice ζ' is a follow-up of the slice ζ.

Suppose $v \in L_G$. Then there exist $u = a_1 \cdots a_s \in L$, a rewrite sequence $\varrho = (g_k, p_k)_{k=1}^r$ and strings $u_0, u_1, \ldots, u_r$ such that $u = u_0$, $u_{k-1} \Rightarrow_{g_k, p_k} u_k$ for $k = 1, \ldots, r$, and $v = u_r$. By Theorem 3 there exists an slice sequence that is equivalent to ϱ. Therefore, by Lemma 3, we can assume that a slice sequence σ for u exists with repetition-free slices and such that $\mathit{yield}(\sigma) = \mathit{yield}(\varrho)$. Say $\sigma = (s\ell_n)_{n=1}^s$ and $s\ell_n = (g_{i,n}, q_{i,n})_{i \in I_n}$ for $n = 1, \ldots, s$. Let $q_0 \xrightarrow{a_1} q_1 \cdots \xrightarrow{a_s} q_s \in F$ be an accepting computation of M for u. Then $q_0 \xrightarrow{\varepsilon}{}' q_0 \times s\ell_1 \xrightarrow{b_1}{}' \cdots q_{s-1} \times s\ell_s \xrightarrow{b_s}{}' q_F$ is an accepting computation of M', where $b_n = \mathit{yield}(s\ell_n)$, $1 \leqslant n \leqslant s$. Since we have $b_1 \cdots b_s = \mathit{yield}(s\ell_1) \cdots \mathit{yield}(s\ell_s) = \mathit{yield}(\sigma) = v$, it follows that $v \in \mathcal{L}(M')$. So, $L_G \subseteq \mathcal{L}(M')$.

Let $v = b_1 \cdots b_s$ be a string in $\mathcal{L}(M')$. Given the definition of the transition relation on M', we can find states $q_0, q_1, \ldots, q_{s-1}$, repetition-free slices $s\ell_1, \ldots s\ell_s$ such that $s\ell_n \rightsquigarrow s\ell_{n+1}$ for $n = 1, \ldots, s{-}1$, and a computation $q_0 \xrightarrow{\varepsilon}{}' q_0 \times s\ell_1 \xrightarrow{b_1}{}' \cdots q_{s-1} \times s\ell_s \xrightarrow{b_s}{}' q_F$. Thus, there exist a final state q_s and a computation $q_0 \xrightarrow{a_1} q_1 \cdots q_{s-1} \xrightarrow{a_s} q_s \in F$ such that $a_n \sim s\ell_s$ for $n = 1, \ldots, s$, i.e. $s\ell_n$ is a slice for a_n. Put $u = a_1 \cdots a_s$. Then $u \in L$, $(s\ell_n)_{n=1}^{\#u}$ is a slice sequence for u and $\mathit{yield}(\sigma) = v$. By Theorem 4 we can find a rewrite sequence ϱ for u such that $\mathit{yield}(\varrho) = \mathit{yield}(\sigma) = v$. It follows that $u \Rightarrow^* v$ and $v \in L_G$. Thus, $\mathcal{L}(M') \subseteq L_G$. We conclude that $L_G = \mathcal{L}(M')$ and regularity of L_G follows. $\square$

As a soundness check, observe $L \subseteq L_G$ the automaton M' should accept any word $a_1 \ldots a_s \in L$, $s > 0$. This can be verified as follows. Let ζ_i be the empty slice yielding a_i, $i = 1, \ldots, s$. Then $a_i \sim \zeta_i$, i.e. $a_i = \mathit{yield}(\zeta_i)$, which holds by definition. Moreover, ζ_1 is a start slice, $\zeta_i \rightsquigarrow \zeta_{i+1}$ for $i = 1, \ldots, s{-}1$, and ζ_s is an end slice. It follows that we can turn an accepting computation of M, say $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_s} q_s \in F$ into an accepting computation of M':

$$q_0 \xrightarrow{\varepsilon}{}' q_0 \times \zeta_1 \xrightarrow{a_1}{}' q_1 \times \zeta_2 \xrightarrow{a_2}{}' \cdots \xrightarrow{a_{s-1}}{}' q_{s-1} \times \zeta_s \xrightarrow{a_s}{}' q_F \,.$$

7 Insertion-deletion preserves regularity

This section provides the proof of Theorem 1, regularity of L implies regularity of $L_{i/d}$, exploiting Theorem 2, regularity of L implies regularity of L_G. For the latter theorem to apply we need a preparatory transformation. The point is, in the setting of guided insertion/deletion of 3 strings are allowed to grow or shrink while guided insertions and deletions are being applied, whereas in the setting of guided rewriting of 4 the strings do not change length.

The key idea of the transformation is that every group of 0's is compressed to a single symbol. So, let us fix for the remainder of this section a regular language L over $\Sigma_0 = \Sigma \cup \{0\}$ and a set of guides $G \subseteq \Sigma \cup \Sigma \cdot \Sigma_0^* \cdot \Sigma$. Let N be the maximum number of consecutive 0's occurring in the elements of G. Then we introduce $N+2$ fresh symbols $0_0, 0_1, \dots, 0_N, 0_+$ and put $\Theta = \{\, 0_0,\, 0_1,\, \dots,\, 0_N, 0_+ \,\}$.

For a string u over Σ_0 we define the string $\bar{u}$ over the alphabet $\overline{\Sigma} = \Sigma \cup \Theta$. The string $\overline{u}$ is obtained from u by replacing every maximal pattern 0^i by the single symbol 0_i, in case $i \leqslant N$, and by 0_+ in case $i > N$. More precisely, if $a_1, \dots, a_n \in \Sigma$ and $u = 0^{k_0} a_1 0^{k_1} a_2 \cdots a_n 0^{k_n}$ with $k_i \geqslant 0$ for $i = 0, \dots, n$, then $\overline{u} = 0_{p_0} a_1 0_{p_1} a_2 \cdots a_n 0_{p_n}$ where $p_i = k_i$ if $0 \leqslant k_i \leqslant N$ and $p_i = +$ if $k_i > N$, for $i = 0, \dots, n$. For such $u = 0^{k_0} a_1 0^{k_1} a_2 \cdots a_n 0^{k_n}$ we write $\mathsf{zeros}(u, i) = k_i$ for $i = 0, \dots, n$. For a set V of strings over Σ_0, we put $\overline{V} = \{\, \bar{v} \mid v \in V \,\}$.

Lemma 4. *If $L \subseteq \Sigma_0^*$ is regular, then $\bar{L} \subseteq \overline{\Sigma}^*$ is regular as well.*

Proof. Let $M = (Q, \Sigma_0, \delta, q_0, F)$ be an NFA accepting L. Obtain the NFA M' from M by putting $M' = (Q, \Sigma \cup \Theta, \delta', q_0, F)$ where

$$\begin{array}{lcl} \delta'(q, \alpha) & = & \delta(q, \alpha) \quad \text{for } \alpha \in \Sigma \cup \{\varepsilon\} \\ \delta'(q, 0_i) & = & \{\, q' \in Q \mid q \overset{0^i}{\Longrightarrow} q' \,\} \quad \text{for } 0 \leqslant i \leqslant N \\ \delta'(q, 0_+) & = & \{\, q' \in Q \mid \exists i > N \colon q \overset{0^i}{\Longrightarrow} q' \,\} \end{array}$$

In particular, we have $q \xrightarrow{0_0}{}' \; q$ for all $q \in Q$. We claim $\bar{L} = \mathcal{L}(M') \cap \Theta \cdot (\Sigma \cdot \Theta)^*$.

Pick $\overline{u} \in \bar{L}$. Suppose $u = 0^{k_0} a_1 0^{k_1} \dots a_n 0^{k_n} \in L$ with $0 \leqslant k_i$ for $i = 0, \dots, n$. Then we have $\overline{u} = 0_{p_0} a_1 0_{p_1} \dots a_n 0_{p_n}$ for suitable indices $p_i \in \{\, 0, \dots, N, + \,\}$. Let

$$q_0 \overset{0^{k_0}}{\Longrightarrow} q_1' \xrightarrow{a_1} q_1 \overset{0^{k_1}}{\Longrightarrow} \cdots q_n' \xrightarrow{a_n} q_n \overset{0^{k_n}}{\Longrightarrow} q_{n+1}' \in F$$

be an accepting computation of M for u. Then

$$q_0 \xrightarrow{0_{p_0}}{}' \; q_1' \xrightarrow{a_1}{}' \; q_1 \xrightarrow{0_{p_1}}{}' \; \cdots \; q_n' \xrightarrow{a_n}{}' \; q_n \xrightarrow{0_{p_n}}{}' \; q_{n+1}' \in F$$

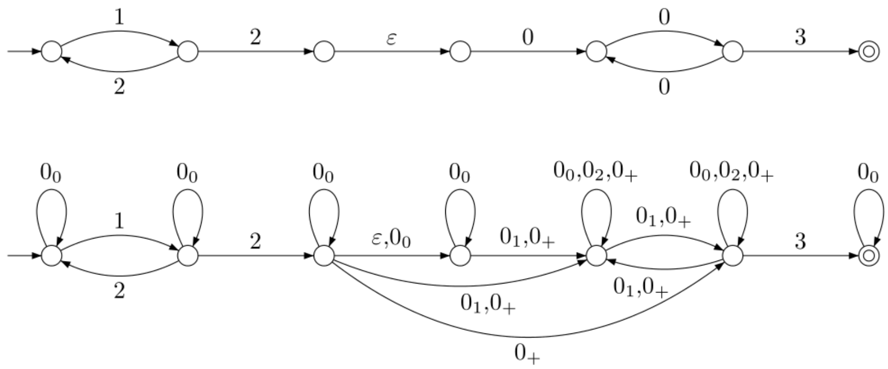


Figure 3: Example automaton construction as used in Lemma 4, with $N = 2$

is an accepting computation of M' for $\overline{u}$. So $\overline{u} \in \mathcal{L}(M')$. Clearly, $\overline{u} \in \Theta \cdot (\Sigma \cdot \Theta)^*$. Thus $\bar{L} \subseteq \mathcal{L}(M') \cap \Theta \cdot (\Sigma \cdot \Theta)^*$.

Conversely, pick $v \in \mathcal{L}(M') \cap \Theta \cdot (\Sigma \cdot \Theta)^*$. Say $v = 0_{p_0} a_1 0_{p_1} \cdots a_n 0_{p_n}$ for some $n \geqslant 0$, $p_0, \ldots, p_n \in \{0, \ldots, N, +\}$ and $a_1, \ldots, a_n \in \Sigma$. Since $v \in \mathcal{L}(M')$ there exists an accepting computation

$$q_0 \xrightarrow{0_{p_0}}{}' \; q_1' \xrightarrow{a_1}{}' \; q_1 \xrightarrow{0_{p_1}}{}' \; \cdots \; q_n' \xrightarrow{a_n}{}' \; q_n \xrightarrow{0_{p_n}}{}' \; q_{n+1}' \in F$$

of M' for v. Then, by construction of M', there also exists an accepting computation

$$q_0 \overset{0^{k_0}}{\Longrightarrow} q_1' \xrightarrow{a_1} q_1 \overset{0^{k_1}}{\Longrightarrow} \cdots q_n' \xrightarrow{a_n} q_n \overset{0^{k_n}}{\Longrightarrow} q_{n+1}' \in F$$

of M for suitable indices $k_0, \ldots, k_n$ such that $k_i = p_i$ if $p_i \in \{ 0, \ldots, N \}$, and $k_i > N$ if $p_i = +$, for $i = 0, \ldots, n$. Therefore, $u = 0^{k_0} a_1 0^{k_1} \cdots a_n 0^{k_n} \in \mathcal{L}(M)$, i.e. $u \in L$. Moreover, by the correspondence of $k_0, \ldots, k_n$ and $p_0, \ldots, p_n$, respectively, it holds that $\overline{u} = v$, hence $v \in \bar{L}$. Thus $\mathcal{L}(M') \cap \Theta \cdot (\Sigma \cdot \Theta)^* \subseteq \bar{L}$.

Conclusion: $\bar{L} = \mathcal{L}(M') \cap \Theta \cdot (\Sigma \cdot \Theta)^*$ and $\bar{L}$ is regular, being the intersection of two regular languages. □

The construction from the proof is illustrated in Figure 3.

We consider the adjustment relation $\sim_0$ on $\overline{\Sigma}$ defined by

$$a \sim_0 b \iff a = b \vee (a \in \Theta \wedge b \in \Theta)$$

and guided rewriting $\Rightarrow$ on $\overline{\Sigma}$ with respect to $\sim_0$ and the set of guides $\overline{G}$. However, for elements of $\overline{G}$ the leading and trailing 0_0's are removed, so

$$\overline{G} \;=\; \{\, a_1 0_{k_1} a_2 \cdots 0_{k_{n-1}} a_n \mid a_1 0^{k_1} a_2 \cdots 0^{k_{n-1}} a_n \in G \,\}$$

Note, the correspondence of the index k_i in 0_{k_i} and the index k_i in 0^{k_i} is literally, since always $0 \leqslant k_i \leqslant N$. Moreover, if, for $u, v \in \Sigma_0^*$, we have $\overline{u} \sim_0 \overline{v}$, then also $\pi(u) = \pi(v)$.

Lemma 5. *Let $u, v \in \Sigma_0^*$.*

(a) If $u \Rightarrow_{i/d} v$ then $\bar{u} \Rightarrow \bar{v}$.

(b) Conversely, if $\overline{u} \Rightarrow \overline{v}$, and $\mathsf{zeros}(v,i) > N$ implies $\mathsf{zeros}(v,i) = \mathsf{zeros}(u,i)$, for all i, then $u \Rightarrow_{i/d} v$.

Proof. (a) By definition, if $u \Rightarrow_{i/d} v$ with respect to G, then there exist strings $x, z \in \Sigma_0^*$, $y \in \Sigma \cdot \Sigma_0^* \cdot \Sigma$, and $g \in G$ such that $\pi(y) = \pi(g)$, $u = xyz$ and $v = xgz$. Say

$$x = 0^{k_0} a_1 0^{k_1} \,\cdots\, a_s 0^{k_s}, \quad y = b_1 0^{\ell_1} \cdots 0^{\ell_{r-1}} b_r\,, \\ z = 0^{m_0} c_1 0^{m_1} \,\cdots\, c_q 0^{m_q}, \quad \text{and} \quad g = b_1 0^{\ell'_1} \cdots 0^{\ell'_{r-1}} b_r$$

for $s, q \geqslant 0$, $r \geqslant 1$, $0 \leqslant k_0, \ldots, k_s$, $\ell_1, \ldots, \ell_{r-1}$, $m_0, \ldots, m_q$ and $a_1, \ldots, a_s, b_1, \ldots, b_r, c_1, \ldots, c_q \in \Sigma$. Then we have

$$\overline{x} = 0_{pk_0} a_1 0_{pk_1} \cdots a_s 0_{pk_s}\,, \quad \overline{y} = b_1 0_{p\ell_1} \cdots 0_{p\ell_{r-1}} b_r\,, \\ \overline{z} = 0_{pm_0} c_1 0_{pm_1} \cdots c_q 0_{pm_q}\,, \quad \text{and} \quad \bar{g} = b_1 0_{p\ell'_1} \cdots 0_{p\ell'_{r-1}} b_r$$

for suitable indices $pk_i, p\ell_j, p\ell'_j, pm_k \in \{\, 0, \ldots, N, + \,\}$. By definition of $\sim_0$ we have $0_{p\ell_j} \sim_0 0_{p\ell'_j}$ for $1 \leqslant j \leqslant r$. Hence $\overline{y} \sim_0 \overline{g}$ and $\overline{u} = \overline{x}\,\overline{y}\,\overline{z} \sim_0 \overline{x}\,\bar{g}\,\overline{z} = \overline{v}$.

(b) Suppose

$$u = 0^{k_0} a_1 0^{k_1} \ldots a_n 0^{k_n} \text{ and } v = 0^{\ell_0} b_1 0^{\ell_1} \ldots b_m 0^{\ell_m}$$

for $n, m \geqslant 0$, $k_0, \ldots, k_n, \ell_0, \ldots, \ell_m \geqslant 0$ and $a_1, \ldots, a_n, b_1, \ldots, b_m \in \Sigma$. Then

$$\overline{u} = 0_{pk_0} a_1 0_{pk_1} \ldots a_n 0_{pk_n} \text{ and } \overline{v} = 0_{p\ell_0} b_1 0_{p\ell_1} \ldots b_m 0_{p\ell_m}$$

for suitable $pk_i, p\ell_j \in \{ 0, \ldots, N, + \}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$. Assuming $\overline{u} \Rightarrow \overline{v}$ with respect to $\sim_0$, we have $n = m$, $a_i = b_i$ for $1 \leqslant i \leqslant n$. Moreover, there exist indices r and s, $1 \leqslant r < s \leqslant n$ such that

$$\begin{array}{rcl} 0_{pk_0} a_1 0_{pk_1} \ldots a_{r-1} 0_{pk_{r-1}} & = & 0_{p\ell_0} a_1 0_{p\ell_1} \ldots a_{r-1} 0_{p\ell_{r-1}} \\ a_r 0_{pk_r} \ldots 0_{pk_{s-1}} a_s & \sim_0 & a_r 0_{p\ell_r} \ldots 0_{p\ell_{s-1}} a_s \\ & a_r 0_{p\ell_r} \ldots 0_{p\ell_{s-1}} a_s \in \overline{G} & \\ 0_{pk_s} a_{s+1} \ldots a_n 0_{pk_n} & = & 0_{p\ell_s} a_{s+1} \ldots a_n 0_{p\ell_n} \end{array}$$

It follows that $pk_i = p\ell_i$ for $1 \leqslant i < r$ and $pk_j = p\ell_j$ for $s \leqslant j \leqslant n$. If $pk_i \neq +$ or $pk_j \neq +$ this implies $k_i = \ell_i$ and $k_j = \ell_j$. If view of the additional assumption that $\mathsf{zeros}(v,i) > N$ implies $\mathsf{zeros}(v,i) = \mathsf{zeros}(u,i)$ for $1 \leqslant i \leqslant n$, it follows that $k_i = \ell_i$ for all $1 \leqslant i < r$ and $k_j = \ell_j$ for all $s \leqslant j \leqslant n$. Now, put

$$x = 0^{k_0} a_1 0^{k_1} \ldots a_{r-1} 0^{k_{r-1}}, \quad y = a_r 0^{k_r} \ldots 0^{k_{s-1}} a_s, \\ \text{and} \quad z = 0^{k_s} a_{s+1} \ldots a_n 0^{k_n}$$

Choose $g \in G$ such that $\bar{g} = 0_0 a_r 0_{p\ell_r} \ldots 0_{p\ell_{s-1}} a_s 0_0$. Say $g = a_r 0^{\ell'_r} \ldots 0^{\ell'_{s-1}} a_s$. Since $p\ell_r, \ldots, p\ell_{s-1} \neq +$ it holds that $\ell_i = p\ell_i = \ell'_i$ for $r \leqslant i < s$, i.e. $g = a_r 0^{\ell_r} \ldots 0^{\ell_{s-1}} a_s$. Thus we have $u = xyz$, $v = xgz$ and $\pi(y) = \pi(g)$. Hence $u \Rightarrow_{i/d} v$ with respect to G. □

Lemma 6. *It holds that*

$$L_{i/d} = \{ v \in \Sigma_0^* \mid \exists u \in L \colon \overline{u} \Rightarrow^* \overline{v} \wedge \\ \forall i \colon (\mathsf{zeros}(v,i) > N \to \mathsf{zeros}(v,i) = \mathsf{zeros}(u,i)) \}$$

where $\Rightarrow$ *is the guided rewriting relation with respect to* $\overline{G}$.

Proof. ($\subseteq$). Let $v \in L_{i/d}$. Thus, there exists $u \in L$ such that $u \Rightarrow_{i/d}^* v$. Using the first claim of Lemma 5 we obtain $\overline{u} \Rightarrow^* \overline{v}$. From $u \in L$ we conclude $\overline{u} \in \bar{L}$ and therefore $\overline{v} \in \bar{L}_{\overline{G}}$. Furthermore, if $\mathsf{zeros}(v,i) > N$ then in $u \Rightarrow_{i/d}^* v$ the corresponding group of $\mathsf{zeros}(v,i)$ many consecutive 0's is not touched, so $\mathsf{zeros}(v,i) = \mathsf{zeros}(u,i)$. This concludes ($\subseteq$).

($\supseteq$). Let $u \in L$ satisfy $\bar{u} \Rightarrow^n \bar{v}$ and $\forall i \colon (\mathsf{zeros}(v,i) > N \to \mathsf{zeros}(v,i) = \mathsf{zeros}(u,i))$, for $n \geqslant 0$. We will prove $u \Rightarrow_{i/d}^n v$ by induction on n. For the base case $n = 0$ this follows from $\bar{u} = \bar{v}$, the definition of the mapping $\bar{-}$ and the assumption $(\mathsf{zeros}(v,i) > N \to \mathsf{zeros}(v,i) = \mathsf{zeros}(u,i))$. For the induction step, $n > 0$, suppose $\overline{u} \Rightarrow^{n-1} \overline{w} \Rightarrow \overline{v}$. For every i, observe if $\mathsf{zeros}(w,i) > N$ then $\mathsf{zeros}(u,i) > N$. Now choose w' such that $\overline{w}' = \overline{w}$ and, for all i, if $\mathsf{zeros}(w,i) > N$ then $\mathsf{zeros}(w',i) = \mathsf{zeros}(u,i)$, otherwise $\mathsf{zeros}(w',i) = \mathsf{zeros}(w,i)$. Applying the induction hypothesis on $\overline{u} \Rightarrow^{n-1} \overline{w}'$ yields $u \Rightarrow_{i/d}^{n-1} w'$, and applying Lemma 5 to $\overline{w}' \Rightarrow \overline{v}$ yields $w' \Rightarrow_{i/d} v$, so $u \Rightarrow_{i/d}^{n} v$. □

A direct consequence of Lemma 6 is the following corollary.

Corollary 1. *In the setting above, let $v = 0^{m_0} a_1 0^{m_1} \cdots a_n 0^{m_n} \in \Sigma_0^*$, where $a_i \in \Sigma$ for $i = 1, \ldots, n$. Then $v \in L_{i/d}$ iff $\overline{v} = 0_{p_0} a_1 0_{p_1} \cdots a_n 0_{p_n} \in \bar{L}_{\overline{G}}$ and $u = 0^{k_0} a_1 0^{k_1} \cdots a_n 0^{k_n} \in L$ exists such that $k_i = p_i$ if $p_i \in \{0, \ldots, N\}$ and $k_i = m_i$ if $p_i = +$, for $i = 0, \ldots, n$.* □

Now we are ready to construct, given an NFA M for the language L over Σ_0, an NFA $M_{i/d}$ exactly accepting the language $L_{i/d}$.

Suppose $M = (\Sigma_0, Q, \rightarrow, q_0, F)$. According to Lemma 4 we have that $\bar{L}$ is regular. By Theorem 2 we obtain that $\bar{L}_{\overline{G}}$ is regular. So, let $\overline{M} = (\overline{\Sigma}, \overline{Q}, \rightarrow, \overline{q}_0, \overline{F})$ be an NFA accepting $\bar{L}_{\overline{G}}$. According to Lemma 6, $L_{i/d}$ consists of strings v such that $\overline{v} \in \bar{L}_{\overline{G}}$, thus for some $u \in L$ we have $\overline{u} \Rightarrow^* \overline{v}$. By mapping v to $\overline{v}$ every maximal group of k 0's with $k > N$ is mapped to 0_+; the extra requirement for being in $L_{i/d}$ is that the size of such a group coincides with the size of the corresponding group in the original string $u \in L$. This leads to the following construction of the NFA $M_{i/d}$ for $L_{i/d}$.

Definition 6. *Suppose $M = (\Sigma_0, Q, \rightarrow, q_0, F)$ and $\overline{M} = (\overline{\Sigma}, \overline{Q}, \rightarrow, \overline{q}_0, \overline{F})$ are NFA's for the languages L and $\bar{L}_{\overline{G}}$, respectively. Then, the NFA $M_{i/d}$ is defined as follows:*

- *the set of states of $M_{i/d}$ is $Q \times \overline{Q} \times \{ 0, \ldots, N \}$*
- *the transition relation of $M_{i/d}$ is given by*

1. *if $q \xrightarrow{a} r$ and $\overline{q} \xrightarrow{a} \overline{r}$ then $(q, \overline{q}, 0) \xrightarrow{a} (r, \overline{r}, 0)$, for $a \in \Sigma$, $q, r \in Q$, $\overline{q}, \overline{r} \in \overline{Q}$*
2. *if $q \xrightarrow{0^*} r$ (zero or more 0-steps) and $\overline{q} \xrightarrow{0_k} \overline{r}$ then $(q, \overline{q}, 0) \xrightarrow{0^k} (r, \overline{r}, 0)$, for $q, r \in Q$, $\overline{q}, \overline{r} \in \overline{Q}$, $k \in \{ 0, \ldots, N \}$. More specifically, for $k = 0$ we have a transition $(q, \overline{q}, 0) \xrightarrow{\varepsilon} (r, \overline{r}, 0)$, for $k = 1$ we have a transition $(q, \overline{q}, 0) \xrightarrow{0} (r, \overline{r}, 0)$, and for $k > 1$ we create $k-1$ fresh states and a path consisting of k 0-steps along these fresh states from $(q, \overline{q}, 0)$ to $(r, \overline{r}, 0)$.*

3. *if* $q \xrightarrow{0} r$ *and* $\overline{q} \xrightarrow{0_+} \overline{r}$ *then* $(q, \bar{q}, 0) \xrightarrow{0} (r, \bar{r}, N)$*, for* $q, r \in Q$*,* $\overline{q}, \overline{r} \in \overline{Q}$

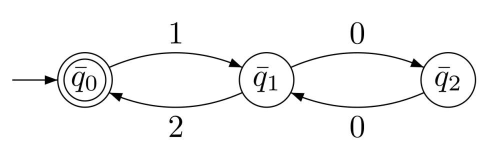
4. *if* $q \xrightarrow{0} r$ *then* $(q, \overline{r}, N) \xrightarrow{0} (r, \overline{r}, N)$ *and* $(q, \overline{r}, i) \xrightarrow{0} (r, \overline{r}, i{-}1)$ *for* $q, r \in Q$*,* $\bar{r} \in \overline{Q}$ *and* $i = 1, \ldots, N$

- *the initial state of* $M_{i/d}$ *is* $(q_0, \bar{q}_0, 0)$

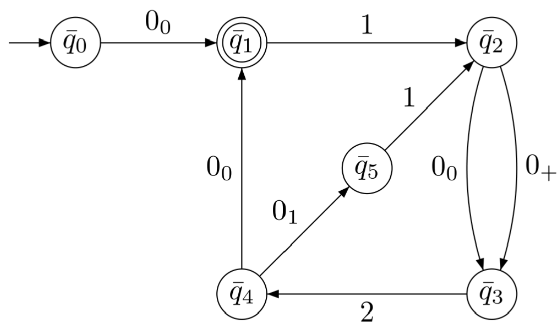
- *the set of final states of* $M_{i/d}$ *is* $F \times \overline{F} \times \{0\}$*.*

The idea of this NFA $M_{i/d}$ is that as long as 0_+ does not come into play, it exactly mimics the NFA $\overline{M}$ executing an a-step, based on rule 1, or replacing every 0_k by k separate 0-steps, based on rule 2. When it has to simulate a 0_+-step of $\overline{M}$, it performs a number of 0-steps as following the NFA M. However, it has to be guaranteed that this latter number is indeed more than N. The third component of states of $M_{i/d}$ serves this purpose. When $M_{i/d}$ takes a transition based on rule 3 above executing a 0-step, the third component is set to it maximal value N. Next, based on the two types of transitions covered by rule 4, either the value N is maintained to cater for a sequence of more than $N{+}1$ zeros, or it is counted down to 0 taking exactly N steps, yielding $N + 1$ zeros at least. We illustrate the behaviour of $M_{i/d}$ by an example.

The language $L = \mathcal{L}(\,(1(00)^*2)^*)$ is accepted by the NFA M:



Let $G = \{201\}$. Then we have $N = 1$, and $\bar{L} = \mathcal{L}(\,0_0(1(0_0 + 0_+)20_0)^*)$. For closure under $\overline{G}$ every substring of the shape 20_01 of a string in $\overline{L}$ may be replaced by 20_11, yielding the following NFA $\overline{M}$ accepting $\overline{L}_{\overline{G}}$:



The automaton $M_{i/d}$ that is constructed as a product of the automata M and $\overline{M}$ is depicted in Figure 4.

For example, the transition $(q_0, \overline{q}_0, 0) \xrightarrow{\varepsilon} (q_0, \overline{q}_1, 0)$ of $M_{i/d}$ is based on the transition $q_0 \stackrel{\varepsilon}{\Longrightarrow} q_0$ of M and the transition $\overline{q}_0 \xrightarrow{0_0} \overline{q}_1$ of $\overline{M}$ using rule 2, while the transition $(q_0, \overline{q}_1, 0) \xrightarrow{1} (q_1, \overline{q}_2, 0)$ of $M_{i/d}$ is based on the transition $q_0 \xrightarrow{1} q_1$ and $\overline{q}_1 \xrightarrow{1} \overline{q}_2$ of $\overline{M}$ using rule 1. The two transitions leaving $(q_1, \overline{q}_2, 0)$ in $M_{i/d}$ correspond to the two transitions for $\overline{q}_2$ in $\overline{M}$, one labeled with 0_0 and one labeled with 0_+. The former induces the transition ε-transition to state $(q_1, \overline{q}_3, 0)$ based on rule 2, the latter however induces the 0-transition to state $(q_2, \overline{q}_3, 1)$ where the counter in the third component is set to $N = 1$. This indicates that at least one more zero needs to be matched. In $M_{i/d}$ this can be done in two ways, either by looping via state $(q_1, \overline{q}_3, 1)$ or by going to state $(q_1, \overline{q}_3, 0)$ directly. Here, the transitions are based on rule 4 of Definition 6. Note that state $(q_0, \overline{q}_3, 0)$ is a deadlock state. Finally, in state $(q_0, \overline{q}_4, 0)$ two transitions are possible again. This reflects that at this point an insertion/deletion step may take place or not. If so, the computation continues via state $(q_0, \overline{q}_5, 0)$. If not, the computation proceeds with an ε-transition to state $(q_0, \overline{q}_1, 0)$.

More concretely, consider the string $10000212 \in L$. It admits an insertion/deletion step with the guide 201 to the string 100002012. So, $100002012 \in L_{i/d}$. For accepting 100002012 by $M_{i/d}$ it is essential to rely for processing the part 100002 on M, since with respect to $\overline{\Sigma}$ every group of more than one consecutive 0's is compressed to 0_+, by which in $\overline{M}$ the information is lost that we should have an even number of 0's. This is handled in the $(q_2, \overline{q}_3, 1)$–$(q_1, \overline{q}_3, 1)$ loop. For the rest of the string the automaton $\overline{M}$ should be followed, since in M the information that 21 was allowed to be replaced by 201 is not available. The 0-transition leaving of state $(q_0, \overline{q}_4, 0)$ makes

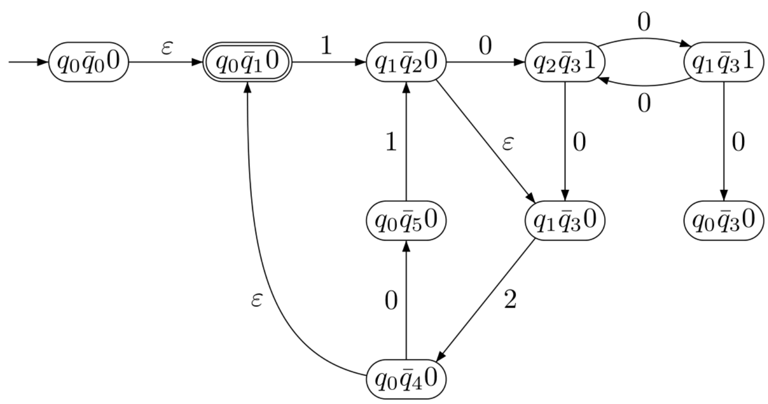


Figure 4: Automaton $M_{i/d}$

this possible. The resulting accepting transition sequence in $M_{i/d}$ reads:

$$\begin{aligned}(q_0,\bar{q}_0,0) \xrightarrow{\epsilon} (q_0,\bar{q}_1,0) \xrightarrow{1} (q_1,\bar{q}_2,0) \xrightarrow{0} (q_2,\bar{q}_3,1) \xrightarrow{0} (q_1,\bar{q}_3,1) \xrightarrow{0} \\ (q_2,\bar{q}_3,1) \xrightarrow{0} (q_1,\bar{q}_3,0) \xrightarrow{2} (q_0,\bar{q}_4,0) \xrightarrow{0} (q_0,\bar{q}_5,0) \xrightarrow{1} \\ (q_1,\bar{q}_2,0) \xrightarrow{\epsilon} (q_1,\bar{q}_3,0) \xrightarrow{2} (q_0,\bar{q}_4,0) \xrightarrow{\epsilon} (q_0,\bar{q}_1,0)\end{aligned}$$

Since $(q_0,\overline{q}_1,0)$ is a final state in $M_{i/d}$, this shows that $100002012 \in \mathcal{L}(M_{i/d})$.

For a formal proof that $L_{i/d} = \mathcal{L}(M_{i/d})$ we need the following lemma.

Lemma 7. *Suppose* $\overline{q} \xrightarrow{0_+} \overline{r}$ *for* $\overline{q},\overline{r} \in \overline{Q}$*. Then* $q \overset{0}{\Longrightarrow}{}^k\, r$ *in* M *iff* $\exists q' \in Q$*:* $(q,\overline{q},0) \xrightarrow{0} (q',\overline{r},N) \overset{0}{\Longrightarrow}{}^{k-1} (r,\overline{r},0)$ *in* $M_{i/d}$*, for all* $q,r \in Q$ *and* $k > N$*.*

Proof. Suppose $q \overset{0}{\Longrightarrow}{}^k\, r$ in M. Then there exists $q',r' \in Q$ such that

$$q \;\xrightarrow{0}\; q' \;\overset{0}{\Longrightarrow}{}^{k-N-1}\; r' \;\overset{0}{\Longrightarrow}{}^{N}\; r$$

and since $\overline{q} \xrightarrow{0_+} \overline{r}$ one has $(q,\overline{q},0) \xrightarrow{0} (q',\overline{r},N)$. Next we get

$$(q',\bar{r},N) \;\overset{0}{\Longrightarrow}{}^{k-N-1}\; (r',\bar{r},N) \;\overset{0}{\Longrightarrow}{}^{N}\; (r,\bar{r},0)$$

in which in each of the last N steps the third argument decreases by 1. Since $1+(k{-}N{-}1)+N = k$, this proves the implication from right to left.

Conversely, suppose $(q,\overline{q},0) \xrightarrow{0} (q',\overline{r},N) \overset{0}{\Longrightarrow}{}^{k-1} (r,\overline{r},0)$ Then, by definition of $M_{i/d}$, we then have $q \xrightarrow{0} q' \overset{0}{\Longrightarrow}{}^{k-1} r$ in M. □

Now we are in a position to provide a proof of Theorem 1.

Proof of Theorem 1. We show that, in the above setting, $L_{i/d} = \mathcal{L}(M_{i/d})$. Suppose $v = 0^{m_0}a_1 0^{m_1} \cdots a_n 0^{m_n} \in L_{i/d}$ where $a_i \in \Sigma$ for $i = 1,\ldots,n$ and $m_i \geqslant 0$ for $i = 0,\ldots,n$. Write $\overline{v} = 0_{p_0}a_1 0_{p_1} \cdots a_n 0_{p_n}$ with $p_i \in \{0,\ldots,N,+\}$ corresponding to m_i, for $i = 0,\ldots n$. Then by Corollary 1 we have $\overline{v} \in \bar{L}_{\overline{G}}$ and for some $u = 0^{k_0}a_1 0^{k_1} \cdots a_n 0^{k_n} \in L$ we have that $k_i = p_i$ if $p_i \in \{0,\ldots,N\}$ and $k_i = m_i$ and $k_i > N$ if $p_i = +$, for $i = 0,\ldots,n$. Thus, $u \in L$ is accepted by M and $\overline{v} \in \bar{L}_{\overline{G}}$ is accepted by $\overline{M}$. So, next to the initial states q_0 of M and $\overline{q}_0$ of $\overline{M}$, there exist $q_1,\ldots,q_n$, $r_0,\ldots,r_n \in Q$ and $\overline{q}_1,\ldots,\overline{q}_n$, $\overline{r}_0,\ldots,\overline{r}_n \in \overline{Q}$ such that

$$\begin{aligned}
&q_i \overset{0}{\Longrightarrow}{}^{k_i} r_i \text{ in } M \text{ and } \overline{q}_i \xrightarrow{0_{p_i}} \overline{r}_i \text{ in } \overline{M}, \text{ for } i = 0,\ldots,n\\
&r_{i-1} \xrightarrow{a_i} q_i \text{ in } M \text{ and } \overline{r}_{i-1} \xrightarrow{a_i} \overline{q}_i \text{ in } \overline{M}, \text{ for } i = 1,\ldots,n\\
&r_n \in F \text{ and } \overline{r}_n \in \overline{F}
\end{aligned}$$

We observe: (i) $(q_i,\overline{q}_i,0) \overset{0}{\Longrightarrow}{}^{m_i} (r_i,\overline{r}_i,0)$, for $i = 0,\ldots,n$. In case $p_i \in \{0,\ldots,N\}$ this follows from $m_i = k_i = p_i$ and second transition type of Definition 6 for $M_{i/d}$. In case $p_i = +$ this follows from Lemma 7. (ii) $(r_{i-1},\overline{r}_{i-1},0) \xrightarrow{a_i} (q_i,\overline{q}_i,0)$, for $i = 1,\ldots,n$ based on the first type of transition for $M_{i/d}$. Thus, $v = 0^{m_0}a_1 0^{m_1} \cdots a_n 0^{m_n} \in \mathcal{L}(M_{i/d})$.

Conversely, pick $v = 0^{m_0}a_1 0^{m_1} \cdots a_n 0^{m_n} \in \mathcal{L}(M_{i/d})$. Then there are states $q_1,\ldots,q_n$, $r_0,\ldots,r_n \in Q$ and $\overline{q}_1,\ldots,\overline{q}_n$, $\overline{r}_0,\ldots,\overline{r}_n \in \overline{Q}$ such that $r_n \in F$ and $\overline{r}_n \in \overline{F}$ and $(q_i,\overline{q}_i,0) \overset{0}{\Longrightarrow}{}^{m_i} (r_i,\overline{r}_i,0)$ for $i = 0,\ldots,n$, and $(r_{i-1},\overline{r}_{i-1},0) \xrightarrow{a_i} (q_i,\overline{q}_i,0)$ for $i = 1,\ldots,n$, using that a_i-steps in $M_{i/d}$ are only possible from and to states having 0 as their third coordinate. From $(r_{i-1},\overline{r}_{i-1},0) \xrightarrow{a_i} (q_i,\overline{q}_i,0)$ we conclude $r_{i-1} \xrightarrow{a_i} q_i$ in M, and $\overline{r}_{i-1} \xrightarrow{a_i} \overline{q}_i$ in $\overline{M}$, for $i = 1,\ldots,n$.

Using $\mathcal{L}(\overline{M}) = \bar{L}_{\overline{G}} \subseteq \mathcal{L}(\Theta\cdot(\Sigma\cdot\Theta)^*)$, we may assume without loss of generality that every $\overline{q} \in \overline{Q}$ only has either only outgoing Θ transitions and incoming Σ transitions in $\overline{M}$, or vice versa. Here, the states $\overline{q}_i$ are of the first type and the states $\overline{r}_i$ are of the second type. Moreover, without loss of generality, we may assume that for every two states $\overline{q},\overline{r} \in \overline{Q}$ there is at most one Θ-transition from $\overline{q}$ to $\overline{r}$ in $\overline{M}$. Now, by the form of the rules in Definition 6, from $(q_i, \overline{q}_i, 0) \xRightarrow{0^{m_i}} (r_i, \overline{r}_i, 0)$ we conclude $q_i \xRightarrow{0^{k_i}} r_i$ and $\overline{q}_i \xrightarrow{0_{p_i}} \overline{r}_i$ for $p_i \in \{0, \ldots, N, +\}$, for some $k_i \geqslant 0$, for $i = 0, \ldots, n$. So,

$$\begin{array}{ccccccccc} q_0 & \xRightarrow{0^{k_0}} r_0 & \xrightarrow{a_1} q_1 & \cdots & r_{n-1} & \xrightarrow{a_n} & q_n & \xRightarrow{0^{k_n}} r_n \in F \\ \overline{q}_0 & \xrightarrow{0_{p_0}} \overline{r}_0 & \xrightarrow{a_1} \overline{q}_1 & \cdots & \overline{r}_{n-1} & \xrightarrow{a_n} & \overline{q}_n & \xrightarrow{0_{p_n}} \overline{r}_n \in \overline{F} \end{array}$$

Therefore, $u = 0^{k_0} a_1 0^{k_1} \cdots a_n 0^{k_n} \in L$ and $\overline{v} = 0_{p_0} a_1 0_{p_1} \cdots a_n 0_{p_n} \in \bar{L}_{\overline{G}}$. If $p_i \in \{0, \ldots, N\}$, then by the assumptions on the transitions in $\overline{M}$ and uniqueness of Θ steps, we conclude $m_i = p_i$. If $p_i = +$, then from Lemma 7 we conclude that $m_i = k_i$. Since, by construction, from a state with its third coordinate of value N it takes at least N steps to get down at 0, we conclude $k_i > N$. This holds for all $i = 0, \ldots, n$. Therefore, by Corollary 1 we conclude that $v = 0^{m_0} a_1 0^{m_1} \cdots a_n 0^{m_n} \in L_{i/d}$. $\square$

8 Related work and concluding remarks

In this paper we discussed specific concepts of string rewriting: a more flexible notion focusing on insertions and deletions of a dummy symbol, another more strict notion based on an equivalence relation. Given a language L we considered the extended languages $L_{i/d}$ and L_G comprising the closure of L for the two types of guided rewriting with guides from a finite set G. In particular, as our main results we proved that these closures preserve regularity. For doing so we investigated the local effect of guided rewriting on two consecutive string positions, leading to a novel notion of a slice sequence. Finally, the theorem for adjustment-based rewriting was proved by an automaton construction exploiting a slice sequence characterization of guided rewriting. Via a compression scheme for strings of dummy symbols, the theorem for guided insertion/deletion followed.

Preservation of regularity by closing a language with respect to a given notion of rewriting arises as a natural question. In Section 3 we observed that by closing the regular language $\mathcal{L}(\,(ab)^*\,)$ under rewriting with respect to the single rewrite rule $ba \to ab$ the resulting language is not regular. So, by arbitrary string rewriting regularity is not necessarily preserved. A couple of specific rewrite formats have been proposed in the literature. In [10] it was proved that regularity is preserved by deleting string rewriting, where a string rewriting system is called deleting if there exists a partial ordering on its alphabet such that each letter in the right-hand side of a rule is less than some letter in the corresponding left-hand side. In [13] it was proved that regularity is preserved by so-called period expanding or period reducing string rewriting. When translated to the setting of [21], as also touched upon in Section 3, our present notion of guided insertions and deletions allows for simultaneous insertion and deletion of the dummy symbol. A phenomenon also supported by biological findings. Remarkably, the more liberal guided insertion/deletion approach preserves regularity, whereas in the more restricted mechanism of [21], not mixing insertions and deletions per rewrite step, regularity is not preserved.

Another crucial difference with the mechanism of [21] is the following: for that format it was shown that strings u, v of length n exist satisfying $u \Rightarrow^* v$, but the length of such a reduction is at least exponential in n. In our present format this is not the case: we expect that our slice characterization of guided rewriting serves to prove that if $u \Rightarrow^* v$ then there is always a corresponding reduction of length linear in n. Details have not yet been worked out.

As mentioned in the introduction, the computational power of a variant of insertion-deletion systems was studied in [20]. There deletion means that a string $u\alpha v$ is replaced by uv for a predefined finite set of triples u, α, v, while by insertion a string uv is replaced by $u\alpha v$ for another predefined finite set of triples u, α, v. This notion of insertion-deletion is quite different from ours, and seems less related to biological RNA editing. In the same vein are the guided insertion/deletion systems of [4]. There a hierarchy of classes of insertion/deletion systems and related closure properties are studied. Additionally, a non-mixing insertion/deletion system that models part of the RNA-editing for kinetoplastids is given. A rather different application of term rewriting in the setting of RNA is reported in [8], where the rewrite engine of Maude is exploited to predict the occurrence of specific patterns in the spatial formation of RNA, with competitive precision compared to techniques that are more frequently used in bioinformatics.

Possible future work includes the investigation of preservation of context-freedom and of lifting the bound on the number of consecutive 0's in Theorem 1. More specifically, for a context-free language L, does it hold, for a finite set of guides G, that L_G is context-free too? Considering the set of guides, a generalization to regular sets G is worthwhile studying. Note that the counter-example given in Section 4 involves a non-regular set of guides. So, if L is regular and G is regular, do we have that L_G is regular? Similarly for L context-free. H.J. Hoogeboom suggested to us [11] to consider cones of languages in the sense of Nivat [16], exploiting the closedness under finite state transductions. Shortly before the submission of the final version of this paper, along these lines a partial result restricting to guided rewriting only has been established by J. van Engelen [7]. Generalizing guided insertion/deletion, we also plan to consider guided rewriting based on other types of adjustment relations. In particular, rather than comparing strings symbol-by-symbol, one can consider two strings compatible if they map to the same string for a chosen string homomorphism. A prime example would be the erasing of the dummy 0 in the context of Section 3 for which we conjecture a variant of Theorem 2 to hold.

Acknowledgment We acknowledge fruitful feedback from Peter van der Gulik and detailed comment from the reviewers.

References

[1] J.D. Alfonzo, O. Thiemann, and L. Simpson. The mechanism of U insertion/deletion RNA editing in kinetoplastid mitochondria. *Nucleic Acids Research*, 25(19):3751–3759, 1997.

[2] G.J. Arts and R. Benne. Mechanism and evolution of RNA editing in kinetoplastida. *Biochimica et Biophysica Acta*, 1307(1):39–54, 1996.

[3] R. Benne, J. van de Burg, J. Brakenhoff, P. Sloof, J. van Boom, and M. Tromp. Major transcript of the frameshifted coxii gene from *Trypanosome Mitochondria* contains four nucleotides that are not encoded in the DNA. *Cell*, 46:819–826, 1986.

[4] F. Biegler, M.J. Burrell, and M. Daley. Regulated RNA rewriting: Modelling RNA editing with guided insertion. *Theoretical Computer Science*, 387(2):103–112, 2007.

[5] S. Binder and A. Brennicke. Gene expression in plant mitochondria: Transcriptional and post-transcriptional control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 358(1429):181–188, 2003.

[6] B. Blum, N. Bakalara, and L. Simpson. A model for RNA editing in kinetoplastid mitochondria: RNA molecules transcribed from maxicircle DNA provide the edited information. *Cell*, 60:189–198, 1990.

[7] J. van Engelen. Guided rewriting in families of languages. Technical Report 2012–2013–12, LIACS, July 2013. 16pp.

[8] X.Z. Fu, H. Wang, W. Harrison, and R. Harrison. RNA pseudoknot prediction using term rewriting. In *Proc. BIBE'05, Minneapolis*, pages 169–176. IEEE Computer Society, 2005.

[9] T. Head. Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, 49(6):737–759, 1987.

[10] D. Hofbauer and J. Waldmann. Deleting string rewriting systems preserve regularity. *Theoretical Computer Science*, 327:301–317, 2004.

[11] H.J. Hoogeboom. Private communication, 2013.

[12] K. Cullik II and T. Harju. Splicing semigroups and dominoes and DNA. *Discrete Applied Mathematics*, 31(3):261–271, 1991.

[13] P. Leupold. On regularity-preservation by string-rewriting systems. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *Proc. LATA 2008*, pages 345–356. LNCS 5196, 2008.

[14] M. Margenstern, G. Paun, Y. Rogozhin, and S. Verlan. Context-free insertion-deletion systems. *Theoretical Computer Science*, 330:339–348, 2005.

[15] D. Pixton. Regularity of splicing languages. *Discrete Applied Mathematics*, 69(1–2):69–124, 1996.

[16] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages: Word, Language, Grammar*. Springer, 1997.

[17] J. Skarda, N. Amariglio, and G. Rechavi. RNA editing in human cancer: Review. *ARMIS*, 118(4):551–557, 2010.

[18] H. van der Spek, G.J. Arts, R.R. Zwaal, J. van den Burg, P. Sloof, and R. Benne. Conserved genes encode guide RNAs in mitochondria of *Crithidia fasciculata*. *EMBO*, 10(5):1217–1224, 1991.

[19] K. Stuart, T.E. Allen, S. Heidmann, and S.D. Seiwert. RNA editing in kinteoplastid protozoa. *Microbiology and Molecular Biology Reviews*, 61(1):105–120, 1997.

[20] A. Takahara and T. Yokomori. On the computational power of insertion-deletion systems. *Natural Computing*, 2(4):321–336, 2003.

[21] H. Zantema. Complexity of guided insertion-deletion in RNA-editing. In A.-H. Dediu, H. Fernau, and C. Martín-Vide, editors, *Proc. LATA 2010*, pages 608–619. LNCS 6031, 2010.