# Effective Lossy Queue Languages

Parosh Aziz Abdulla[1], Luc Boasson[2], and Ahmed Bouajjani[2]

[1] DoCS, Uppsala University, P.O. Box 325, S-75105 Uppsala, Sweden.
parosh@docs.uu.se

[2] Liafa, Univ. of Paris 7, Case 7014, 2 place Jussieu, F-75251 Paris Cedex 05, France.
{lub,abou}@liafa.jussieu.fr

**Abstract.** Although the set of reachable states of a lossy channel system (LCS) is regular, it is well-known that this set cannot be constructed effectively. In this paper, we characterize significant classes of LCS for which the set of reachable states can be computed. Furthermore, we show that, for slight generatlizations of these classes, computability can no longer be achieved.

To carry out our study, we define *rewriting systems* which capture the behaviour of LCS, in the sense that (i) they have a FIFO-like semantics and (ii) their languages are downward closed with respect to the substring relation. The main result of the paper shows that, for context-free rewriting systems, the corresponding language can be computed. An interesting consequence of our results is that we get a characterization of classes of meta-transitions whose post-images can be effectively constructed. These meta-transitions consist of sets of nested loops in the control graph of the system, in contrast to previous works on meta-transitions in which only single loops are considered.

Essentially the same proof technique we use to show the result mentioned above allows also to establish a result in the theory of $0L$-systems, i.e., context-free parallel rewriting systems. We prove that the downward closure of the language generated by any $0L$-system is effectively regular.

## 1 Introduction

We consider the problem of model checking of *lossy channel systems (LCS)* which consist of finite-state processes communicating over FIFO-buffers. The buffers are unbounded and *lossy* in the sense that they can lose messages. Such systems can be used to model the behaviour of communication protocols such as link protocols [AJ96] and bounded retransmission protocols [GvdP93,AAB99] which are designed to operate correctly even in the case where the underlying communication medium cannot provide reliable communication. In [AJ96], an algorithm for checking safety properties for LCS is described. The algorithm performs a fixed point iteration where each iteration computes the pre-image of a set of configurations with respect to a single transition of the system. This corresponds to a backward reachability analysis algorithm which constructs a characterization (as a regular set) of the set of configurations from which a set of *final configurations* is reachable.

F. Orejas, P.G. Spirakis, and J. van Leeuwen (Eds.): ICALP 2001, LNCS 2076, pp. 639–651, 2001.

Often, it is also important to be able to perform a forward reachability analysis, and characterize the set of configurations which are reachable from the initial configuration of the system. For instance, several efficient verification methods, such as on-the-fly algorithms [Hol91,CVWY90], are based on forward search of the state space. Also, using forward analysis, we can often automatically generate finite-state abstractions of the system. The abstract model may then be used to check several classes of properties which cannot be analyzed directly in the original model. It is well known that the set of reachable configurations in an LCS can be characterized as a regular set [CFI96,ABJ98]. On the other hand, using undecidability results reported in [May00], it can be shown that this characterization cannot be effectively constructed. Therefore, it is interesting to characterize the class of LCS for which the characterization is effective.

To achieve that, we model the behaviour of an LCS as a rewriting system. To reflect the behaviour of the LCS, we provide a FIFO-like semantics for the rewriting system, i.e., when a rule is applied to a string, the left hand side is removed from the head of the string, while the right hand side is appended to the end of the string. Furthermore, corresponding to lossiness, we require that the language generated is downward closed with respect to the substring relation. Intuitively, the symbols in the left hand side of a rule correspond to the messages received from the channels, while the symbols in the right hand side correspond to messages sent to the channels. This implies that characterizing the class of rewriting systems with effectively computable languages gives also a characterization of the class of LCS with computable reachability sets.

In this paper, we study the limits of the computability of the reachability set, and characterize significant classes of rewriting systems which have effectively computable languages. The main result of the paper is showing that all context-free rewriting systems have effectively computable languages. Furthermore, we show that slight generalizations of the context-free class lead to uncomputability of the language. For instance, it is sufficient to allow two symbols in the left hand sides of the rewriting rules in order to lose computability. Moreover, if we consider rewriting systems on vectors of words (representing FIFO-channel systems with several queues), then computability is lost starting from dimension 2. Nevertheless, we show that for particular $n$-dim context-free rewriting systems called *rings*, the generated language is effectively computable. The class of rings corresponds to $n$-dim systems where each rewriting rule consists in receiving a message from some channel of index say $i$, and sending a string (sequence of messages) to the channel of index $(i + 1) \bmod n$. These rules correspond to actions of communicating processes connected according to a ring topology, where each component has an input and an output buffer.

A significant application of our results is to characterize classes of *meta-transitions* which can be used in forward reachability analysis of LCS. A *meta-transition* is an *arbitrarily long sequence* of transitions, which often corresponds to repeated executions of loops in the syntactic code of the program. The idea is to speed up the iterations corresponding to forward reachability analysis by computing the post-image of a *meta-transition* rather than a single transition.

In almost all practical cases, applying meta-transitions is necessary to achieve termination of the iteration procedure. Obviously, the key issue is to decide which meta-transitions to apply during each iteration step. A necessary criterion is that a meta-transition should allow computability of post-images, i.e., the post-image of each constraint with respect to the meta-transition should be computable. Since the set of transitions in a meta-transition can themselves be viewed as an LCS, our result also gives a characterization of meta-transitions which allow computing of post-images. Observe that the language generated describes the application of an arbitrary sequence of rules, which means that we are able to describe the effect of arbitrary sequences of control loops. In this way we extend significantly the class of LCS on which accelerations can be applied, since earlier works [BG96,BGWW97,BH97,ABJ98] only consider single loops.

It turns out that our proof can also be adapted to establish a result in the language theoretical framework of $0L$-systems [RS80]. These systems consist of context-free sets of rules with a parallel rewriting relation. In fact, there is a tight relation between FIFO rewriting and parallel rewriting (which is actually used in our proof). We show that the set of subwords of the language generated by any $0L$-system is an effective regular set.

**Related Work:** Several works have studied computing post-images of meta-transitions in the context of communicating finite-state processes [BG96, BGWW97] [BH97,ABJ98]. In contrast to this paper, all these works consider meta-transitions which correspond to *single loops*. This excludes several common structures which occur in protocols such as *nested loops*. An example of a nested loop is two transitions sending two different messages to the channels. For instance, in [ABJ98], we show how to compute post-images of single loops for lossy channel systems. In [BGWW97], a complete characterization is given of the class of simple loops which preserve regularity in the context of systems with *perfect channels*. In [BH97] a constraint system is presented and shown to be closed under the execution of any single loop in a system with perfect channels.

Context-free rewriting systems have been used for description and analysis for several classes of infinite-state systems such as BPA and pushdown processes (e.g., [BBK93,CHS92,GH94,Sti96]). All these works deal with the classical (stack-like) semantics, and hence cannot be applied to analyze systems with FIFO buffers. [Bur98] considers rewriting systems with *perfect* FIFO behaviour. It shows that all interesting verification problems (including reachability) are undecidable for these systems.

$0L$-systems have been studied as an alternative of the classical sequential rewriting systems, following the main lines suggested by the theory of context-free grammars. It has been shown that the properties of the two kinds of formalisms are quite different. In particular, the expressive power of the $0L$-systems is not comparable with the usual Chomsky hierarchy. Our result concerning $0L$-system corresponds to the result established in [Cou91] for the context-free languages.

**Outline:** In Section 2 we define some basic notions and introduce FIFO rewriting systems (cyclic rewriting systems) and $0L$-systems. In Section 3 we prove our main result concerning the effective regularity of the language of downward closed context-free systems. We give also in this section our result on the effectiveness of the downward closure of $0L$ languages. In Section 4 we give unconstructibility results for downward closed rewriting systems with unrestricted (non context-free) rules, as well as for 2-dim context-free systems. Finally, in Section 5, we address the effective regularity of the $n$-dim context-free rings languages. For lack of space, some proofs are omitted in this version of the paper.

## 2 Preliminaries

In this section, we give some basic definitions and introduce the FIFO rewriting systems which we call *cyclic* rewriting systems.

### 2.1 Words and Languages

Let $\Sigma$ be a finite alphabet. We denote by $\Sigma^*$ the set of *words* over $\Sigma$, i.e., sequences (or strings) of symbols in $\Sigma$. We denote by $\epsilon$ the empty word.

We denote by $\preceq \subseteq \Sigma^* \times \Sigma^*$ the subword relation, i.e., $a_1 \dots a_n \preceq b_1 \dots b_m$ if there exists $i_1, \dots, i_n \in \{1, \dots, m\}$ such that $i_1 < \dots < i_n$ and $\forall j \in \{1, \dots, n\}.\ a_j = b_{i_j}$. We consider the product generalization of this relation to vectors of words.

An $n$-dim language over $\Sigma$, for $n \geq 1$, is any subset of $(\Sigma^*)^n$. 1-dim languages are called languages as usual. An $n$-dim language $L$ is *downward closed* if $\forall \boldsymbol{u}, \boldsymbol{v} \in (\Sigma^*)^n$, if $\boldsymbol{v} \in L$ and $\boldsymbol{u} \preceq \boldsymbol{v}$, then $\boldsymbol{u} \in L$. Let $L{\downarrow}$ denote the *downward closure* of $L$, i.e., the smallest downward closed set which includes $L$. Notice that a set $L$ is downward closed if and only if $L = L{\downarrow}$.

### 2.2 Simple Regular Expressions

Let us call *atomic expression* any expression $e$ of the form $(a + \epsilon)$ where $a \in \Sigma$, or of the form $A^*$ where $A \subseteq \Sigma$. A *product* $p$ is either the empty word $\epsilon$, or a finite sequence of the form $e_1 \cdots e_m$ of atomic expressions. A *simple regular expression* (SRE) is either $\emptyset$, or a finite union $p_1 + \cdots + p_n$ where each $p_i$ is a product.

**Theorem 1 ([ABJ98]).** *SRE definable sets are precisely downward closed sets. Moreover, for every effective regular language $L$, $L{\downarrow}$ is an effective SRE language.*

**Proposition 1 ([Cou91]).** *For every context-free language $L$ (effectively described by a context-free grammar), the set $L{\downarrow}$ is an effective regular set.*

### 2.3 Cyclic Rewriting Systems

For $n \geq 1$, an $n$-dim *rewriting rule* $r$ over the alphabet $\Sigma$ is a pair $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ where $\boldsymbol{x}, \boldsymbol{y} \in (\Sigma^*)^n$. We denote such a rule by $r : \ \boldsymbol{x} \mapsto \boldsymbol{y}$. The *left hand side* (resp. *right hand side*) of $r$, denoted by $lhs(r)$ (resp. $rhs(r)$), is the vector $\boldsymbol{x}$ (resp. $\boldsymbol{y}$).

An $n$-dim rewriting rule $r$ is *context-free* if $lhs(r) \in (\Sigma \cup \{\epsilon\})^n$. A set of rules $R$ is context-free if all its rules are context-free.

An $n$-dim *cyclic rewriting system* is a pair $(R, R_c)$ where $R$ is a finite set of $n$-dim rewriting rules, and $R_c : (\Sigma^*)^n \to 2^{(\Sigma^*)^n}$ is the *cyclic rewriting mapping* over $R$ defined as follows: for every $\boldsymbol{u} = (u_1, \dots, u_n), \boldsymbol{v} = (v_1, \dots, v_n) \in (\Sigma^*)^n$, we have $\boldsymbol{v} \in R_c(\boldsymbol{u})$ if and only if there exists a rule $r : (x_1, \dots, x_n) \mapsto (y_1, \dots, y_n) \in R$ such that $\forall i \in \{1, \dots, n\}.\ \exists w_i \in \Sigma^*.\ u_i = x_i w_i$ and $v_i = w_i y_i$.

A *weak cyclic rewriting system* is a pair $(R, R_{wc})$ where $R$ is a finite set of $n$-dim rules and $R_{wc}$ is the *weak cyclic rewriting mapping* over $R$ defined as follows: for every $\boldsymbol{u}, \boldsymbol{v} \in (\Sigma^*)^n$, we have $\boldsymbol{v} \in R_{wc}(\boldsymbol{u})$ if and only if there exist $\boldsymbol{u}', \boldsymbol{v}' \in (\Sigma^*)^n$ such that $\boldsymbol{u}' \preceq \boldsymbol{u}$, $\boldsymbol{v}' \in R(\boldsymbol{u}')$, and $\boldsymbol{v} \preceq \boldsymbol{v}'$.

The definition of the mappings $R_c$ and $R_{wc}$ are extended straightforwardly to sets of vectors of words. We denote by $R_c^*$ and $R_{wc}^*$ the reflexive-transitive closure of $R_c$ and $R_{wc}$ respectively. Given a language $L$, $R_c^*(L)$ (resp. $R_{wc}^*(L)$ is called its *cyclic closure* (resp. *weak cyclic closure*) by $R$.

Cyclic rewriting systems can be used to model processes communicating through unbounded FIFO queues (channels). Intuitively, a rule $r : (x_1, \dots, x_n) \mapsto (y_1, \dots, y_n)$ corresponds to an action which, for a queue of index $i$, receives the sequence $x_i$ from the queue, and then sends the sequence $y_i$ to the queue. Weak cyclic rewriting systems model processes communicating through unbounded *lossy* channels (which may lose messages at any time).

Cyclic rewriting systems correspond to Post tag systems. It is well known that they are Turing machine powerful, and hence, the set $R_c^*(L)$ is in general not constructible, for any language $L$. On the other hand, we know by Theorem 1 that for every $n$-dim language $L$ and every set of rewriting rules $R$, the set $R_{wc}^*(L)$ is SRE definable. However, this set is not necessarily effectively constructible.

## 2.4 Substitutions and 0*L*-Systems

A *substitution* is a mapping $\phi : \Sigma \to 2^{\Sigma^*}$ which associates with each symbol $a \in \Sigma$ a language over $\Sigma$ (which may be empty).

A substitution is extended to words by taking $\phi(a_1 \cdots a_n) = \phi(a_1) \cdots \phi(a_n)$ and $\phi(\epsilon) = \epsilon$, and then generalized straightforwardly to languages. We denote by $\phi^i$ the substitution obtained by $i$ compositions of $\phi$. Then, let $\phi^{\geq n}(a)$ denote the set $\bigcup_{i \geq n} \phi^i(a)$, for any $n \geq 0$ and $a \in \Sigma$. We denote by $\phi^*$ (resp. $\phi^+$) the reflexive-transitive (resp. transitive) closure of $\phi$, i.e., $\phi^{\geq 0}(a)$ (resp. $\phi^{\geq 1}(a)$).

A substitution $\phi$ is *finite* if, for every $a \in \Sigma$, $\phi(a)$ is a finite language. A substitution is *downward closed* if it associates with each symbol a downward closed language, and it is *complete* if, for every $a \in \Sigma$, $\phi(a)$ is nonempty. A *weak substitution* is a downward closed complete substitution.

Given a set of 1-dim context-free rewriting rules $R$, we define a substitution $\varphi_R$ such that, for every $a \in \Sigma$, $\varphi_R(a) = \{rhs(r) : r \in R \text{ and } lhs(r) = a\}$. We define also a substitution $\phi_R$ such that, for every $a \in \Sigma$, $\phi_R(a) = \{\epsilon\} \cup \varphi_R(a){\downarrow}$. Notice that, since $R$ is finite, both $\varphi_R$ and $\phi_R$ are finite substitutions. Moreover, $\phi_R$ is a weak substitution whereas $\varphi_R$ is in general not necessarily complete and not downward closed. Since $\varphi_R$ is not complete, $\varphi_R(a)\downarrow$ is in general different from $\phi_R(a)$ (since $\emptyset\downarrow = \emptyset$).

A $0L$*-system* (resp. *weak* $0L$*-system*) is a pair $(R, \varphi_R)$ (resp. $(R, \phi_R)$) where $R$ is a finite set of context-free rewriting rules.

The following lemma shows the link between cyclic context-free rewriting and substitution (parallel rewriting).

**Lemma 1.** *For every finite set of 1-dim context-free rules $R$, and every language $L$, if $\phi_R^*(L)$ is an effective SRE set, then $R_{wc}^*(L)$ is also an effective SRE set.*

## 3 Constructibility for Context-Free Systems

We give in this section our main result (Theorem 2) and present its proof. This result says that the language generated by a weak cyclic context-free rewriting system is an effective regular (actually SRE) set.

An adaptation of the proof for this theorem can be done to establish a result concerning the languages generated by $0L$-systems (Theorem 3). The result says that the set of subwords of such a language is effectively regular.

**Theorem 2.** *For every context-free language $L$ (effectively described by a context-free grammar), and for every finite set of (1-dim) context-free rewriting rules $R$, the set $R_{wc}^*(L)$ is effectively SRE representable.*

*Proof.* Clearly, $R_{wc}^*(L) = R_{wc}^*(L\downarrow)$ for every language $L$. Hence, by Proposition 1 and Theorem 1, it is sufficient to prove that $R_{wc}^*(L)$ is an effective SRE set for every SRE language $L$. Moreover, by Lemma 1, this can be done by proving that $\phi_R^*(L)$ is an effective SRE set. For that, we need some definitions.

Given a finite weak substitution $\phi$ (effectively given), we define $\Sigma_\phi(a) = \phi^*(a) \cap \Sigma$. Intuitively, $\Sigma_\phi(a)$ is the set of accessible symbols from $a$ by iterated substitutions. Notice that the set $\Sigma_\phi(a)$ contains the symbol $a$ itself. Notice also that this set can be computed since the alphabet $\Sigma$ is finite.

We partition the set of symbols $\Sigma$ according to the following equivalence relation: two symbols $a$ and $b$ are equivalent if $\Sigma_\phi(a) = \Sigma_\phi(b)$. The equivalence class of a symbol $a$ is denoted $[a]$.

The accessibility relation induces an ordering between equivalences classes: $[a] \leq [b]$ if and only if $\Sigma_\phi(a) \subseteq \Sigma_\phi(b)$. Let *level*$(a)$ be the level of $[a]$ according to this ordering: minimal elements (equivalence classes) have level 0, and level $k+1$ corresponds to minimal elements of the set of classes without the elements of level less than $k$.

Given a finite weak substitution $\phi$, a symbol $a$ is *recursive* if $a \in \phi^+(a)$. It is *expansive* if $aa \in \phi^+(a)$. A symbol is *linear* if it is recursive and not expansive. Let us denote by $Rec_\phi$ (resp. $Exp_\phi$, $Lin_\phi$, $NRec_\phi$) the set of recursive (resp. expansive, linear, nonrecursive) symbols according to $\phi$. Deciding whether a symbol is recursive, linear, or expansive is straightforward. Notice that for every $a, b \in \Sigma$ such that $[a] = [b]$, we have $a \in Rec_\phi$ (resp. $Exp_\phi$, $Lin_\phi$, $NRec_\phi$) iff $b \in Rec_\phi$ (resp. $Exp_\phi$, $Lin_\phi$, $NRec_\phi$).

A substitution $\phi$ is *normal* if, for every $a \in Exp_\phi$ (resp. $Lin_\phi$), the set $\phi(a)$ contains $aa$ (resp. $a$).

Let $R$ be a finite set of 1-dim context-free rewriting rules. We suppose hereafter that $R$ does not contain rules of the form $\epsilon \mapsto y$. The consideration of this kind of rules is not difficult.

For every $a \in Exp_{\phi_R}$ (resp. $Lin_{\phi_R}$), let $\pi(a)$ be the smallest strictly positive integer such that $aa$ (resp. $a$) is in $\phi_R^{\pi(a)}(a)$. Then, let $\pi = \text{lcm}\{\pi(a) \ : \ a \in Rec_{\phi_R}\}$, and let $\psi_R = \phi_R^\pi$. It is easy to see that $\psi_R$ is a normal finite weak substitution. Since for every symbol $a \in \Sigma$, we have $\phi_R^*(L) = \bigcup_{i=1}^{\pi-1} \phi_R^i(\psi_R^*(L))$, we deduce that:

**Lemma 2.** *For every finite set of 1-dim context-free rewriting rules $R$, and every language $L$, if the set $\psi_R^*(L)$ is effective, then the set $\phi_R^*(L)$ is effective.*

By Lemma 2, and since $\psi_R$ is a normal finite weak substitution, in order to compute the image by $\phi_R^*$ of a language $L$, it is sufficient to know how to compute the image of $L$ for every given normal finite weak substitution. The remainder of the proof consists in showing that this is possible. For the sake of simplicity, we show this fact for $L$ reduced to a symbol. The generalization to any regular language is not difficult. Therefore, we prove the following key proposition.

**Proposition 2.** *For every normal finite weak substitution $\phi$, and every symbol $a \in \Sigma$, the set $\phi^*(a)$ is effectively SRE representable.*

*Proof.* To present the proof, we need to introduce some notions and to establish several lemmas. Let $\phi$ be a substitution. Then, for any positive integer $B$, we say that a symbol $a$ is $B$-regular if $\phi^{B+n}(a) \subseteq \phi^{B+n+1}(a)$ for every $n \geq 0$. This means that, after the $B$ first applications of $\phi$, the iteration of $\phi$ yields a non-decreasing sequence of sets (i.e., each application of $\phi$ augments or at least preserves the given set).

*Remark 1.* By definition, if a symbol is $B$-regular, then it is necessarily $B'$-regular for every $B' \geq B$.

Let us consider some simple facts. The first one is that, since $a \in \phi(a)$, all recursive symbols (hence, all linear and expansive symbols) are 0-regular.

**Lemma 3.** *Let $\phi$ be normal finite weak substitution. Then, for every $a \in Rec_\phi$, $a$ is 0-regular.*

Next lemma says that $\phi^*(a)$ is effective when $a \in Exp_\phi$.

**Lemma 4.** *Let $\phi$ be a normal finite weak substitution. Then, for every $a \in Exp_\phi$, $\phi^*(a) = (\Sigma_\phi(a))^*$.*

Now, we give the main lemma which allows to prove Proposition 2. The proof of this lemma itself involves several lemmas (Lemmas 6, 7, 8, 9, 10, 11, and 12) which allow to establish that $\phi^*(a)$ is effective when $a \in Lin_\phi$.

**Lemma 5.** *Let $\phi$ be a normal finite weak substitution. Then, for every $a \in \Sigma$, there exists a $B \geq 0$ such that $a$ is $B$-regular and $\phi^*(a)$ is effectively regular.*

*Proof.* The proof is by induction on the level of the symbols.

**Basis:** Let $a \in \Sigma$ such that $level(a) = 0$. If $a$ is nonrecursive, then $\phi(a) = \{\epsilon\}$, and hence, $a$ is 1-regular, and $\phi^*(a) = \{\epsilon\}$ is effectively regular. If $a$ is expansive, then by Lemma 3 and Lemma 4, we know that $a$ is 0-regular and $\phi^*(a)$ is effectively regular. If $a$ is linear, then by Lemma 3 $a$ is 0-regular. Moreover, since $level(a) = 0$, we have $\phi(a) = \Sigma_\phi(a)$, and hence $\phi^*(a) = [a]$ is effectively regular.

**Inductive step:** Let us now consider a symbol $a$ such that $level(a) = k + 1$ for some $k \geq 0$, and assume, by induction hypothesis and Remark 1, that there exists $B \geq 0$ such that every symbol $b$ of level at most $k$ is $B$-regular, and $\phi^*(b)$ is effectively regular. We show that the symbol $a$ is $(B + 1)$-regular and that $\phi^*(a)$ is effectively regular.

**Case 1:** $a$ **is nonrecursive**. Then, for every $b \in \Sigma_\phi(a)$, $level(b) < level(a)$. By induction hypothesis, for every $b \in \Sigma_\phi(a)$, $b$ is $B$-regular and $\phi^*(b)$ is effective. It easy to see that $a$ is $(B + 1)$-regular and $\phi^*(a)$ is effectively regular.

**Case 2:** $a$ **is expansive**. By Lemma 3, we know that $a$ is 0-regular, and hence, $a$ is $(B + 1)$-regular (for any $B \geq 0$). Moreover, by Lemma 4, we know that $\phi^*(a)$ is effectively regular.

**Case 3:** $a$ **is linear**. We know by Lemma 3 that $a$ is 0-regular, and hence that it is $(B + 1)$-regular. It remains to show that $\phi^*(a)$ is effectively regular.

Let us consider the context-free grammar $G_a = (N_a, T_a, S_a, \rho_a)$ such that:

- the set of nonterminal symbols is $N_a = [a]$,
- the set of terminal symbols is $T_a = \Sigma_\phi(a) \setminus [a]$,
- the start symbol is $S_a = a$,
- the set of rules is $\rho_a = \{b \to w \;:\; b \in [a] \text{ and } w \in \phi(b)\}$.

Notice that, since $a$ is linear, the rules of $G_a$ are either *nonterminal rules* of the form $b \to ub'v$, where $b' \in [a]$ and $u, v \in T_a^*$, or *terminal rules* of the form $b \to w$ where $w \in T_a^*$. Since $\phi$ is normal, we have by definition rules of the form $b \to b$ for any $b \in [a]$.

We index the nonterminal rule from 1 to $n_1$ and the terminal rules from 1 to $n_2$, where $n_1$ (resp. $n_2$) is the number of nonterminal (resp. terminal) rules in $G_a$. Then, we derive from $G_a$ another context-free grammar $\widehat{G_a} = (N_a, \widehat{T_a}, S_a, \widehat{\rho_a})$ where:

- $\widehat{T_a} = \{\langle u\rangle_i, \langle v\rangle_i \;:\; \exists r = b \to ub'v \in \rho_a.\ index(r) = i\} \cup \{\langle\langle w\rangle\rangle_i \;:\; \exists r = b \to w \in \rho_a.\ w \in T_a^* \text{ and } index(r) = i\}$.
- $\widehat{\rho_a} = \{b \to \langle u\rangle_i b' \langle v\rangle_i \;:\; \exists r = b \to ub'v \in \rho.\ index(r) = i\} \cup \{b \to \langle\langle w\rangle\rangle_i \;:\; \exists r = b \to w \in \rho.\ w \in T_a^* \text{ and } index(r) = i\}$.

The following lemmas show the links between derivation in the grammar $\widehat{G_a}$ and the iterations of the substitution $\phi$.

**Lemma 6.** *If there is a derivation in $\widehat{G_a}$ of length $\ell$ from $a$ to some nonterminal word, then this word is of the form $\langle u_\ell\rangle_{i_\ell} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1} b\langle v_1\rangle_{i_1}\langle v_2\rangle_{i_2} \cdots \langle v_\ell\rangle_{i_\ell}$, and*

$$\phi^{\ell-1}(u_\ell)\cdots\phi(u_2)u_1bv_1\phi(v_2)\cdots\phi^{\ell-1}(v_\ell) \subseteq \phi^\ell(a).$$

**Lemma 7.** *For every word $ubv \in \phi^\ell(a)$ where $b \in [a]$, there exists a derivation in $\widehat{G_a}$ of length $\ell$ from $a$ to some nonterminal word of the form*

$$\langle u_\ell\rangle_{i_\ell} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1} b\langle v_1\rangle_{i_1}\langle v_2\rangle_{i_2} \cdots \langle v_\ell\rangle_{i_\ell}$$

*such that $u \in \phi^{\ell-1}(u_\ell)\cdots\phi(u_2)u_1$, and $v \in v_1\phi(v_2)\cdots\phi^{\ell-1}(v_\ell)$.*

**Lemma 8.** *If there is a derivation in $\widehat{G_a}$ of length $\ell$ from $a$ to some terminal word, then this word is of the form $\langle u_\ell\rangle_{i_\ell} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1}\langle\langle w\rangle\rangle_i\langle v_1\rangle_{i_1}$ $\langle v_2\rangle_{i_2} \cdots \langle v_\ell\rangle_{i_\ell}$, and*

$$\phi^\ell(u_\ell)\cdots\phi^2(u_2)\phi(u_1)w\phi(v_1)\phi^2(v_2)\cdots\phi^\ell(v_\ell) \subseteq \phi^\ell(a).$$

The proofs of the three lemmas above are by straightforward inductions.

**Lemma 9.** *For every terminal word $\sigma$ in $\phi^\ell(a)$ (i.e., $\sigma \in T_a^* \cap \phi^\ell(a)$), there exists an integer $m \leq \ell$ and a derivation in $\widehat{G_a}$ of length $m$ from $a$ to some terminal word $\mu = \langle u_m\rangle_{i_m} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1}\langle\langle w\rangle\rangle_i\langle v_1\rangle_{i_1}\langle v_2\rangle_{i_2} \cdots \langle v_m\rangle_{i_m}$ such that*

$$\sigma \in \phi^{\ell-m}[\phi^m(u_m)\cdots\phi^2(u_2)\phi(u_1)w\phi(v_1)\phi^2(v_2)\cdots\phi^m(v_m)].$$

*Proof.* The fact that $\sigma \in \phi^\ell(a)$ implies that there are words $\sigma_0, \sigma_1, \ldots, \sigma_\ell$ such that $\sigma_0 = a$, $\sigma = \sigma_\ell$, and $\sigma_{i+1} \in \phi(\sigma_i)$ for every $i \in \{0, \ldots, \ell-1\}$. Let $\sigma_i$ be the last word in this sequence which contains a symbol $b \in [a]$. This means that after $i$ applications of $\phi$, we get a terminal word $\sigma_{i+1}$. Let $m = i+1$ and $\mu = \sigma_m$. By Lemma 7, we know that there exists a derivation in $\widehat{G_a}$ of length $i = m-1$ from $a$ to some nonterminal word $\mu' = \langle u_\ell\rangle_{i_\ell} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1} b\langle v_1\rangle_{i_1}\langle v_2\rangle_{i_2} \cdots \langle v_\ell\rangle_{i_\ell}$ such that $\mu \in \phi^{m-1}(u_m)\cdots\phi(u_2)u_1bv_1\phi(v_2)\cdots\phi^{m-1}(v_m)$. The result follows from the fact that $\mu$ can be derived in $\widehat{G_a}$ from $\mu'$ using a terminal rule, and from the fact that $\sigma \in \phi^{\ell-m}(\mu)$. □

Now, the following lemmas show the key property which allows to compute $\phi^*(a)$ by iterating the application of $\phi$ only a finite number of times, thanks to $B$-regularity.

**Lemma 10.** *If there is a derivation in $\widehat{G_a}$ of length $\ell$ from $a$ to some nonterminal word $\langle u_\ell\rangle_{i_\ell} \cdots \langle u_2\rangle_{i_2}\langle u_1\rangle_{i_1} b\langle v_1\rangle_{i_1}\langle v_2\rangle_{i_2} \cdots \langle v_\ell\rangle_{i_\ell}$, then*

$$\begin{aligned}\phi^{\geq B}(u_\ell)\cdots\phi^{\geq B}(u_{B+1})\phi^{B-1}(u_B)\cdots\phi(u_2)u_1b&\\ v_1\phi(v_2)\cdots\phi^{B-1}(v_B)\phi^{\geq B}(v_{B+1})\cdots\phi^{\geq B}(v_\ell) &\subseteq \phi^*(a)\end{aligned}$$

*Proof.* Suppose that there is a derivation $\delta$ in $\widehat{G_a}$ from $a$ to the nonterminal word $\alpha = \langle u_\ell \rangle_{i_\ell} \cdots \langle u_2 \rangle_{i_2} \langle u_1 \rangle_{i_1} b \langle v_1 \rangle_{i_1} \langle v_2 \rangle_{i_2} \cdots \langle v_\ell \rangle_{i_\ell}$, and consider a word $\sigma$ in

$$\phi^{\geq B}(u_\ell) \cdots \phi^{\geq B}(u_{B+1}) \phi^{B-1}(u_B) \cdots \phi(u_2) u_1 b \\ v_1 \phi(v_2) \cdots \phi^{B-1}(v_B) \phi^{\geq B}(v_{B+1}) \cdots \phi^{\geq B}(v_\ell). \quad (1)$$

Hence, $\sigma$ can be written $\mu_\ell \cdots \mu_{B+1} \mu_B \cdots \mu_2 \mu_1 b \nu_1 \nu_2 \cdots \nu_B \nu_{B+1} \cdots \nu_\ell$ where $\mu_i \in \phi^{i-1}(u_i)$ and $\nu_i \in \phi^{i-1}(v_i)$ for every $i \in \{1, \ldots, B\}$, and for every $i \in \{B+1, \ldots, \ell\}$, there is $m_i, n_i \geq B$ such that $\mu_i \in \phi^{m_i}(u_i)$ and $\nu_i \in \phi^{n_i}(v_i)$. Let $\kappa = \max\{n_i, m_i \ : \ B+1 \leq i \leq \ell\}$. Then, using the property of $B$-regularity (see Remark 1), we deduce from (1) that $\sigma$ is in the set:

$$\phi^{\kappa+\ell-(B+1)}(u_\ell) \cdots \phi^{\kappa}(u_{B+1}) \phi^{B-1}(u_B) \cdots \phi(u_2) u_1 b \\ v_1 \phi(v_2) \cdots \phi^{B-1}(v_B) \phi^{\kappa}(v_{B+1}) \cdots \phi^{\kappa+\ell-(B+1)}(v_\ell). \quad (2)$$

Now, consider the derivation $\delta$ in $\widehat{G_a}$ leading to the word $\alpha$ defined in the beginning of the proof. This derivation can be decomposed as $a \stackrel{*}{\Longrightarrow} \beta \stackrel{*}{\Longrightarrow} \alpha$, where $\beta = \langle u_\ell \rangle_{i_\ell} \cdots \langle u_{B+1} \rangle_{i_{B+1}} b' \langle v_{B+1} \rangle_{i_{B+1}} \cdots \langle v_\ell \rangle_{i_\ell}$ where $b' \in [a]$. This means that we have a sub-derivation $\delta'$ of the form $b' \stackrel{*}{\Longrightarrow} \langle u_B \rangle_{i_B} \cdots \langle u_1 \rangle_{i_1} b \langle v_1 \rangle_{i_1} \cdots \langle v_B \rangle_{i_B}$. We define another derivation in $\widehat{G_a}$ which produces first the word $\beta$ as in $\delta$, then iterates $\kappa - B$ times the rule $b' \rightarrow \langle \epsilon \rangle b' \langle \epsilon \rangle$ (recall that $b' \rightarrow b'$ must exist in $G_a$ since the original substitution $\phi$ is normal), and finally applies the rules of $\delta'$. This derivation produces a nonterminal word $\alpha'$ of the form

$$\langle u_\ell \rangle_{i_{\kappa+\ell-B}} \cdots \langle u_{B+1} \rangle_{i_{\kappa+1}} \langle \epsilon \rangle^{\kappa-B} \langle u_B \rangle_{i_B} \cdots \langle u_1 \rangle_{i_1} b \\ \langle v_1 \rangle_{i_1} \cdots \langle v_B \rangle_{i_B} \langle \epsilon \rangle^{\kappa-B} \langle v_{B+1} \rangle_{i_{\kappa+1}} \cdots \langle v_\ell \rangle_{i_{\kappa+\ell-B}}.$$

Then, we deduce from (2) and Lemma 6 that $\sigma \in \phi^*(a)$. This completes the proof of Lemma 10. □

**Lemma 11.** *If there is a derivation in $\widehat{G_a}$ of length $\ell$ from $a$ to some terminal word $\langle u_\ell \rangle_{i_\ell} \cdots \langle u_2 \rangle_{i_2} \langle u_1 \rangle_{i_1} \langle \langle w \rangle \rangle_i \langle v_1 \rangle_{i_1} \langle v_2 \rangle_{i_2} \cdots \langle v_\ell \rangle_{i_\ell}$, then*

$$\phi^{\geq B}(u_\ell) \cdots \phi^{\geq B}(u_B) \phi^{B-1}(u_{B-1}) \cdots \phi(u_1) w \\ \phi(v_1) \cdots \phi^{B-1}(v_{B-1}) \phi^{\geq B}(v_B) \cdots \phi^{\geq B}(v_\ell) \subseteq \phi^*(a).$$

*Proof.* Similar to the proof of Lemma 10. □

We are now able to give a construction for $\phi^*(a)$: Consider the nonterminal words $\langle u_\ell \rangle_{i_\ell} \cdots \langle u_1 \rangle_{i_1} b \langle v_1 \rangle_{i_1} \cdots \langle v_\ell \rangle_{i_\ell}$ generated by the grammar $\widehat{G_a}$. They form an effective linear context-free language. By a regular transduction, we can decorate the terminal symbols in these words by indices in $\{0, 1, ..., B\}$ in the following manner: each nonterminal symbol $\langle u_j \rangle_{i_j}$ or $\langle v_j \rangle_{i_j}$ is decorated by $j - 1$ if $j < B$, otherwise by $B$. This yields another effective context-free linear language. Now, by induction hypothesis, we know that for every symbol $b$ in $T_a$, the set $\phi^*(b)$ is effective, and hence, the set $\phi^{\geq B}(u)$ is effective, for any $u \in T_a^*$. Therefore, we can use another transduction to substitute the finite set $\phi^j(u)$ to each nonterminal symbol $\langle u \rangle_i$ decorated by an index $j < B$, and to substitute the (effective) regular set $\phi^{\geq B}(u)$ to each $\langle u \rangle_i$ decorated by the index $B$. We get in this manner a new linear context-free language, we call $L_1$.

We apply a similar process to terminal words $\langle u_\ell \rangle_{i_\ell} \cdots \langle u_1 \rangle_{i_1} \langle\langle w \rangle\rangle_i \langle v_1 \rangle_{i_1} \cdots \langle v_\ell \rangle_{i_\ell}$ generated by $\widehat{G_a}$. In this case, the central symbol $\langle\langle w \rangle\rangle_i$ is decorated by index 0, and each nonterminal symbol $\langle u_j \rangle_{i_j}$ or $\langle v_j \rangle_{i_j}$ is decorated by $j$ if $j < B$, otherwise by $B$. As in the previous cases, we obtain an effective linear language. Let us call this language $L_2^0$. Then, we repeat this process, but this time, we move up all the indices by one. In this manner, the regular substitution performed after the indexing step replaces $\langle\langle w \rangle\rangle_i$ by $\phi(w)$, all the $\langle u_j \rangle_{i_j}$'s such that $j < B-1$ by $\phi^{j+1}(u_j)$, and all the $\langle u_j \rangle_{i_j}$'s such that $j \geq B-1$ by $\phi^B(u_j)$. Let us call $L_2^1$ the so obtained linear language. We repeat this process $B$ times. The union of all the languages $L_2^j$ is an effective linear language $L_2$.

**Lemma 12.** *The language $L_1 \cup L_2$ is precisely $\phi^*(a)$ and it is effectively regular.*

*Proof.* By Lemma 10 and Lemma 11, we know that $L_1 \cup L_2 \subseteq \phi^*(a)$. Let us consider the reverse inclusion. Let $\sigma$ be a word in $\phi^*(a)$. There are two cases to consider:

First, suppose that $\sigma$ is a nonterminal word (i.e., of the form $ubv$ where $b \in [a]$). Then, by Lemma 7 we know that $\sigma \in L_1$.

Consider now that $\sigma$ is a terminal word (i.e. $\sigma \in T_a^*$). Then, by Lemma 9, there exists an intermediate terminal word $\sigma'$ such that $\sigma' \in L_2^0$ and $\sigma \in \phi^n(\sigma)$ for some $n \geq 0$. It can be seen that, if $n < B$, then $\sigma \in L_2^n$, otherwise $\sigma \in L_2^B$. In both cases, $\sigma \in L_2$. This completes the proof of the fact that $L_1 \cup L_2 = \phi^*(a)$.

As for the effective regularity of $L_1 \cup L_2$, we know that this language is an effective context-free language, but we know also that it must be a downward closed language since $\phi$ is downward closed. Then, the result follows from Proposition 1. $\square$

Lemma 12 completes the case where $a$ is a linear symbol in the inductive step of the proof of Lemma 5. Hence, Lemma 5 is proved. $\square$

Proposition 2 follows from Lemma 5 and Theorem 1. $\square$

Theorem 2 follows from Lemma 1, Lemma 2, and Proposition 2. $\square$

*Remark 2.* By proving the result above (in fact from Lemma 2, and Proposition 2), we have proved that the set of words generated by any weak $0L$-system is an SRE effective set.

We can adapt the proof presented above in order to show that the set of subwords of the language generated by any $0L$-system is an effective SRE set.

**Theorem 3.** *For every context-free language $L$, and every $0L$-system $(R, \varphi_R)$, the set $\varphi_R^*(L){\downarrow}$ is effectively SRE representable.*

## 4 Unconstructibility Results

We show in this section that the weak cyclic closure is unconstructible in general, even though we know that it is always regular. For that, we use known results on *lossy channel systems* (LCS for short). Such a system consists of a finite-state machine operating on a single FIFO buffer which is unbounded and lossy in the sense that it can nondeterministically lose messages (see [AJ96] for a formal definition).

The set of reachable configurations of any LCS from any given configuration is downard closed. Thus, we can deduce from Theorem 1 that this set is SRE definable. However, from [May00] we know the following:

**Lemma 13.** *For any LCS $\mathcal{L}$ and any configuration $\gamma$, the set of reachable configuration of $\mathcal{L}$ starting from $\gamma$ is in general not computable.*

We use this lemma to establish that the weak cyclic closure is not constructible already for 1-dim rewriting systems. For that, given a lossy channel system $\mathcal{L}$, we derive a 1-dim cyclic rewriting system $\mathcal{R}$ which "simulates" $\mathcal{L}$. This simulation can be done as soon as we allow rules with two symbols in their left hand sides.

**Theorem 4.** *There is no algorithm which can construct the set $R^*_{wc}(\epsilon)$ for any given finite set of* 1*-dim rewriting rules $R$.*

The actions of a LCS can be straightforwardly encoded as rules of a 2-dim context-free cyclic rewriting system. Hence, we have the following fact.

**Theorem 5.** *There is no algorithm which can construct the set $R^*_{wc}(\epsilon, \epsilon)$ for any given finite set of* 2*-dim context-free rewriting rules $R$.*

## 5 Constructibility for $n$-Dim Context-Free Rings

We show in this section that the weak cyclic closure is constructible for a special kind of 2-dim context-free rewriting systems.

A set of $n$-dim rewriting rules $R$ is a *ring* if, for every rule $r : (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ in $R$, $\exists i \in \{1, \ldots, n\}$. $\forall j \neq i.\ x_j = \epsilon$ and $\forall j \neq (i+1) \bmod n.\ y_j = \epsilon$. Hence, $r$ is either of the form $(\epsilon, \ldots, \epsilon, x_i, \epsilon, \ldots, \epsilon) \mapsto (\epsilon, \ldots, \epsilon, y_{i+1}, \epsilon, \ldots, \epsilon)$ or of the form $(\epsilon, \ldots, \epsilon, x_n) \mapsto (y_1, \epsilon, \ldots, \epsilon)$. Notice that every set of 1-dim rewriting rules is a 1-dim ring and vice-versa.

We consider hereafter $n$-dim context-free rings, i.e., the $x_i$'s are symbols in $\Sigma$. Intuitively, the rules of these systems correspond to actions of FIFO-channel systems where a symbol $x$ is received from some channel of index $i$, and a word $y$ is sent to the channel of index $(i+1) \bmod n$.

We can show that the computation of the weak cyclic closure for rings can be reduced to the problem of computing the weak cyclic closure of 1-dim context free rewriting systems.

**Theorem 6.** *For every $n$-dim context-free ring $R$, and every $n$-dim language $L$ which is a product of $n$ context-free languages (effectively described by context-free grammars), the set $R^*_{wc}(L)$ is effectively SRE representable.*

## References


[AAB99] Parosh Aziz Abdulla, Aurore Annichini, and Ahmed Bouajjani. Algorithmic verification of lossy channel systems: An application to the bounded retransmission protocol. In *TACAS'99*. LNCS 1579, 1999.

[ABJ98] Parosh Aziz Abdulla, Ahmed Bouajjani, and Bengt Jonsson. On-the-fly analysis of systems with unbounded, lossy fifo channels. In *CAV'98*. LNCS 1427, 1998.

[AJ96] Parosh Aziz Abdulla and Bengt Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[BBK93] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *Journal of the ACM*, (40):653–682, 1993.

[BG96] B. Boigelot and P. Godefroid. Symbolic verification of communication protocols with infinite state spaces using QDDs. In *CAV'96*. LNCS 1102, 1996.

[BGWW97] B. Boigelot, P. Godefroid, B. Willems, and P. Wolper. The power of QDDs. In *SAS'97*. LNCS 1302, 1997.

[BH97] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of fifo-channel systems with nonregular sets of configurations. In *ICALP '97*. LNCS 1256, 1997.

[Bur98] O. Burkart. Queues as processes. *Electrnic Notes in Theoretical Computer Science*, 18, 1998.

[CFI96] Gérard Cécé, Alain Finkel, and S. Purushothaman Iyer. Unreliable Channels Are Easier to Verify Than Perfect Channels. *Information and Computation*, 124(1):20–31, 1996.

[CHS92] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In *CONCUR '92*. LNCS, 1992.

[Cou91] B. Courcelle. On constructing obstruction sets of words. *EATCS*, 44:178–185, June 1991.

[CVWY90] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory efficient algorithms for the verification of temporal properties. In *CAV'90*, 1990.

[GH94] J.F. Groote and H. Hüttel. Undecidable equivelences for basic process algebra. *Information and Computation*, 1994.

[GvdP93] J.F. Groote and J. van de Pol. A bounded retransmission protocol for large data packets. Tech. report, Dept. of Philosophy, Utrecht University, Oct. 1993.

[Hol91] G.J. Holzmann. *Design and Validation of Computer Protocols.* Prentice Hall, 1991.

[May00] R. Mayr. Undecidable problems in unreliable computations. In *LATIN'2000*. LNCS 1776, 2000.

[RS80] G. Rozenberg and A. Salomaa. *The Mathematical Theory of L-systems.* Academic Press, 1980.

[Sti96] C. Stirling. Decidability of bisimulation equivalence for normed push-down processes. In *CONCUR'96*. LNCS 1119, 1996.