Generating Bounded Languages Using Bounded Control Sets

Pierre Ganty\textsuperscript{1} and Radu Iosif\textsuperscript{2}

1 IMDEA Software Institute, Madrid, Spain
2 VERIMAG/CNRS, Grenoble, France

Abstract
We study context-free grammars whose generated language is bounded (that is, subset of some expression $w_1 \ldots w_d$ called bounded expression). We investigate the underlying generating process of such language and show that there exists a bounded expression $u_1^* \ldots u_m^*$ over the production rules, such that the language is generated only by sequences of production rules conforming to the bounded expression. We give an algorithm to compute such a bounded expression, and an optimal upper bound on its running time.

1998 ACM Subject Classification F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases bounded context-free languages, finite index derivations, control sets

1 Introduction

Bounded languages are languages contained within sets of the form $w_1^* \ldots w_d^*$, for some non-empty words $w_1, \ldots, w_d$. Due to their nice tractability \cite{3} and combinatorial properties \cite{5}, they represent an important means to understanding key properties of formal languages.

In particular, bounded context-free languages \cite{5} have recently found applications such as the verification of sequential \cite{4} and concurrent \cite{3} recursive programs, and the study of combinatorial properties of vector addition systems \cite{10, 11}.

We investigate the generating process of bounded context-free languages, that is, the order in which grammar productions fire. We ask the following question: can such a highly structured language be generated by firing productions in a similarly structured way? More precisely, given grammar $G$ with productions $\Delta$, knowing that the language of the grammar is included in the language of a bounded expression $b = w_1^* \ldots w_d^*$, i.e. $L(G) \subseteq b$, is there another bounded expression $\Gamma_b = u_1^* \ldots u_m^*$ over the alphabet of productions $\Delta$, such that only the derivations of $G$ that conform to the bounded expression $u_1^* \ldots u_m^*$ are sufficient to generate each word in $L(G)$?

Besides providing a positive answer, we present an algorithm which computes a bounded expression $u_1^* \ldots u_m^*$, with the above properties. Even though the proof of correctness of the algorithm is intricate and relies on some complex combinatorial properties of derivations, the algorithm itself is easily implementable. Indeed, it only relies on some basic context-free grammar primitives (intersecting with a regular language \cite{7}, testing inclusion in a regular language \cite{1} ) and Dijkstra’s shortest path algorithm.

Finally, we provide an optimal upper bound on the running time of this algorithm, which is exponential in the maximal number $k$ of simultaneous occurrences of nonterminals needed to produce each word in the language (also called the language index) and in the number $d$ of words that occur under the Kleene star in the bounded expression, but polynomial in the size of the grammar $|G|$ and the size of the bounded expression, namely $(\sum_{i=1}^{d} |w_i| \cdot |G|)^{O(k)} + d$.

The main application of our result is interprocedural program analysis. Context-free grammars are used in interprocedural analysis to represent the so-called interprocedurally
Generating Bounded Languages Using Bounded Control Sets

valid control paths of the program. As we argue below, an analysis that considers the structure of the derivation sequence for a given control path, instead of the path itself, is better suited for handling local variables in programs, with a high degree of precision. The result in this paper characterizes the derivation process of bounded context-free languages, providing interprocedural static analyzers a strategy to compute their results, and a tool for evaluating their complexity.

Interprocedural program analyses are usually defined by considering a set of interprocedurally valid paths (IVP) in a control flow graph model of a program. This set of paths (sequences $e_1 \ldots e_n$ of edges representing the program “small” steps) is usually defined as the language of a context-free grammar $G$, which accounts for the correct pairing of the call and return edges of the graph: each return edge corresponds to a preceding call edge, and the pairs of positions of the call/return edges in a path are well-nested. In such framework, each edge $e$ is associated a so-called transfer functions $f_e$. Then transfer functions are extended to IVP, or, equivalently, to words $w = e_1e_2\ldots e_k$ of $L(G)$, by defining the transfer function $f_w$ as the homomorphism $f_w = f_{e_1} \circ f_{e_2} \circ \cdots \circ f_{e_k}$. The goal of the meet-over-all-paths (MVP) analysis is to compute $\bigcup_{w \in L(G)} f_w(\mathbf{0})$, where $\mathbf{0}$ is the input value of the analysis.

Although MVP is elegant and simply formulated, it has the issue of handling poorly local variables and the frame condition thereof which states that the values of local variables are the same before and after a procedure call (except for the variable used to store the return value). This is due to the fact that the homomorphism is applied on the leaves of the parse tree of $w$ and ignores all the additional information provided by the parse tree itself. Then the homomorphism defining the transfer function for a path must be replaced by a more complex construction which we illustrate next. Take the path $w = c_1r_1c_2r_2l_1c_3$, where $c_1$ and $r_1$ are matching call and return site, respectively. Further assume that $\phi_{(c_1, r_1)}$ is a relation describing the frame condition for the call site $(c_1, r_1)$. The transfer function for $w$ becomes $f_w = [(f_{c_1} \circ f_{r_1} \circ f_{r_2} \circ f_{c_2}) \cap \phi_{(c_1, r_1)}] \circ f_{c_3}$. The difficulty is that, for arbitrarily long subwords (e.g. $c_1c_1c_2r_1$), the relation defined by homomorphism $(f_{c_1} \circ f_{r_1} \circ f_{r_2} \circ f_{c_2})$ needs to be intersected with the frame condition $\phi_{(c_1, r_1)}$. Due to the fact that this intersection is not local, i.e. the distance between the call and the return positions is not bounded, the transfer function $f_w$ is indeed difficult to compute.

Various solutions to this problem were proposed. One is to introduce the frame condition through an abstraction of the stack of procedure calls. Another associates in the control flow graph a special edge with each pair of matching call and return site. Contrary to the other edges representing small steps of the program, this edge models a “big” step which accounts for an arbitrary number of small steps. This latter solution essentially boils down to lifting the transfer functions from the terminals to the productions of the grammar $G$. Hence the analysis outcome instead of being defined by $\bigcup_{w \in L(G)} f_w(\mathbf{0})$, is now defined as $\bigcup_{\pi \in \Gamma_G} f_{\pi}(\mathbf{0})$ where $\Gamma_G$ is the set of productions sequences occurring along the derivations of $G$, i.e. $p_1 \ldots p_n \in \Gamma_G$ if the application of $p_1, \ldots, p_n$ in that order from the start symbol yields a word in $L(G)$.

In this work, we show that whenever $L(G)$ is bounded, i.e. $L(G) \subseteq u^*_1 \ldots u^*_d$, then $L(G)$ can be generated by a set of productions $\Gamma_G \subseteq u^*_1 \ldots u^*_n$. The pattern $u^*_1 \ldots u^*_n$ can be used to guide the analyzer, as we explained in a previous work, where we compute the summary relations between input and output values of procedural programs over integer (local and global) variables. Here MVP is defined by a sequence of underapproximations with increasing coverage of the set of IVP. Whenever the set of IVP is, moreover, a bounded language, the sequence converges in finitely many steps. Thus the result presented here can be seen as the first step to assess the complexity of analyses for recursive programs over integers with
both local and global variables.

We conjecture that our result is relevant also beyond interprocedural program analysis. Future work includes the use of our result for the evaluation of Datalog queries and in particular those over infinite data domain [16], and also for the analysis of branching vector addition systems, and, in particular, their equivalent recursive vector addition systems formulation [2].

2 Preliminaries

Let \( \Sigma \) be an alphabet, that is, a finite nonempty set of symbols. We denote by \( \Sigma^* \) the set of finite words over \( \Sigma \) including \( \varepsilon \), the empty word. We denote concatenation of two words \( u \) and \( v \) by \( u \cdot v \) or simply by \( uv \). Given a word \( w \in \Sigma^* \), let \( |w| \) denote its length and let \( \langle w \rangle \) with \( 1 \leq i \leq |w| \) be the \( i \)-th symbol of \( w \). Given \( w \) and an alphabet \( \Theta \), we define \( w|_{\Theta} \) to be the word obtained by deleting from \( w \) all symbols not in \( \Theta \), for instance \( aaccabdbdb|_{\{a,b\}} = aaabb \).

When \( \Theta = \{a\} \), we abuse notation and write \( w|_{a} \). Let \( \mathbb{N} \) be the set of positive integers, including zero. Let \( v \) be a \( d \)-dimensional vector, that is \( v \in \mathbb{N}^d \), define \((v)_i \) (with \( 1 \leq i \leq d \)) as the \( i \)-th element of \( v \) and \((+ \) as the pointwise addition . For a finite set \( S \), \( |S| \) denotes its cardinality.

Context-Free Grammars. A context-free grammar (or simply grammar) is a tuple \( G = \langle \mathbb{E}, \Sigma, \Delta \rangle \) where \( \mathbb{E} \) is a finite nonempty set of nonterminals, \( \Sigma \) is an alphabet, such that \( \mathbb{E} \cap \Sigma = \emptyset \), and \( \Delta \subseteq \mathbb{E} \times (\Sigma \cup \mathbb{E})^* \) is a finite set of productions. For a production \( \langle X, w \rangle \in \Delta \), often conveniently noted \( X \rightarrow w \), we define its size as \( |\langle X, w \rangle| = |w| + 1 \), and \(|G| = \sum_{p \in \Delta} |p| \) defines the size of the grammar \( G \). Given two strings \( u, v \in (\Sigma \cup \mathbb{E})^* \), a production \( \langle X, w \rangle \in \Delta \) and a position \( 1 \leq j \leq |u| \), we define a step \( u \xrightarrow{\langle X,w \rangle} j v \) if and only if \( (u)_j = X \) and \( v = (u)_1 \cdots (u)_{j-1} w (u)_{j+1} \cdots (u)_{|u|} \). We omit \( \langle X, w \rangle \) or \( j \) above the arrow when it is not important. Step sequences (including the empty sequence) are defined using the reflexive transitive closure of the step relation \( \Rightarrow_G \), denoted \( \Rightarrow^*_G \).

A control word is a sequence of productions. We annotate step sequences with control words as expected: given a control word \( \gamma \) of length \( n \) we write \( u \xrightarrow{\gamma} G v \) whenever there exists \( w_0, \ldots, w_n \in (\Sigma \cup \mathbb{E})^* \) such that \( u = w_0 \xrightarrow{(\gamma)_1} w_1 \xrightarrow{(\gamma)_2} \cdots \xrightarrow{(\gamma)_n} w_n = v \). We call any step sequence \( u \Rightarrow^*_G v \) a derivation whenever \( u \in \mathbb{E} \) and \( v \in \Sigma^* \). We omit the argument \( G \) in the following, when it is clear from the context.

Given a nonterminal \( X \in \mathbb{E} \) and \( Y \in \mathbb{E} \cup \{\varepsilon\} \), i.e. \( Y \) is either a nonterminal or the empty word, we define the set \( L_{X,Y}(G) = \{ u v \in \Sigma^* \mid X \Rightarrow^* G u Y v \} \). The set \( L_{X,Y}(G) \) is called the language of \( G \) produced by \( X \), and is denoted \( L_X(G) \) in the following. For a set \( \Gamma \subseteq \Delta^* \) of control words (also called a control set), we denote by \( \hat{L}_{X,Y}(\Gamma, G) = \{ u v \in \Sigma^* \mid \exists \gamma \in \Gamma : X \xrightarrow{\gamma} u Y v \} \) the language generated by \( G \) using only control words in \( \Gamma \). Given \( G = \langle \mathbb{E}, \Sigma, \Delta \rangle \) and \( X \in \mathbb{E} \), we say that \( G \) is reduced for \( X \) (or just reduced, when \( X \) is not important) if \( L_{X,Y}(G) \neq \emptyset \) and \( L_Y(G) \neq \emptyset \) for every \( Y \in \mathbb{E}, Y \neq X \).

Example 1. Given the grammar \( G = \langle \{X,Y\}, \{a,b,c,d\}, \Delta \rangle \), with productions \( \Delta = \{ X \rightarrow aYb, Y \rightarrow cXd, Y \rightarrow \varepsilon \} \), the language \( L_X(G) = \{(ac)^n ab (db)^n \mid n \geq 0 \} \).

Labeled Graphs. In this paper we use the notion of labeled graph \( G = \langle Q, \mathcal{L}, \delta \rangle \), where \( Q \) is a finite set of vertices, \( \mathcal{L} \) is a set of edge labels, and \( \delta \subseteq Q \times \mathcal{L} \times Q \) is the edge relation. We denote by \( q \xrightarrow{\ell} q' \) the fact that \( (q, \ell, q') \in \delta \). A path \( p = q_0 \ell_1 q_1 \ell_2 \cdots \ell_n q_n \) in \( G \) is a sequence which alternates vertices and labels, starts and ends with vertices and such that \( (q_i, \ell_{i+1}, q_{i+1}) \in \delta \)
Generating Bounded Languages Using Bounded Control Sets

Index-bounded depth-first derivations. For a set $S \subseteq (\Sigma \cup \Xi)$, of both nonterminal and terminal symbols, and a set $I \subseteq \mathbb{N}$ of positive integers, we define $S^I = \{x^{(i)} \mid x \in S, i \in I\}$ to be a set of ranked symbols. When $I = \{i\}$ we write $S^{(i)}$ instead of $S^{[i]}$. Given a word $w \in (\Sigma \cup \Xi)^*$ of length $n$, and a $n$-dimensional vector $\alpha \in \mathbb{N}^n$, we define $w^{\alpha}$ as the ranked word (r-word) $(w)_1^{(\alpha_1)} \cdots (w)_n^{(\alpha_n)}$ over the alphabet $(\Sigma \cup \Xi)^n$. Given $\epsilon \in \mathbb{N}$, we write $w^{(\epsilon)}$ for the r-word $w^{(\epsilon,c\cdots,c)}$. For instance, $abc^{(1,2,3)} = a^{(1)}b^{(2)}c^{(3)}$ and $ab^{(2)} = a^{(2)}b^{(2)}$. The following function is useful in the upcoming developments: $\text{Rank}(w^{\alpha}) = \{w_j \in \Xi \mid \text{there exists ranked word } w^\alpha \text{ such that } w^\alpha \text{ contains the nonterminal } w_j\}$

Let $G = (\Xi, \Sigma, \Delta)$ be a grammar and $u \xrightarrow{\alpha} v$ be a step, for some production $(Z, w) \in \Delta$ and $1 \leq j \leq |u|$. If $\alpha \in \mathbb{N}^{|u|}$ is a vector, the corresponding ranked step (r-step) is defined as follows: $u^{\alpha} \xrightarrow{\alpha} v^{\beta}$ if and only if $u^{(j)} = Z$ and $v^{\beta} = (u^{(\alpha_1)})_{j-1} (u^{(\alpha_{j+1})})_1 \cdots (u^{(\alpha_n)})_1$, where $m = \max \{\text{Rank}((u^{(\alpha_1)})_1 \cdots (u^{(\alpha_m)})_1 \cdots (u^{(\alpha_n)})_1) \mid -1\}$. In other words, each symbol in $w$ is given rank $m + 1$, where $m$ is the maximum among the ranks of the nonterminals in $w^\alpha$ with position $j$ omitted. If there are no nonterminals in $w^\alpha$, other than then $(w^{(\alpha_j)})$, then $m + 1 = -1 + 1 = 0$. An r-step is said to be depth-first, denoted $u^{\alpha} \xrightarrow{\text{df}} v^{\beta}$ whenever, in the above definition, we have $m \leq (\alpha_j)$, i.e. the rank of the nonterminal at position $j$ where the rule applies, is maximal. Observe that the ranks of the terminals play no role, so we can safely ignore them.

An r-step sequence is said to be depth-first if all of its r-steps are depth-first. Finally, an unranked step sequence $w_0 \xrightarrow{(\gamma)_{i/j}} w_1 \cdots w_{n-1}^{(\gamma)_{n/j}} w_n$ is said to be depth-first, written $w_0 \xrightarrow{\text{df}} w_n$, if there exist vectors $\alpha_1 \in \mathbb{N}^{|w_1|}$, ..., $\alpha_n \in \mathbb{N}^{|w_n|}$ such that $w_0(0)^{(\gamma)_{i/j}} \xrightarrow{\text{df}} w_1^{(\alpha_1)} \cdots w_n^{(\alpha_n)}$ holds.

Next, we define $\Upsilon = \{u^{\alpha} \in ((\Sigma \cup \Xi)^*)^n \mid \forall j \in \text{Rank}(u^{\alpha}), \{0, \ldots, j\} \subseteq \text{Rank}(u^{\alpha})\}$, i.e. if a nonterminal in a r-word $w^\alpha \in \Upsilon$ has rank $j$, then each value between $0$ and $j$ is also a rank of nonterminal in the r-word $w^\alpha$. The set $\Upsilon$ contains the empty word $\epsilon$ and is closed under the application of depth-first r-steps. Hence, if $w_0^{(0)} \xrightarrow{\text{df}} w_1^{(\alpha_1)} \cdots w_{n-1}^{(\alpha_{n-1})} \xrightarrow{\text{df}} w_n^{(\alpha_n)}$, we have $w_i^{(\alpha_i)} \in \Upsilon$ for all $i$, $1 \leq i \leq n$, since $w_0^{(0)} \in \Upsilon$ (because $\text{Rank}(w_0^{(0)}) = \{0\}$).

Example 2. Consider $G = \langle \{X, Y, Z\}, \{a, b\}, \Delta \rangle$ with productions $\Delta = \{X \rightarrow Y Z, Y \rightarrow a Y, Y \rightarrow e, Z \rightarrow Z b, Z \rightarrow \epsilon\}$. Then $X^{(0)} = \{X, Y Z\}, Y^{(0)} = \{Y, a Y\}, a^{(1)} = \{Y (1) Z^{(0)}\}, a^{(1)} Z^{(0)} = \{Z b\}, a^{(1)} Z^{(0)} b^{(0)} = \{Z e\}, a^{(1)} b^{(0)}$ is a depth-first derivation, whereas $X^{(0)} = \{X, Y Z\}, Y^{(0)} = \{Y, a Y\}, a^{(1)} = \{Y (1) Z^{(0)}\}, a^{(1)} Z^{(0)} = \{Z b\}, a^{(1)} Y^{(1)} Z^{(0)} = \{Z e\}, a^{(1)} Z^{(0)} b^{(0)} = \{Z e\}, a^{(1)} b^{(0)}$ is not a depth-first derivation, because in the r-step $a^{(1)} = \{X, Y (1) Z^{(0)}\}, a^{(1)} Z^{(0)} = \{Z b\}, a^{(1)} Y^{(1)} Z^{(0)} = \{Z e\}, a^{(1)} Z^{(0)} b^{(0)} = \{Z e\}, a^{(1)} b^{(0)}$ max $\{\text{Rank}(a^{(1)} Y^{(1)})\} = 1 \notin 0$ (the rank of the nonterminal $Z$, to which the production applies). Also, observe that $a^{(1)} Y^{(1)} Z^{(0)} b^{(2)} \notin \Upsilon$ because $[0, 1, 2] \notin \{1, 2\}$, however all the r-words of the depth-first derivation belong to $\Upsilon$.

Index-bounded derivations. For a given integer constant $k > 0$, a word $u \in (\Sigma \cup \Xi)^*$ is said to be of index $k$, if $u$ contains at most $k$ occurrences of nonterminals (formally, $|u|_\Xi \leq k$). A
step \( u \Rightarrow v \) is said to be \( k \)-index, denoted \( u \Rightarrow_k v \), if and only if both \( u \) and \( v \) are of index \( k \). As expected, a step sequence is \( k \)-index if all its steps are \( k \)-index. For instance, both derivations from Ex. 2 are of index 2.

The previous definitions extend naturally to \( r \)-steps and \( r \)-step sequences. We further define \( \Upsilon^{(k)} = \{ w^\alpha \in \Upsilon \mid |w^\alpha| \leq k \} \) as those \( r \)-words from \( \Upsilon \) with at most \( k \) occurrences of nonterminals. It can be shown using the pigeonhole principle that each \( w^\alpha \in \Upsilon^{(k)} \) is such that \( (\alpha)_j \leq k - 1 \) for all \( j \) with \( (w)_j \in \Xi \), that is, no word of \( \Upsilon^{(k)} \) has a nonterminal whose rank exceeds \( k - 1 \).\footnote{Because ranks start from 0, and the set \( \text{Rank}(w^\alpha) \) forms an interval of integers.} We write the fact that a \( r \)-step sequence \( u^\alpha \Rightarrow^* v^\beta \) is both \( k \)-index and depth-first as \( u^\alpha \Rightarrow^{df(k)} v^\beta \). Similar to the fact that \( \Upsilon \) is closed under depth-first \( r \)-steps, we have that \( \Upsilon^{(k)} \) is closed under the application of \( k \)-index depth-first \( r \)-steps. Given \( X \in \Xi \), \( Y \in \Xi \cup \{\varepsilon\} \) and a constant \( k > 0 \), we define the \( k \)-index language (with respect to \( X \) and \( Y \) and \( k \)-index depth-first control set, respectively, as:

\[
L^{(k)}_{X,Y}(G) = \{ u \in \Sigma^* \mid X \Rightarrow^* u \, Y \, v \}, \quad \Gamma^{df(k)}(G) = \{ \gamma \in \Delta^* \mid \exists u^\alpha, v^\beta \in \Upsilon, u^\alpha \Rightarrow^{df(k)} v^\beta \}
\]

\textbf{Example 3.} For \( G \) of Ex. 2 \( \Gamma^{df(1)} = (Y, aY)^*(Y, \varepsilon) \cup (Z, Zb)^*(Z, \varepsilon) \) and \( \Gamma^{df(2)} = (X, YZ)(Y, aY)'*(Y, \varepsilon)(Z, Zb)'*(Z, \varepsilon) \cup (X, YZ)(Z, Zb)'*(Z, \varepsilon)(Y, aY)'*(Y, \varepsilon) \cup \Gamma^{df(1)} \).

\textbf{Depth-first index-bounded control sets.} For a \( r \)-word \( w^\alpha \in \Upsilon \), let \( [w^\alpha] \) be the \( r \)-word \( (w^\alpha \downarrow_{\Xi^{(0)}}) \, (w^\alpha \downarrow_{\Xi^{(1)}}) \ldots (w^\alpha \downarrow_{\Xi^{(|\alpha|)}}) \). Intuitively, \([w^\alpha] \) projects out the terminals of \( w \), and orders the remaining nonterminals in the increasing order of their ranks. For instance, \( [Y^{(0)}] = Y^{(0)} \), and \([a^{(1)}] = Z^{(0)} \). The \([\cdot] \) operator is naturally lifted from \( r \)-words to sets of \( r \)-words. In particular, the set \( [\Upsilon^{(k)}] \) (a subset of \( \Upsilon^{(k)} \)) is of interest in the following.

Let \( A^{df(k)}(G) = (\Upsilon^{(k)}), \Gamma, \rightarrow) \) be a labeled graph, where \( \Upsilon^{(k)} \) is the set of vertices, the set \( \Gamma \) of productions of \( G \) is the set of edge labels, and \( \rightarrow \) is the edge relation defined as \( u^\alpha \rightarrow v^\beta \) if and only if \( \exists w^\eta \in \Upsilon^{(k)}, u^\alpha \Rightarrow^{df(k)} w^\eta \land v^\beta = [w^\eta] \). We denote by \( |A^{df(k)}(G)| = \|\Upsilon^{(k)}\| \) the size of the graph \( A^{df(k)}(G) \). As an example, Fig. 1 shows the subgraph of \( A^{df(k)}(G) \) reachable from \( X \), where \( G \) is the grammar from Ex. 2.

\textbf{Figure 1} The subgraph \( A^{df(k)}(G) \) reachable from \( X \), for \( k \geq 2 \) and for the grammar \( G \) of Ex. 2.
Generating Bounded Languages Using Bounded Control Sets

Lemma 4. Given $G = \langle \Xi, \Sigma, \Delta \rangle$, and $k > 0$, for each $X \in \Xi$, $Y \in \Xi \cup \{\varepsilon\}$ and $\gamma \in \Delta^*$, there exist $v_1, v_2 \in \Sigma^*$ such that $X \xrightarrow{\gamma} v_1 Y v_2$ if and only if $[X^{(0)}] \leq_k [Y^{(0)}]$ in $A^{\text{df}(k)}(G)$.

Assumption 5 (2NF). A grammar $G$ is in 2-normal form (2NF for short) if it has only productions $X \rightarrow w$, with at most two symbols on the right-hand side, i.e. $|w| \leq 2$. Any grammar $G$ can be converted into an equivalent 2NF grammar $H$, such that $|H| = O(|G|)$, in time $O(|G|^{1+})$. From now on, we shall consider without loss of generality that each grammar is in 2NF.

Example 6. The following 2NF grammar is equivalent to the grammar from Ex. 1: $G' = \langle \{X, Y, Z, T\}, \{a, b, c, d\}, \Delta' \rangle$ where $\Delta' = \{X \rightarrow aY, Y \rightarrow Zb, Z \rightarrow cT, Z \rightarrow \varepsilon, T \rightarrowXd\}$, i.e. we have $L_X(G) = L_X(G') = \{(ac)^n ab (db)^n \mid n \in \mathbb{N}\}$.

Recall that each vertex of $A^{\text{df}(k)}$ is a r-word $u^\alpha \in \Upsilon^{(k)}$ which consists of nonterminals only, $|u^\alpha| \leq k$, and all ranks in $\alpha$ are between 0 and $k - 1$. Also, if the grammar $G$ is in 2NF, the subset of $\Upsilon^{(k)}$ consisting of r-words in which at most 2 nonterminals have the same rank, is easily shown to be closed under the application of k-index depth-first r-steps. For simplicity, from now on (because of Assumption 5) we shall write $\Upsilon^{(k)}$ for this subset, hence $A^{\text{df}(k)}$ for the labeled graph with vertex set $\Upsilon^{(k)}$. Since the length of each vertex is at most $k$ and, for each $i$, $0 \leq i \leq k - 1$, there are at most $|\Xi|^2$ choices of nonterminals with rank $i$, the number of vertices in $A^{\text{df}(k)}$ is bounded by $|\Xi|^{2k} \leq |G|^{2k}$, hence $|A^{\text{df}(k)}| = |G|^{O(k)}$.

4 Bounded Languages

A bounded expression $b$ (over the alphabet $\Sigma$) is a regular expression of the form $w_1^* \ldots w_d^*$ where $w_1, \ldots, w_d \in \Sigma^*$ are nonempty words. We define the size of a bounded expression as $|b| = \sum_{i=1}^d |w_i|$. We abuse notation and use $b$ to denote both the bounded expression and its language. When each word $w_i$ consists of a single letter $b$ is said to be a d-letter-bounded expression, or simply letter-bounded expression, when $d$ is not important. A letter-bounded expression is said to be strict if all its symbols are distinct (e.g. $a^* b^* c^*$ is strict while $a^* b^* a^*$ is not). A language $L \subseteq \Sigma^*$ is bounded (for $b$) (resp. letter-bounded) if $L \subseteq b$, for some bounded expression $b$ (resp. letter-bounded expression). For instance, the language $\{(ac)^n ab (db)^n \mid n \geq 0\}$ generated by the grammar from Ex. 1 is bounded, as it is included in the language of the bounded expression $(ac)^*(ab)^*(db)^*$.

In the rest of this section, for a given bounded expression $b = w_1^* \ldots w_d^*$ over $\Sigma$, we associate the strict d-letter-bounded expression $\bar{b} = a_1^* \ldots a_d^*$ over an alphabet $\mathcal{A}$, disjoint from $\Sigma$, i.e. $\mathcal{A} \cap \Sigma = \emptyset$, and a homomorphism $h: \mathcal{A} \rightarrow \Sigma^*$ mapping as follows: $h(a_i) = w_i$, for all $1 \leq i \leq d$.

Given $G = \langle \Xi, \Sigma, \Delta \rangle$, $X \in \Xi$, and a bounded expression $b$ over $\Sigma$, Luker [13, Theorem 1] shows that if $L_X(G) \subseteq b$ then $L_X^{(k)}(G) = L_X(G)$ for some $k \geq 0$. Our goal is thus to compute a bounded expression $\Gamma_b$ over $\Delta$, such that $L_X^{(k)}(G) \subseteq L(\Gamma_b \cap \Gamma^{\text{df}(k)})$. Lemma 2 simplifies the problem by reducing the construction of such a bounded expression for a $k$-index bounded language to the construction of a bounded expression for a $k$-index strict letter-bounded language. The problem for a grammar $G$, whose language is bounded, is thus reduced to a similar but simpler problem for another grammar $G^o$, whose language is included in a strict letter-bounded expression.

\footnote{From his proof, we deduce that taking $k = O(|\Xi|)$ suffices.}
Lemma 7. Given a grammar $G = \langle \Xi, \Sigma, \Delta \rangle$, $X \in \Xi$, a bounded expression $b$ over $\Sigma$ such that $L_X(G) \subseteq b$, the following hold:
1. there exists a grammar $G'^n = \langle \Xi^n, A_n, \Delta^n \rangle$ and a set of nonterminals $V_X \subseteq \Xi^n$, such that $\bigcup_{Y \in V_X} L^{(k)}(G'^n) = h^{-1}(L^{(k)}(G)) \cap b$, for all $k > 0$.
2. there exists a mapping $\xi : \Delta^n \rightarrow \Delta$, such that, given $k > 0$ and $\Gamma \subseteq \Gamma^{dr(k)}(G^n)$, a control set such that, for all $Y \in V_X$, it is the case that $L_Y^{(k)}(G^n) \subseteq \tilde{L}_Y(\Gamma, G^n)$ holds, then $L_X^{(k)}(G) \subseteq \tilde{L}_X(\Gamma, G)$, and
3. both $G^n$ and $\xi$ are computable in time $O(|b|^3 \cdot |G|)$.

Ex. 8 below illustrates the construction of such a grammar $G^n$, for the grammar of Ex. 6. The formal definitions and proofs of correctness are given in Appendix A.

Example 8. Let us consider the bounded expression $b = (ac)^n (ab)^n (db)^n$. Consider the grammar $G^b$ with the following productions:

\[
\begin{align*}
Q_1^{(1)} &\rightarrow aQ_2^{(1)} | \varepsilon & \quad Q_1^{(2)} &\rightarrow aQ_2^{(2)} | \varepsilon & \quad Q_1^{(3)} &\rightarrow dQ_2^{(3)} | \varepsilon \\
Q_2^{(1)} &\rightarrow cQ_1^{(1)} | cQ_1^{(2)} | cQ_1^{(3)} & \quad Q_2^{(2)} &\rightarrow bQ_1^{(2)} | bQ_1^{(3)} & \quad Q_2^{(3)} &\rightarrow bQ_1^{(3)}
\end{align*}
\]

It is easy to check that $b = \bigcup_{i=1}^3 L_1^{(i)}(G^b)$. Call $G$ the grammar from Ex. 6 – we have $L_X(G) = \{(ac)^n (ab)^n (db)^n | n \in \mathbb{N}\}$. The following productions define a grammar $G^c$:

\[
\begin{align*}
[q_1^{(j)}Xq_3^{(3)}] &\xrightarrow{p_1} a[q_2^{(j)}Yq_3^{(3)}] & \quad [q_2^{(1)}Yq_3^{(3)}] &\xrightarrow{p_2} [q_2^{(1)}Zq_3^{(2)}]b \\
[q_2^{(2)}Zq_3^{(2)}] &\xrightarrow{p_3} c[q_1^{(j)}Tq_3^{(3)}] & \quad [q_2^{(2)}Zq_3^{(2)}] &\xrightarrow{p_4} \varepsilon \\
[q_1^{(j)}Tq_3^{(3)}] &\xrightarrow{p_5} [q_1^{(j)}Xq_3^{(3)}]d, \text{ for } j = 1, 2 & \quad [q_2^{(1)}Yq_3^{(3)}] &\xrightarrow{p_6} a[q_2^{(2)}Yq_3^{(3)}] \\
[q_2^{(2)}Yq_3^{(1)}] &\xrightarrow{p_7} [q_2^{(2)}Zq_3^{(2)}]b & \quad [q_2^{(2)}Zq_3^{(2)}] &\xrightarrow{p_8} [q_2^{(2)}Zq_3^{(2)}]b
\end{align*}
\]

It is routine to check that $L_{[q_1^{(1)}Xq_3^{(3)}]}(G^c) \cup L_{[q_1^{(2)}Xq_3^{(3)}]}(G^c) = L_X(G) \cap L(b) = L_X(G)$.

Finally, let $A = \{a_1, a_2, a_3\}$ and $h : A \rightarrow \Sigma^*$ be the homomorphism given by $h(a_1) = ac, h(a_2) = ab$ and $h(a_3) = db$. The grammar $G^m$ results from deleting $a$’s and $d$’s in $G^c$ and replacing $b$ in $p_2$ by $a_3$, $b$ in $p_7$ by $a_2$ and $c$ by $a_1$. Then, it is an easy exercise to check that $h^{-1}(L_X(G)) \cap b = L_{[q_1^{(1)}Xq_3^{(3)}]}(G^m) \cup L_{[q_1^{(2)}Xq_3^{(3)}]}(G^m) = \{a_1^m a_2 a_3^m | n \in \mathbb{N}\}$.

5 Building Bounded Control Sets

This section first gives the construction of a bounded control set for a strict letter-bounded context-free language, together with an upper bound on the time complexity of the algorithm implementing this construction (Theorem 11), and a proof of its optimality (Lemma 13). The results are then extended to the more general case of bounded languages (Theorem 12).

For the rest of this section, let $G = \langle \Xi, A, \Delta \rangle$ be a grammar in 2NF (Assumption 5), and $X \in \Xi$ be a nonterminal such that $L_X(G) \subseteq b$, where $b = a_1^* \ldots a_n^*$ is a strict $d$-letter-bounded expression over $A$. Without losing generality, we assume from now on that $b$ is minimal in the sense that $L_X(G) \subseteq b$, and for every subexpression $b'$, resulting from deleting some $a_i^*$ from $b$, we have $L_X(G) \nsubseteq b'$. Since $b$ is a strict letter-bounded expression, the minimal such subexpression is unique. Furthermore, we assume, without losing generality, that $G$ is reduced for $X[A]$ a grammar can be reduced in polynomial time, by eliminating unreachable and unproductive nonterminal.

---

3 That is $L_{X,Y}(G) \neq \emptyset$ and $L_Y(G) \neq \emptyset$, for every $Y \in \Xi, X \neq Y$.

4 In [5] Lemma 1.4.4.
Let us now survey the main idea of the construction. We shall build a bounded expression \( \Gamma_b \) over \( \Delta \), that captures the \( k \)-index language of \( G \) in the following sense: \( L_{X}^{k}(G) = \hat{L}(\Gamma_b \cap \Gamma_{bf(k)}G) \). The construction is based on the following ingredients: (i) a decomposition of \( k \)-index depth-first derivations, that distinguishes between a prefix producing a word from the 2-letter bounded expression \( a_{1}^{*}a_{2}^{*} \), and a suffix producing two words included in bounded expressions strictly smaller than \( b \) (Section 5.1), and (ii) an algorithm (Algorithm 1) for building bounded control sets for \( s \)-letter bounded languages, where \( s \geq 0 \) is a constant (in our case, at most 2) (Section 5.2). The required bounded expression \( \Gamma_b \) is built inductively over the structure of this decomposition, applying at each step Algorithm 1 in order to build bounded expressions for the sets of derivations producing 2-letter bounded languages. Both the main algorithm (Algorithm 2) and theorem (Theorem 11) of the paper rely on a combination of these results (Section 5.3).

### 5.1 A Decomposition Lemma

Our construction of the control set \( \Gamma_b \), for a given letter-bounded context-free language \( L_{X}(G) \subseteq \hat{b} = a_{1}^{*} \ldots a_{d}^{*} \), is by induction on \( d \geq 1 \), and is inspired by a decomposition of the derivations in \( G \), given by Ginsburg\(^5\) Because his decomposition is oblivious to the index or the depth-first policy, it is too weak for our needs. Therefore, we give first a stronger decomposition result for \( k \)-index depth-first derivations (Lemma 9).

Basically, for every \( k \)-index depth-first derivation with control word \( \gamma \), its productions can be rearranged into a \((k + 1)\)-index depth-first derivation, consisting of a prefix \( \gamma' \) producing a word \( a_{1}^{d} \ldots a_{d}^{*} \), then a production \( (X_i, w) \) followed by two control words \( \gamma' \) and \( \gamma' \) such that they respectively produce words included in some bounded expression \( a_{1}^{*} \ldots a_{m}^{*} \) and \( a_{m}^{*} \ldots a_{r}^{*} \) where \( \max(m - t, r - m) < d - 1 \) (Lemma 9).

Let us first define the partition \((\Xi_{\mathbb{R}d}, \Xi_{\mathbb{R}d})\) of \( \Xi \), as follows:

\[
Y \in \Xi_{\mathbb{R}d} \iff L_Y(G) \cap (a_1 \cdot A^*) \neq \emptyset \quad \text{and} \quad L_Y(G) \cap (A^* \cdot a_d) \neq \emptyset; \quad \Xi_{\mathbb{R}d} = \Xi|_{\Xi_{\mathbb{R}d}}.
\]

Since \( \hat{b} = a_{1}^{*} \ldots a_{d}^{*} \) is the minimal bounded expression such that \( L_{X}(G) \subseteq \hat{b} \), then \( a_1 \) occurs in some word of \( L_{X}(G) \) and \( a_d \) occurs in some word of \( L_{X}(G) \). Thus it is always the case that \( \Xi_{\mathbb{R}d} \neq \emptyset \), since \( X \in \Xi_{\mathbb{R}d} \). The partition of nonterminals into \( \Xi_{\mathbb{R}d} \) and \( \Xi_{\mathbb{R}d} \) induces a decomposition of the grammar \( G \). First, let \( G^2 = (\Xi, A, \Delta^3) \), where:

\[
\Delta^3 = \{ (X_j, w) \in \Delta \mid X_j \in \Xi_{\mathbb{R}d} \} \cup \{ (X_j, u, X_r, v) \in \Delta \mid X_j, X_r \in \Xi_{\mathbb{R}d} \}.
\]

Observe that, because \( G \) is in 2NF (Assumption 5), each production \( (X_j, u, X_r, v) \in \Delta \) is such that either \( u = \varepsilon \), or \( v = \varepsilon \). Then, for each production \( (X_i, w) \in \Delta \) such that \( X_i \in \Xi_{\mathbb{R}d} \) and \( w \in (\Xi_{\mathbb{R}d} \cup A)^* \), we define the grammar \( G_{i, w} = (\Xi, A, \Delta_{i, w}) \), where:

\[
\Delta_{i, w} = \{ (X_i, v) \in \Delta \mid X_j \in \Xi_{\mathbb{R}d} \} \cup \{ (X_i, w) \}.
\]

The decomposition of derivations is formalized in the following lemma:

\> **Lemma 9.** Given a grammar \( G = (\Xi, A, \Delta) \), a nonterminal \( X \in \Xi \) such that \( L_{X}(G) \subseteq \hat{b} = a_{1}^{*} \ldots a_{d}^{*} \) for some \( d \geq 3 \), and \( k > 0 \), for every derivation \( \hat{\gamma}_{df(k)} \in G \), there exists a production \( (X_i, yz) \in \Delta \) with \( X_i \in \Xi_{\mathbb{R}d} \), \( y, z \in \Xi_{\mathbb{R}d} \cup A \cup \{ \varepsilon \} \), and control words \( \gamma^2 \in (\Delta^3)^* \), \( \gamma_y, \gamma_z \in (\Delta_{i,yz})^* \), such that \( \gamma^2 \in (\Delta^3)^* \), \( \gamma_y, \gamma_z \) is a permutation of \( \gamma \) and:

\(^5\) In [5] Chapter 5.3, Lemma 5.3.3].
1. \( X \xrightarrow{\gamma_1} G^2 u X_{t+1} v \) is a step sequence in \( G^2 \) with \( u, v \in A^* \);

2. \( y \xrightarrow{\gamma_y} G_{t+1} u_y \) and \( z \xrightarrow{\gamma_z} G_{t+1} u_z \) are (possibly empty) derivations in \( G_{t+1} \) (\( u_y, u_z \in \mathcal{A}^* \)), for some integers \( k_y, k_z > 0 \), such that \( \max(k_y, k_z) \leq k \) and \( \min(k_y, k_z) \leq k - 1 \);

3. \( X \xrightarrow{\gamma} G_w \) if \( y \xrightarrow{\gamma_y} G_{t+1} u_y \), and \( X \xrightarrow{\gamma} G_w \) if \( z \xrightarrow{\gamma_z} G_{t+1} u_z \);

4. \( L_{G,X}(G^2) \subseteq a_1^s \ldots a_m^s \);

5. \( L_{G}(a_{\ell}^s \ldots a_{r}^s) \subseteq a_{\ell}^s \ldots a_{m}^s \) if \( \ell \leq m \leq r \leq d \), such that \( \max(m - \ell, r - m) < d - 1 \).

### 5.2 Constant \( s \)-Letter Bounded Languages

The second ingredient of our construction is an algorithm for building bounded control sets for \( s \)-letter bounded languages \( L_X(G) \subseteq a_1^s \ldots a_m^s \), when \( s \geq 0 \) is a constant\(^6\), i.e. not part of the input. In the following, we consider the labeled graph \( A^{df(k)} = \langle [\{\gamma\}(k)], \Delta, \rightarrow \rangle \), whose paths correspond to the \( k \)-index depth-first step sequences of \( G \) (Lemma 5). Recall that the number of vertices in this graph is \( |A^{df(k)}| \leq |G|^{2k} \).

First we introduce several notions related to the paths of \( A^{df(k)} \). The endpoints of a path \( \pi \) are the positions in \( \pi \) corresponding to the first and last vertex, respectively. The concatenation \( \pi : \pi' \) of two paths \( \pi, \pi' \) is defined as expected, provided the right endpoint of \( \pi \) and the left endpoint of \( \pi' \) refer to the same vertex. The set of repeating positions of \( \pi \) is the set of positions \( (i, j) \) in \( \pi \) where \( i < j \), and \( (\pi)_i, (\pi)_j \) are the same vertex. A path \( \pi \) is said to be acyclic if its set of repeating positions is empty; elementary if its set of repeating positions is empty or equal to the endpoints; and cyclic if its set of repeating positions includes the endpoints. So when we talk about elementary cycles we mean those paths of which each has a set of repeating positions equal to the endpoints. For a pair \( q, q' \in \mathcal{Q} \) of vertices, we denote by \( \Pi(q,q') \) the set of paths from \( q \) to \( q' \); and we denote by \( \Omega(q) \) the set of elementary cycles starting and ending in \( q \). Hence \( \Omega(q)^s \) denotes the set of cyclic paths (not necessarily elementary) from and to \( q \) each of which results from the concatenation of elementary cycles from \( \Pi(q,q') \) to \( q \).

Given a path \( \pi \in \Pi(X^{\infty}, \varepsilon) \) labeled with a control word \( \omega(\pi) = \gamma \), the word \( w \) induced by \( \pi \) is the word produced by the derivation \( X \xrightarrow{\gamma} w \). Further observe that \( w \) has the form \( a_1^{s_1} \ldots a_r^{s_r} \) and coincides with \( (\gamma_{i_1} \ldots \gamma_{i_s}) \) since \( w \in L_X(G) \subseteq a_1^s \ldots a_m^s \). Following up on Ex. 3 given the control word \( \gamma = (X, Y, Z) (Y, aY)^3 (Y, \varepsilon) (Z, b) \in \gamma(a, b) \) we have \( (\gamma_{i_1} \ldots \gamma_{i_s}) = a_i^{s_i} b^d \).

Algorithm 4 describes the effective construction of the bounded expression \( \Gamma_k \) over the productions of \( G \) using the sets of elementary cycles \( \Omega(q) \), for the vertices \( q \) of \( A^{df(k)} \). The crux is to find, for each vertex \( q \) of \( A^{df(k)} \), a subset \( C_q \subseteq \Omega(q) \) of the elementary cycles having \( q \) at the endpoints, such that the set of words induced by \( C_q \) is that of \( \Omega(q) \). Recall that each induced word is necessarily of the form \( a_1^{s_1} \ldots a_r^{s_r} \). Since the only vertex occurring more than once in an elementary cycle \( \rho \in \Omega(q) \) is the endpoint \( q \), we have that \( |\rho| \) is at most the number of vertices \( |A^{df(k)}| \). Moreover, since \( G \) is in 2NF, no word induced by a elementary cycle in \( \Omega(q) \) is longer than \( 2|A^{df(k)}| \leq 2|G|^{2k} \). The number of induced words \( a_1^{s_1} \ldots a_r^{s_r} \) by cycles of \( \Omega(q) \) is thus bounded by the number of nonnegative solutions of the

---

\(^6\) In our case \( s = 0, 1, 2 \), but the construction can be generalized to any constant \( s \geq 0 \).

\(^7\) For a production \( p = (Z, v) \in \Delta \), we define \( p_{i_1 \ldots i_s} = v_{i_1 \ldots i_s} \).
inequality $x_1 + \cdots + x_s \leq 2|G|^{2k}$, which, in turn, is $|G|^{O(k)}$. So for each $\langle i_1, \ldots, i_s \rangle$ such that $i_1 + \cdots + i_s \leq 2|G|^{2k}$, it suffices to include in $C_\varphi$ one cycle from $\Omega(q)$ which induces the word $a_{i_1} \cdots a_{i_s}$. We explain this next.

Lines 2–5 of Algorithm 1 build a graph $H$ with vertices $\langle q, a_{i_1} \cdots a_{i_s} \rangle$, where $q \in \mathcal{T}(k)$. The graph is a bounded finite and computable graph. There is an edge between two vertices $\langle q, v \rangle$ and $\langle q', v' \rangle$ in $H$ if and only if $q \xrightarrow{\delta} q'$ in $A^{dr}(k)$ and $v'$ is obtained from $v$ by adding the terminals produced by $p$ to the corresponding positions (line 4). The sets $C_\varphi$ are computed by applying Dijkstra’s shortest path algorithm to the graph $H$ (all edges weight 1). The algorithm populates $C_\varphi$ only with the shortest paths $\langle q, v \rangle \rightarrow \ast \langle q, a_{i_1} \cdots a_{i_s} \rangle$ such that $i_1 + \cdots + i_s \leq 2|G|^{2k}$ (line 8). Once the $C_\varphi$ sets are computed, we move towards computing $\Gamma_\beta$, by first computing the bounded expression $B_0$ (lines 9–10), which captures, for each $\rho$ of $C_\varphi$ and vertex $q$ of $A^{dr}(k)$, the control word $\omega(p)$ and all its repetitions ($\omega(p)^2, \omega(p)^3, \ldots$). This is achieved via the function $\text{Concat}(\{\gamma_1, \ldots, \gamma_n\}) = \gamma_1 \cdots \gamma_n$ (line 9). Finally, we compute $\Gamma_\beta$.

The reason $\Gamma_\beta$ is sufficient to cover all induced words stems from the following result: for each path $\pi$ in $A^{dr}(k)$, there exists another path $\pi'$ with the same endpoints as $\pi$, such that their induced words coincide, and, moreover, $\pi'$ can be factorized as $\pi_1 \cdot \theta_1 \cdot \pi_2 \cdot \theta_2 \cdot \pi_3$, where $\pi_1 \in \Pi(q_0, q_1), \pi_2 \in \Pi(q_j, q_{j+1})$ and $\pi_3 \in \Pi(q_1, q_k)$ for each $1 < j < k$ are acyclic paths, $\theta_1 \in (\Omega(q_1))^*$, $\ldots$, $\theta_k \in (\Omega(q_k))^*$ are cycles, and $k \leq |A^{dr}(k)|$ (Proposition 19 in Appendix B). We cover each segment $\pi_1$ by a bounded expression $C$ (lines 10–12), and each segment $\pi_j$ by $B_0$ (lines 14–15), which yields the required expression $\Gamma_\beta$. The following lemma formalizes the correctness of Algorithm 1 and provides an upper bound on its time complexity.

\textbf{Lemma 10.} Let $G = (\Sigma, A, \Delta)$ be a grammar and $\tilde{b}$ is a strict $s$-letter-bounded expression over $A$, where $s > 0$ is a fixed constant. Then, for each $k > 0$ there exists a bounded expression $\Gamma_\beta$ over $\Delta$ such that, for all $X \in \Sigma$ and $Y \in \Sigma \cup \{\varepsilon\}$, we have $L_{X,Y}^{(k)}(G) \subseteq \tilde{L}_{X,Y}(\Gamma_\beta \cap A^{dr}(k), G)$, provided that $L_{X,Y}(G) \subseteq \tilde{b}$. Moreover, $\Gamma_\beta$ is computable in time $|G|^{O(k)}$.

\begin{algorithm}[h]
\caption{Control Sets for Grammars with Constant Size Bounded Expressions}
\begin{algorithmic}[1]
\REQUIRE A grammar $G = (\Sigma, A, \Delta)$, a strict $s$-letter-bounded expression $\tilde{b} = a_1^s \cdots a_s^s$ over $A$, where $s > 0$ is a fixed constant, and $k > 0$
\ENSURE a bounded expression $\Gamma_\beta$ over $\Delta$ such that $L_{X,Y}^{(k)}(G) \subseteq \tilde{L}_{X,Y}(\Gamma_\beta \cap A^{dr}(k), G)$ for all nonterminal(s) $X \in \Sigma$ and $Y \in \Sigma \cup \{\varepsilon\}$, such that $L_{X,Y}(G) \subseteq \tilde{b}$

\STATE $\text{Val} \leftarrow \{a_1^s \cdots a_s^s \mid \sum_{j=1}^s k_j \leq 2|G|^{2k}\}$
\STATE $\mathcal{V} \leftarrow \mathcal{T}(k) \times \text{Val}$ \quad \Rightarrow \text{considering } |\mathcal{T}(k)| \leq |G|^{2k} \text{ suffices}
\STATE $\delta \leftarrow \{(q, a_{i_1}^s \cdots a_{i_s}^s) \xrightarrow{\mathcal{V}} \{(q', a_{i_1}^s \cdots a_{i_s}^s) \mid q \xrightarrow{\delta} q' \text{ in } A^{dr}(k), \forall \ell, a_{i_\ell}^t = a_{j_\ell}^t \cdot (P_{i_\ell})_\eta\}\}$
\STATE $\mathcal{H} \leftarrow (\mathcal{V}, \Delta, \delta)$
\STATE $B_0 \leftarrow \varepsilon$
\STATE $\text{for } q \in \mathcal{T}(k)$ \do
\STATE $C_q \leftarrow \text{ComputeShortestPaths}(H_q, \gamma, c) \cap (\bigcup_{\eta \in \mathcal{V}} \Pi_X(\varepsilon, \gamma, c, \langle q, w \rangle))$
\STATE $B_0 \leftarrow B_0 \cap \text{Concat}(\{\omega(\pi) \mid \pi \in C_q\})$
\STATE $\Gamma_\beta \leftarrow \varepsilon$
\STATE $\text{for } i = 1, \ldots, |G|^{2k} - 1 \do$
\STATE $C \leftarrow \varepsilon \cdot \varepsilon$
\STATE $\Gamma_\beta \leftarrow \Gamma_\beta \cdot C \cdot B_0$
\STATE $\Gamma_\beta \leftarrow \Gamma_\beta \cdot C \cdot B_0$
\STATE $\text{return } \Gamma_\beta$
\end{algorithmic}
\end{algorithm}
5.3 The General Case

Let us now turn to a more general case, in which the size of the strict letter-bounded expression $a_1^* \ldots a_d^*$ is not constant, i.e. $d$ is part of the input. The construction of the bounded control expression $\Gamma_b$ by Algorithm 2 (function LetterBoundedControlSet) follows the structure of the decomposition of control words defined at Section 5.1 which we briefly recall next. For every $k$-index depth-first derivation with control word $\gamma$, its productions can be rearranged into a $(k+1)$-index depth-first derivation, consisting of (i) a prefix $\gamma^1$ producing a word $a_1^* \ldots a_m^*$, then (ii) a pivot production $(X_i, w)$ followed by two words $\gamma'$ and $\gamma''$ such that: (iii) $\gamma'$ and $\gamma''$ produce words included in some bounded expression $a_m^* \ldots a_d^*$, respectively, where $\max(m - \ell, r - m) < d - 1$ (Lemma 9).

Algorithm 2 follows the aforementioned decomposition and builds the bounded expressions $\Gamma^x$, $(X_i, w)^*$, $\Gamma^r$ and $\Gamma^w$ with the goal of capturing $\gamma^1$, $(X_i, w)$, $\gamma$ and $\gamma''$, respectively. For all the control words such as $\gamma$, because $\gamma^2$ produces a word included in $a_m^* \ldots a_d^*$ the bounded expression $\Gamma^x$ is built calling ConstantBoundedControlSet (line 9). Whereas $\gamma'$ and $\gamma''$ produce words in a sub expression of $a_m^* \ldots a_d^*$ with as many as $d - 2$ letters. These cases are handled by recursively calling (lines 16 and 19) the function LetterBoundedControlSet. For each of the above calls, the appropriate grammar and bounded expression capturing its language are passed as arguments to the algorithm.

Because it needs to cover all the possible control words such as $\gamma$, the algorithm iterates (line 11) over all possible valid choices for the pivot production. Each choice for the pivot yields a bounded expression $[\Gamma^x \cdot (X_i, yz)^* \cdot \Gamma^r \cdot \Gamma^w]$ to capture the union of all these bounded expressions, they are concatenated (the ordering does not matter) into $\Gamma_b$ (line 21). The rationale is that $\tilde{b}_1 \cup \tilde{b}_2$ is included in the language of $\tilde{b}_1 \cdot \tilde{b}_2$.

Theorem 11. Given a grammar $G = \langle \Sigma, A, \Delta \rangle$, and a nonterminal $X \in \Sigma$, such that $L_X(G) \subseteq b$, where $b$ is a strict d-letter bounded expression over $A$, for each $k > 0$, there exists a bounded expression $\Gamma_b^k$ over $\Delta$ such that $L_X^{(k)}(G) \subseteq L_X(\Gamma_b \cap \Gamma^{df(k+1)}, G)$. Moreover, $\Gamma_b$ can be constructed in time $|G|^{O(k)+d}$.

Theorem 12 further generalizes from the strict letter-bounded case to the bounded case (using Lemma 9). This theorem gives the main result of the paper.

Theorem 12. Given $G = \langle \Sigma, \Sigma, \Delta \rangle$, a bounded expression $b = w_1^* \ldots w_d^*$, and $X \in \Sigma$ such that $L_X(G) \subseteq b$, for each $k > 0$ there exists a bounded expression $\Gamma_b^k$ over $\Delta$ such that $L_X^{(k)}(G) \subseteq L_X(\Gamma_b \cap \Gamma^{df(k+1)}, G)$. Moreover, $\Gamma_b$ can be constructed in time $(|b| \cdot |G|)^{O(k)+d}$.

We now show that the exponential growth of the size of the bounded control expression $\Gamma_b$ with respect to the maximal derivation index $k > 0$ is necessary. Given $k > 0$, consider the following grammar:

$G_k = \langle \{X_i \mid 0 \leq i \leq k\}, \{a\}, \{X_i \rightarrow X_{i-1} X_{i-1} \mid 1 \leq i \leq k\} \cup \{X_0 \rightarrow a\}\rangle$.

Notice that $L_{X_k}(G_k) = \{a^{2^k}\} \subseteq a^*$ and $|G_k| = O(k)$. Moreover, every depth-first derivation of $G_k$ has index $k + 1$.

Lemma 13. Every bounded expression $\Gamma_b$ with $L_{X_k}(\Gamma_b \cap \Gamma^{df(k)}, G) = L_{X_k}(G_k)$, is such that $|\Gamma_b| \geq 2^k$, for all $k > 0$.

Observe that, when ignoring the depth-first derivation policy for $G_k$, the control set $\Gamma_b = (X_k \rightarrow X_{k-1} X_{k-1} X_{k-1} \rightarrow X_{i-2} X_{i-2})^* \cdots (X_1 \rightarrow X_0 X_0)^*(X_0 \rightarrow a)^*$ captures the language, i.e. $L_{X_k}(\Gamma_b, G) = L_{X_k}(G)$. In this case, $\Gamma_b$ corresponds to a breadth-first traversal of the parse tree of $a^{2^k}$. Notice however that any derivation corresponding to such a traversal has index $2^{k+1}$.
Generating Bounded Languages Using Bounded Control Sets

Algorithm 2 Control Sets for Letter-Bounded Grammars

1: function LETTERBOUNDEDCONTROLSET(G_0, X, A, Δ_0, k)
2: \[ a_1^* \cdots a_d^* \leftarrow \text{MINIMIZEEXPRESSION}(G_0, X, a_1^* \cdots a_d^*) \] \[ \Leftrightarrow \{j_1, \ldots, j_d\} \subseteq \{i_1, \ldots, i_d\} \]
3: if [a_1^* \cdots a_d^*] \not\subseteq \{j_1, \ldots, j_d\} \not\subseteq \{i_1, \ldots, i_d\} \not\subseteq \{j_1, \ldots, j_d\} then
4: return \text{CONSTANTBOUNDEDCONTROLSET}(G_0, a_1^* \cdots a_d^*, k)
5: (Ξ_{j_1 \cdots j_d}, Ξ_{j_1 \cdots j_d}) \leftarrow \text{PARTITIONNONTERMINALS}(G_0, a_1^* \cdots a_d^*)
6: Δ^* \leftarrow \{ (X_j, w) \in Δ_0 \mid X_j \in Ξ_{j_1 \cdots j_d} \} \cup \{ (X_j, y, v) \in Δ_0 \mid X_j, y, v \in Ξ_{j_1 \cdots j_d} \}
7: G^* \leftarrow (Ξ, A, Δ^*)
8: Γ_k \leftarrow \text{CONSTANTBOUNDEDCONTROLSET}(G^*, a_1^* \cdots a_d^*, k + 1)
9: Γ_k \leftarrow \text{LETTERBOUNDEDCONTROLSET}(G_k, x_1 \cdots x_d, k)
10: \text{return } Γ_k

Related Work. Context-free languages and their generating process have been extensively studied throughout the ‘60s, ‘70s and ‘80s. For instance, Gruska [5] proved undecidable to determine, given context-free grammar \( G \), whether \( L(G) \) has bounded index, that is whether there exists a \( k \) such that \( L(G) = L^{(k)}(G) \). Luker [13] showed that every bounded context-free language has a finite index, whereas Salomaa [17] exhibits a language for which no finite index exists. Luker [13] investigated regularity questions around control sets. In his book [5], Ginsburg reviews his contributions with Spanier on bounded context-free languages including the decidability of checking whether \( L(G) \) is bounded for a given grammar \( G \), the construction of \( w_1, \ldots, w_d \) such that \( L(G) \subseteq w_1^* \cdots w_d^* \), and the structural properties of the semilinear sets arising from bounded context-free languages. The latter was instrumental in the work of Leroux et al [11].

References

1. A. Bouajjani, J. Esparza, A. Finkel, O. Maler, P. Rossmanith, B. Willems, and P. Wolper. An efficient automata approach to some problems on context-free grammars. Information Processing Letters, 74:221–227, 2000.
2. Ahmed Bouajjani and Michael Emmi. Analysis of recursively parallel programs. ACM Trans. Program. Lang. Syst., 35(3):10, 2013. doi:10.1145/2518188
Appendix

The appendix is divided in two parts. Appendix A contains the full development of Section 4 whose proofs are relatively easy and have been included for the sake of being self-contained. They are variations of classical constructions so as to take into account index and depth-first policy. Appendix B contains the rest of the proofs about the combinatorial properties of derivations.

A Proofs of Section 4

It is well-known that the intersection between a context-free and a regular language is context-free. Below we define the grammar that generates the intersection between the language of a given grammar $G = \langle \Xi, \Sigma, \Delta \rangle$ and a regular language given by a bounded expression
b = w_1^* \ldots w_d^* over Σ where ℓ_i denotes the length of each w_i. Let \( G^b = \langle Ξ^b, Σ, Δ^b \rangle \) be the grammar generating the regular language of \( b \), where:

\[
\begin{align*}
\Xi^b &= \{ Q_s^{(s)} \mid 1 \leq s \leq d \land 1 \leq r \leq ℓ_s \} \\
\Delta^b &= \{ Q_s^{(s)} \to (w_s), Q_{l+1}^{(s)} \mid 1 \leq s \leq d \land 1 < l \leq ℓ_s \} \cup \\
& \quad \{ Q_s^{(s)} \to (w_l) Q_{l'}^{(s')} \mid 1 \leq s \leq s' \leq d \} \cup \\
& \quad \{ Q_1^{(s)} \to ε \mid 1 \leq s \leq d \}.
\end{align*}
\]

It is routine to check that \( \{ w \mid Q_i^{(s)} \Rightarrow^* w \text{ for some } 1 \leq i \leq d \} = L(b) \). Moreover, notice that the number of nonterminals in \( G^b \) equals the size of \( b \), i.e. \( |Ξ^b| = |b| \).

> Remark. Note that when \( b \) is letter-bounded (\( b = a_1^* \ldots a_d^* \)), the grammar \( G_1^b = (Ξ_1^b, Σ, Δ_1^b) \) generating is given by:

\[
\begin{align*}
\Xi_1^b &= \{ Q_s^{(s)} \mid 1 \leq s \leq d \} \cup \{ Q_{sink} \} \\
\Delta_1^b &= \{ Q_s^{(s)} \to a_{s'} Q_{s'}^{(s')} \mid 1 \leq s \leq s' \leq d \} \cup \\
& \quad \{ Q_s^{(s)} \to b Q_{sink} \mid b \in Σ \{ a_s, a_{s+1}, \ldots, a_d \} \} \cup \\
& \quad \{ Q_s^{(s)} \to ε \mid 1 \leq s \leq d \} \cup \\
& \quad \{ Q_{sink} \to b Q_{sink} \mid b \in Σ \}.
\end{align*}
\]

is such that \( L_{q(1)}(G_1^b) = L(b) \). Furthermore, \( G_1^b \) is complete – all terminals can be produced from all nonterminals – and it is deterministic when \( b \) is strict. Then a grammar \( G_1^b \), such that \( L_{q(1)}(G_1^b) = Σ^* \setminus L(b) \), can be computed in time \( O(\|G_1^b\|) \), by replacing each production \( Q_s^{(s)} \to ε \), 1 ≤ s ≤ d, with \( Q_{sink} \to ε \).

Given \( G^b \), and a grammar \( G = (Ξ, Σ, Δ) \) in 2NF (by Assumption [5]) and \( X ∈ Ξ \), our goal is to define a grammar \( G^\sim = (Ξ^\sim, Σ, Δ^\sim) \) that produces the language \( L_X(G) \cap L(b) \), for some \( X ∈ Ξ \). The definition of \( G^\sim = (Ξ^\sim, Σ, Δ^\sim) \) follows:

\[
\begin{align*}
Ξ^\sim &= \{ [Q_s^{(r)} X Q_v^{(u)}] \mid X ∈ Ξ \land Q_s^{(r)} ∈ Ξ^b \land Q_v^{(u)} ∈ Ξ^b \land r \leq u \} \\
Δ^\sim &= \text{defined as follows:} \quad \\
& \quad \text{for every production } X \to w ∈ Δ \text{ where } w ∈ Σ^*, Δ^\sim \text{ has a production} \\
& \quad \quad [Q_s^{(r)} X Q_v^{(u)}] \to w \quad \text{if } Q_s^{(r)} \Rightarrow^* w Q_v^{(u)} \quad \text{(1)} \\
& \quad \text{for every production } X \to Y ∈ Δ, \text{ where } Y ∈ Ξ, Δ^\sim \text{ has a production} \\
& \quad \quad [Q_s^{(r)} X Q_v^{(u)}] \to [Q_s^{(r)} Y Q_v^{(u)}] \quad \text{(2)} \\
& \quad \text{for every production } X \to a Y ∈ Δ, \text{ where } a ∈ Σ \land Y ∈ Ξ, Δ^\sim \text{ has a production} \\
& \quad \quad [Q_s^{(r)} X Q_v^{(u)}] \to a [Q_y^{(z)} Y Q_v^{(u)}] \quad \text{if } Q_s^{(r)} \to a Q_y^{(z)} \in Δ^b \quad \text{(3)} \\
& \quad \text{for every production } X \to Y a ∈ Δ, \text{ where } Y ∈ Ξ \land a ∈ Σ, Δ^\sim \text{ has a production} \\
& \quad \quad [Q_s^{(r)} X Q_v^{(u)}] \to [Q_s^{(r)} Y Q_y^{(z)}] a \quad \text{if } Q_y^{(z)} \to a Q_v^{(u)} \in Δ^b \quad \text{(4)} \\
& \quad \text{for every production } X \to Y Z ∈ Δ, Δ^\sim \text{ has a production} \\
& \quad \quad [Q_s^{(r)} X Q_v^{(u)}] \to [Q_s^{(r)} Y Q_y^{(z)}] [Q_y^{(z)} Z Q_v^{(u)}] \quad \text{(5)}
\end{align*}
\]
Consequently, we have Lemma 14.

Proof. Let \( \gamma : \Xi \to \Xi \) be the function that “strips” every nonterminal \( [Q_s(XQ_v^u)] \in \Xi \) of the nonterminals from \( \Xi \), i.e. \( \gamma([Q_s(XQ_v^u)]) = X \). In the following, we abuse notation and extend the \( \gamma \) function to symbols from \( \Sigma \cup \Xi \), by defining \( \gamma(a) = a \), for each \( a \in \Sigma \), and further to words \( w \in (\Sigma \cup \Xi)^* \) as \( \gamma(w) = \gamma((w)_1) \cdots \gamma((w)_{|w|}) \). Finally, for a production \( p = (X, w) \in \Delta^\circ \), we define \( \zeta(p) = (\zeta(X), \zeta(w)) \), and for a control word \( \gamma \in (\Delta^\circ)^* \), we write \( \zeta(\gamma) \).

Lemma 14. Given a grammar \( G = (\Xi, \Sigma, \Delta) \) and a grammar \( G^b = (\Xi, \Sigma, \Delta^b) \) generating \( b, \) for every \( X \in \Xi, Q_s(XQ_v^u) \in \Xi^b, w \in \Sigma^* \), and every \( k > 0 \), we have:

\[
\begin{align*}
(i) & \text{ for every } \gamma \in (\Delta^\circ)^*, [Q_s(XQ_v^u)] \xrightarrow{\gamma} w \text{ only if } X \xrightarrow{\gamma} w \text{ and } Q_s \xrightarrow{\gamma} w Q_v^u; \\
(ii) & \text{ for every } \delta \in \Delta^*, X \xrightarrow{\delta} w \text{ and } Q_s \xrightarrow{\delta} w Q_v^u \text{ only if } [Q_s(XQ_v^u)] \xrightarrow{\delta} w, \text{ for some } \gamma \in (\Xi^\circ)^\gamma.
\end{align*}
\]

Consequently, we have \( \bigcup_{i=1}^{t} L_{[Q_s(XQ_v^u)]}(\Delta^\circ) = L_X \cap L_b \).

Proof. By induction on \( |\gamma| > 0 \). For the base case \( |\gamma| = 1 \) is the production \( ([Q_s(XQ_v^u)] \to w) \in \Delta^\circ \) with \( w \in \Sigma^* \) — by case (1) of the definition of \( \Delta^\circ \), we have \( Q_s \xrightarrow{\gamma} w Q_v^u \) and there exists a production \( X \to w \in \Delta \). Since, moreover, \( \zeta([Q_s(XQ_v^u)] \to w) = (X \to w) \), we have that \( X \xrightarrow{\gamma} w \) in \( G \).

For the induction step \( |\gamma| > 1 \), we have \( \gamma = ([Q_s(XQ_v^u)] \to \tau) \cdot \gamma' \), for some production \( [Q_s(XQ_v^u)] \to \tau \in \Delta^\circ \), and a word \( \tau \in (\Sigma \cup \Xi)^\tau \) of length \( |\tau| \leq 2 \). We distinguish four cases, based on the structure of \( \tau \\
1. \text{ if } \tau = [Q_s(YQ_v^u)] \text{ then } \tau \xrightarrow{\gamma} w \text{ is a derivation of } \Delta^\circ. \text{ By the induction hypothesis, we obtain that } Q_s \xrightarrow{\gamma} w Q_v^u \text{ and } Y \xrightarrow{\gamma} w \text{ is a derivation of } G. \text{ But } X \to Y \in \Delta \) — case \( 0 \).
2. \text{ if } \tau = a [Q_s(x)YQ_v^u] \text{ then } w = a \cdot w' \text{ and } G \text{ has derivation } [Q_s(x)YQ_v^u] \xrightarrow{\gamma} w' \text{. By the induction hypothesis, we obtain } Q_s \xrightarrow{\gamma} w' Q_v^u \text{ and } G \text{ has derivation } Y \xrightarrow{\gamma} w'. \text{ By the case } \( \) of the definition of \( \Delta^\circ \), we have } Q_s \xrightarrow{\gamma} a Q_s \xrightarrow{\gamma} \text{ and } \zeta([Q_s(XQ_v^u)] \to \tau) = (X \to a Y) \in \Delta. \text{ Thus } Q_s \xrightarrow{\gamma} w' Q_v^u \text{ and } X \xrightarrow{\gamma} w, \text{ where } \zeta(\gamma) = (X \to a Y) \cdot \zeta(\gamma').
3. \text{ the case } \tau = [Q_s(x)YQ_v^u] \text{ is symmetric, using the case } \( \) of the definition of } \Delta^\circ.
4. \text{ if } \tau = [Q_s(x)YQ_v^u][Q_s(x)ZQ_v^u] \text{ then, by Lemma 18 there exist words } w_1, w_2 \in \Sigma^* \text{ such that } w = w_1 w_2 \text{ and one of the following applies:}
\begin{align*}
a. & [Q_s(x)YQ_v^u] \xrightarrow{\gamma} w_1, [Q_s(x)ZQ_v^u] \xrightarrow{\gamma} w_2 \text{ and } \gamma = \gamma_1 \gamma_2, \text{ or } \\
b. & [Q_s(x)YQ_v^u] \xrightarrow{\gamma} w_1, [Q_s(x)ZQ_v^u] \xrightarrow{\gamma} w_2 \text{ and } \gamma = \gamma_2 \gamma_1.
\end{align*}
\text{ We consider the first case only, the second being symmetric. Since } \gamma_1 \leq |\gamma| \text{ and } \gamma_2 \leq |\gamma|, \\text{ we apply the induction hypothesis and find out that } Q_s \xrightarrow{\gamma} w_1 Q_s \text{, } Q_s \xrightarrow{\gamma} Q_s \text{, } Q_s \xrightarrow{\gamma} Q_s \text{, } Q_s \xrightarrow{\gamma} Q_s \text{, and } G \text{ has derivations } Y \xrightarrow{\gamma} w_1 Q_s \text{ and } Z \xrightarrow{\gamma} w_2 Q_s \text{. Then } Q_s \xrightarrow{\gamma} w_1 w_2 Q_v^u \text{ where } w_1 w_2 = w \text{. By case } \( \) of the definition of } \Delta^\circ, \Delta \text{ has a production } (X \to Y Z) =
\[ \zeta([Q^{(r)}_x X Q^{(w)}_y]) \rightarrow \tau. \]

If \( \gamma' = \gamma_1 \gamma_2 \), then \( \zeta(\gamma) = (X \rightarrow Y Z) \cdot \zeta(\gamma_1) \cdot \zeta(\gamma_2) \), and \( G \) has a 
k-index depth-first derivation \( X \xrightarrow{\zeta(\gamma)} w. \)

\( \boxed{4} \) By induction on \(|\delta| > 0\). For the base case \(|\delta| = 1\), we have \( \delta = (X \rightarrow w) \in \Delta \). By the case \( \boxed{1} \) from the definition of \( \Delta^\circ \), \( G^\circ \) has a rule \( [Q^{(s)}_y X Q^{(w)}_y] \rightarrow w \) and, since, moreover,
\[ \zeta([Q^{(r)}_x X Q^{(w)}_y]) \rightarrow w = \delta, \]
we have \( \gamma = ([Q^{(s)}_x X Q^{(w)}_y] \rightarrow w). \)

For the induction step \(|\delta| > 1\), we have \( \delta = (X \rightarrow \tau) \cdot \delta' \). We distinguish four cases, based on the structure of \( \tau \):

1. if \( \tau = Y \), for some \( Y \in \Xi \), by the induction hypothesis, \( G^\circ \) has a derivation \( [Q^{(s)}_x Y Q^{(w)}_y] \xrightarrow{\gamma'} \xrightarrow{\delta} w \), for some \( \gamma' \in \zeta^{-1}(\delta') \). Since \( Q^{(r)}_s \Rightarrow^{*} w Q^{(u)}_y \) — by case \( \boxed{2} \) of the definition of \( \Delta^\circ - G^\circ \) has a production \( p = ([Q^{(s)}_y X Q^{(w)}_y] \rightarrow [Q^{(s)}_x Y Q^{(w)}_y]). \) We define \( \gamma = p \cdot \gamma' \). It is immediate to check that \( \zeta(\gamma) = \delta \).

2. if \( \tau = a Y \), for some \( a \in \Sigma \) and \( Y \in \Xi \), then \( w = a \cdot w' \). Hence \( Q^{(r)}_s \Rightarrow^{*} a Q^{(y)}_y \), \( Q^{(s)}_y \Rightarrow^{*} a Q^{(x)}_y w' Q^{(y)}_y \) and \( G \) has a derivation \( Y \xrightarrow{\delta'} w' \). By the induction hypothesis, \( G^\circ \) has a derivation \( [Q^{(s)}_y Y Q^{(w)}_y] \xrightarrow{\gamma'} \xrightarrow{\delta} w', \) for some \( \gamma' \in \zeta^{-1}(\delta') \). By the case \( \boxed{3} \) of the definition of \( \Delta^\circ \), there exists a production \( p = ([Q^{(s)}_y X Q^{(w)}_y] \rightarrow a Y) \in \Delta^\circ \). We define \( \gamma = p \cdot \gamma' \). It is immediate to check that \( \zeta(\gamma) = \delta \), hence \( [Q^{(s)}_x X Q^{(w)}_y] \xrightarrow{\delta} w \).

3. the case \( \tau = Y a \), for some \( Y \in \Xi \) and \( a \in \Sigma \), is symmetrical.

4. if \( \tau = Y Z \), for some \( Y, Z \in \Xi \), then, by Lemma \( \boxed{18} \) there exist words \( w_1, w_2 \in \Sigma^* \) such that \( w = w_1 w_2 \) and either one of the following cases applies:

   a. \( Y \xrightarrow{\delta_1} w_1, Z \xrightarrow{\delta_2} w_2 \) and \( \delta' = \delta_1 \delta_2 \), or

   b. \( Y \xrightarrow{\delta_1} w_1, Z \xrightarrow{\delta_2} w_2 \) and \( \delta' = \delta_2 \delta_1 \).

Moreover, we have \( Q^{(r)}_s \Rightarrow^{*} w_1 Q^{(x)}_y \) and \( Q^{(s)}_y \Rightarrow^{*} w_2 Q^{(w)}_y \), for some \( Q^{(x)}_y \in \Xi^b \). We consider the first case only, the second being symmetric. Since \( |\delta_1| < |\delta| \) and \( |\delta_2| < |\delta| \) we apply the induction hypothesis and find two control words \( \gamma_1 \in \zeta^{-1}(\delta_1) \) and \( \gamma_2 \in \zeta^{-1}(\delta_2) \) such that \( G^\circ \) has derivations \( [Q^{(s)}_y Y Q^{(w)}_y] \xrightarrow{\gamma_1} w_1 \) and \( [Q^{(s)}_y Z Q^{(w)}_y] \xrightarrow{\gamma_2} w_2 \). By case \( \boxed{4} \) of the definition of \( \Delta^\circ \), \( G^\circ \) has a production \( p = ([Q^{(r)}_s X Q^{(w)}_y] \rightarrow [Q^{(s)}_x Y Q^{(w)}_y]) \rightarrow [Q^{(s)}_x Z Q^{(w)}_y]). \) If \( \delta' = \delta_1 \delta_2 \), we define \( \gamma = p \gamma_1 \gamma_2 \). It is immediate to check that \( \zeta(\gamma) = \delta \) and \( [Q^{(r)}_x X Q^{(w)}_y] \xrightarrow{\delta} w \).

In the rest of this section, for a given bounded expression \( b = w_1^a \ldots w_s^a \) over \( \Sigma \), we associate the strict \( d \)-letter-bounded expression \( \widehat{b} = a_1^a \ldots a_d^a \) over an alphabet \( \mathcal{A} \), disjoint from \( \Sigma \), i.e. \( \mathcal{A} \cap \Sigma = \emptyset \), and a homomorphism \( h : \mathcal{A} \rightarrow \Sigma^* \) mapping as follows: \( a_i \mapsto w_i \), for all \( 1 \leq i \leq d \). The next step is to define a grammar \( G^\circ = (\Sigma^\circ, \mathcal{A}, \Delta^\circ) \), such that, for all \( X \in \Xi, 1 \leq s \leq x \leq d \):

\[ h^{-1}(L_{[Q^{(r)}_x X Q^{(w)}_y]}(G^\circ)) \cap \widehat{b} = L_{[Q^{(r)}_x X Q^{(w)}_y]}(G^\circ). \]

The grammar \( G^\circ \) is defined from \( G^\circ \), by the following modification of the productions from \( \Delta^\circ \), defined by a function \( \iota : \Delta^\circ \rightarrow \Delta^\circ \):

1. if \( |w| = 0 \) then \( z = \varepsilon \).

2. if \( |w| = 1 \) then we have \( Q^{(r)}_s \Rightarrow^{*} w Q^{(w)}_y \) and we let \( z = a_r \) if \( v = 1 \) else \( z = \varepsilon \).
3. if \(|w| = 2\) then we have \(Q^{(r)}_x \Rightarrow_{C^b} (w)_1 Q^{(y)}_x \Rightarrow_{C^b} (w)_1 (w)_2 Q^{(w)}_x\) for some \(x, y\). Define the word \(z = z' \cdot z''\) of length at most 2 such that \(z' = a_r\) if \(x = 1\); else \(z' = \varepsilon\) and \(z'' = a_y\) if \(y = 1\) else \(z'' = \varepsilon\).

\[ \iota\left(\left[Q^{(r)}_x X^{(u)}_y\right]\right) \rightarrow b\left[Q^{(y)}_x Y^{(w)}_y\right]\] \(\iota\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow c\left[Q^{(y)}_x Y^{(w)}_y\right] \) where \(c = a_r\) if \(y = 1\); else \(c = \varepsilon\).

\[ \iota\left(\left[Q^{(r)}_x X^{(u)}_y\right]\right) \rightarrow \left[Q^{(r)}_x X^{(u)}_y\right] b\left[Q^{(r)}_x Y^{(w)}_y\right] \rightarrow \left[Q^{(r)}_x Y^{(w)}_y\right] c\] where \(c = a_x\) if \(v = 1\); else \(c = \varepsilon\).

\[ \iota\left(p\right) = p \text{ otherwise.} \]

Let \(\Delta^\infty = \{\iota\left(p\right) \mid p \in \Delta^\gamma\}\). In addition, for every control word \(\gamma \in (\Delta^\gamma)^*\) of length \(n\), let \(\iota\left(\gamma\right) = \iota\left(\gamma_1\right) \cdots \iota\left(\gamma_n\right) \in \Delta^\infty\). A consequence of the following proposition is that the inverse relation \(\iota^{-1} \subseteq \Delta^\infty \times \Delta^\infty\) is a total function.

**Proposition 15.** For each production \(p \in \Delta^\infty\), the set \(\iota^{-1}(p)\) is a singleton.

**Proof.** By case split, based on the type of the production \(p \in \Delta^\infty\). Since \(G^\infty\) is in 2NF:

\[ \text{if } p = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow a\right) \text{ then } \iota^{-1}(p) = \left\{\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow w\right\} \text{, where } Q^{(r)}_x \Rightarrow_{C^b} w Q^{(u)}_x \]

is the shortest step sequence of \(G^b\) between \(Q^{(r)}_x\) and \(Q^{(u)}_x\) which is unique by \(G^b\) and produces \(w \in \Sigma^*\).

\[ \text{if } p = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow b\left[Q^{(y)}_x X^{(u)}_y\right]\right) \text{ then either one of the cases below must hold:} \]

\(i)\ Q^{(r)}_x = Q^{(r)}_u \quad \text{and} \quad Q^{(u)}_x \Rightarrow_{C^b} b Q^{(y)}_x\), for some \(y \neq 1\). In this case \(b\) is uniquely determined by \(Q^{(r)}_x\) and \(Q^{(u)}_x\), thus we get \(\iota^{-1}(p) = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow b\left[Q^{(y)}_x X^{(u)}_y\right]\right)\).

\(i)\ Q^{(r)}_x = Q^{(r)}_y \quad \text{and} \quad Q^{(u)}_x \Rightarrow_{C^b} b Q^{(u)}_x\), for some \(t \neq \varepsilon\). In this case we get, symmetrically,

\[ \iota^{-1}(p) = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow b\left[Q^{(r)}_y Y^{(w)}_y\right] \right) \quad \text{b}\]

\(i)\ Q^{(r)}_x = Q^{(r)}_u \quad \text{and} \quad Q^{(u)}_x = Q^{(u)}_y. \text{ Then } \iota^{-1}(p) = \{p\}.

\[ \text{if } p = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow a_r\left[Q^{(y)}_x X^{(u)}_y\right]\right) \text{ for some } a_r \in A, \text{ hence } y = 1 \text{ (respectively,} \quad \left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow \left[Q^{(r)}_x Y^{(w)}_y\right] a_r \text{, hence } v = 1\right) \text{ and then the only possibility is } \iota^{-1}(p) = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow \left(w_r\right)_r\left[Q^{(r)}_y Y^{(w)}_y\right] \right) \text{ (respectively,} \quad \left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow \left[Q^{(r)}_y Y^{(w)}_y\right] \left(w_r\right)_r\)).

\[ \text{if } p = \left(\left[Q^{(r)}_x X^{(u)}_y\right] \rightarrow \left[Q^{(r)}_x Y^{(w)}_y\right] \left[w\right] \right) \text{ then } \iota^{-1}(p) = \{p\}. \]

\[ \bullet \]

**A.1 Proof of Lemma 7**

**Proof.** Let \(G^\infty = (\Xi^\infty, A, \Delta^\infty)\), where \(\Xi^\infty = \Xi^\gamma\), and let \(\xi : \Delta^\infty \rightarrow \Delta\) be the homomorphism defined as \(\xi = \zeta \circ \iota^{-1}\). Also let \(V_X = \{\left[Q^{(s)}_1 X^{(s)}_1\right] \mid 1 \leq s \leq d\}\), for all \(X \in \Xi\). We start by proving the following facts:

**Fact 16.** For all \(X \in \Xi\) and \(1 \leq s \leq x \leq d\), we have \(L_{\left[Q^{(s)}_1 X^{(s)}_1\right]}(G^\infty) \subseteq \bar{\mathcal{B}}\).

**Proof.** Let \(\bar{w} \in L_{\left[Q^{(s)}_1 X^{(s)}_1\right]}(G^\infty)\). We have \([Q^{(s)}_1 X^{(s)}_1] \Rightarrow \bar{w}\) is a derivation of \(G^\infty\) for some control word \(\gamma\) over \(\Delta^\infty\). By contradiction, assume \(\bar{w} \notin \bar{\mathcal{B}}\), that is there exist \(p, p'\) such that \(p < p'\) and \((\bar{w})_p = a_j\) and \((\bar{w})_{p'} = a_i\) with \(i < j\). The definition of \(\iota\) shows that there exists \(w \in L_{\left[Q^{(s)}_1 X^{(s)}_1\right]}(G^\infty)\) such that \([Q^{(s)}_1 X^{(s)}_1] \Rightarrow_{\xi^{-1}(\gamma)} w\) in \(G^\infty\), hence that \(w \in \mathcal{B}\) since \(L_{\left[Q^{(s)}_1 X^{(s)}_1\right]}(G^\infty) \subseteq \mathcal{B}\), and finally that \(Q^{(s)}_1 \Rightarrow_{C^b} w Q^{(s)}_1\). Now, the mapping \(\iota\) is defined such that a production in its image produces a \(a_r\) when, in the underlying \(G^b\), either control moves forward from \(Q^{(r)}_x\) to \(Q^{(s)}_1\), e.g. \([Q^{(r)}_x X^{(u)}_y] \rightarrow a_r\left[Q^{(s)}_1 Y^{(w)}_y\right]\) or control moves backward form \(Q^{(s)}_1\) to \(Q^{(r)}_x\), e.g. \([Q^{(s)}_1 X^{(u)}_y] \rightarrow \left[Q^{(s)}_1 Y^{(w)}_y\right] a_r\). Therefore, by the previous assumption on \(\bar{w}\) where \(a_j\) occurs before \(a_i\), we have that a production of \(Q^{(j)}_1 \rightarrow (w_j)_r Q^{(j)}_1\) for some \(u \geq j\)
and then a production of $Q_1^{(i)} \rightarrow (w_i)_i Q_1^{(u')}$ for some $u' \geq i$ necessarily occurs in that order in $\tau^{-1}(\gamma)$. But this is a contradiction because $j > i$ and the definition of $GB$ prohibits control to move from $Q_p^{(j)}$ to $Q_p^{(i)}$ for any $p_i, p_j$.

**Fact 17.** For all $X \in \Xi$, $1 \leq s \leq x \leq d$, $\gamma \in (\Delta^\circ)^*$, $k > 0$ and $i_1, \ldots, i_d \in \mathbb{N}$:

$$[Q_1^{(s)} X Q_1^{(x)}] \xrightarrow{\gamma_{df(k)}} w_1^{i_1} \ldots w_d^{i_d} \text{ in } G^\circ \text{ if, and only if, } [Q_1^{(s)} X Q_1^{(x)}] \xrightarrow{\gamma_{df(k)}} a_1^{i_1} \ldots a_d^{i_d} \text{ in } G^\circ \text{.}$$

**Proof.** By induction on $|\gamma| > 0$, and a case analysis on the right-hand side of the first production in $\gamma$.

1. “$\subseteq$” Let $\tilde{w} \in L_Y^k(G^\circ)$. By Fact 16 we have that $\tilde{w} \in \hat{\mathbf{b}}$. It remains to show that $\tilde{w} \in h^{-1}(L_X(G))$, i.e. that $h(\tilde{w}) \in L_X(G)$. But this is an immediate consequence of Fact 17. “$\supseteq$” Let $w \in h^{-1}(L_X^k(G))$ be a word. Then $h(w) \in L_X^k(G) \cap \mathbf{b} = \bigcup_{Y \in \mathcal{X}} L_Y(G^\circ)$, by Lemma 14. By Fact 17 we have that $\tilde{w} \in L_Y(G^\circ)$.

2. Let $w = w_1^{i_1} \ldots w_n^{i_n} \in L_X^k(G) \cap \mathbf{b}$ be a word. Then $G$ has a derivation $X \xrightarrow{\gamma_{df(k)}} w$, and by Lemma 14, there exists $Y \in V_X$ such that $G^\circ$ has a derivation $Y \xrightarrow{\gamma_{df(k)}} w_1^{i_1} \ldots w_n^{i_n}$. By Fact 17, $G^\circ$ has a derivation $Y \xrightarrow{\gamma_{df(k)}} a_1^{i_1} \ldots a_n^{i_n}$. By the hypothesis $L_X^k(G^\circ) \subseteq \hat{L}_Y(\Gamma, G^\circ)$, there exists a control word $\gamma \in \Gamma$ such that $Y \xrightarrow{\gamma_{df(k)}} a_1^{i_1} \ldots a_n^{i_n}$ in $G^\circ$, and by Fact 17, we have $Y \xrightarrow{\gamma_{df(k)}} w_1^{i_1} \ldots w_n^{i_n}$ in $G^\circ$. But then, by Lemma 14, $X \xrightarrow{\gamma_{df(k)}} w$ is a derivation of $G$, and moreover, $\zeta(\tau^{-1}(\gamma)) = \xi(\gamma) \in \xi(\Gamma)$.

3. Given that each production $p^\circ \in \Delta^\circ$ is the image of a production $p \in \Delta^\circ$ via $\tau$, we have $|p^\circ| = |\tau(p)| \leq |p|$. Hence $|G^\circ| \leq |G|$. Now, each production $p^\circ \in \Delta^\circ$ corresponds to a production $p$ of $G$, such that the nonterminals occurring on both sides of $p$ are decorated with at most 3 nonterminals from $\Xi^b$. Since $|\Xi^b| = |\mathbf{b}|$, we obtain that, for each production $p$ of $G$, $G^\circ$ has at most $|\mathbf{b}|^3$ productions of size $|p|$. Hence $|G^\circ| \leq |G^\circ| \leq |\mathbf{b}|^3 \cdot |G|$, and both $G^\circ$ and $\xi$ can be constructed in time $|\mathbf{b}|^3 \cdot |G|$.

**Remark.** Given $G = \langle \Xi, \Sigma, \Delta \rangle$, $X \in \Xi$, and a strict $d$-letter-bounded expression $\hat{\mathbf{b}} = a_1^* \ldots a_d^*$, the check $L_X(G) \subseteq \hat{\mathbf{b}}$ can be decided in time $O(|\hat{\mathbf{b}}| \cdot |G|)$, by building a grammar $G_P$ such that $L_P^d(\mathbf{b}) = \Sigma^d \cap L(\hat{\mathbf{b}})$ (see Remark A) and checking $L_X(G) \cap L_P^d(\mathbf{b}) = \emptyset$. A similar argument shows that queries $L_X(G) \cap (\Delta^* \cdot a_s \cdot \Delta^*) = \emptyset$, $1 \leq s \leq d$, can be answered in time $O(|G|)$ [5 Section 5].

## B Other proofs

**Lemma 18.** Given $G = \langle \Xi, \Sigma, \Delta \rangle$ and a $k$-index depth-first step sequence $XY \xrightarrow{\gamma_{df(k)}} w$, for two nonterminals $X, Y \in \Xi$, $w \in \Sigma^*$, and $\gamma \in \Delta^k$. There exist $w_1, w_2 \in \Sigma^*$ such that $w_1 w_2 = w$, and $\gamma_1, \gamma_2 \in \Delta^k$ such that either one of the following holds:

1. $X \xrightarrow{\gamma_{df(k-1)}} w_1$ and $Y \xrightarrow{\gamma_{df(k)}} w_2$ and $\gamma = \gamma_1 \gamma_2$, or
2. $X \xrightarrow{\gamma_1} w_1$ and $Y \xrightarrow{\gamma_2} w_2$ and $\gamma = \gamma_2 \gamma_1$.

**Proof.** The step sequence $XY \xrightarrow{\gamma} w$ has one of two possible forms, by the definition of a depth-first sequence:

- $XY \xrightarrow{\gamma_1} w_1$ $Y \xrightarrow{\gamma_2} w_2$, or
- $XY \xrightarrow{\gamma_2} X w_2 \xrightarrow{\gamma_1} w_1 w_2$,

for some words $w_1, w_2 \in \Sigma^*$ and control words $\gamma_1, \gamma_2 \in \Delta^*$. Let us consider the first case, the second being symmetric. Since $XY \xrightarrow{\gamma_1} w_1 Y$ is a $k$-index step sequence, the sequence $X \xrightarrow{\gamma} w_1$ obtained by erasing the $Y$ nonterminal from the right in all steps of the sequence, is of index $k - 1$, i.e. $X \xrightarrow{\gamma_1} w_1$. Also, since $w_1 Y \xrightarrow{\gamma_2} w_1 w_2$, we obtain $Y \xrightarrow{\gamma_2} w_2$, by erasing $w_1$ from the left, in all steps of the sequence. Clearly, in this case we have $\gamma = \gamma_1 \gamma_2$.  

### B.1 Proof of Lemma 4

**Proof.** $\Rightarrow$ We shall prove the following more general statement. Let $u^\alpha \xrightarrow{\gamma} w^\beta$ where $u^\alpha \in \Upsilon$ be a $k$-index depth-first $r$-step sequence. By induction on $|\gamma| \geq 0$, we show the existence of a path $[u^\alpha] \xrightarrow{\gamma} [w^\beta]$ in $A^{df(k)}$. For the base case $|\gamma| = 0$, we have $u^\alpha = w^\beta$ which yields $[u^\alpha] = [w^\beta]$ and since $u^\alpha \in \Upsilon$ the assumption shows that $u^\alpha, w^\beta \in \Upsilon(k)$, hence that $[u^\alpha], [w^\beta] \in [\Upsilon(k)]$ and we are done. For the induction step $|\gamma| > 0$, let $v^\eta \xrightarrow{p} w^\beta$ be the last step of the sequence, for some $p \in \Delta$, i.e. $\gamma = \sigma \cdot p$ with $\sigma \in \Delta^*$. By the induction hypothesis, $A^{df(k)}$ has a path $[u^\alpha] \xrightarrow{\gamma} [v^\eta]$. Since $[v^\eta], [w^\beta] \in [\Upsilon(k)]$, and $v^\eta \xrightarrow{p} w^\beta$, we have that $[v^\eta] \xrightarrow{p} [w^\beta]$ by definition of $\Rightarrow$, hence a path $[u^\alpha] \xrightarrow{\gamma} [w^\beta]$.

$\Leftarrow$ We prove a more general statement. Let $\bar{u} \xrightarrow{\gamma} \bar{w}$ be a path of $A^{df(k)}(\Gamma^r)$. We show by induction on $|\gamma|$ that there exist r-words $u^\alpha, w^\beta \in \Upsilon$ such that $[u^\alpha] = \bar{u}$, $[w^\beta] = \bar{w}$, and $u^\alpha \xrightarrow{\gamma} w^\alpha$. The case $|\gamma| = 0$ is trivial, because $\bar{u} = \bar{w}$ and since $\bar{u} \in [\Upsilon^r(k)]$ then there exists $u^\alpha \in \Upsilon$ such that $[u^\alpha] = \bar{u}$, and we are done. For the induction step $|\gamma| > 0$, let $\gamma = \sigma \cdot p$, for some production $p \in \Delta$ and $\sigma \in \Delta^*$. By the induction hypothesis, there exist r-words $u^\alpha, v^\eta \in \Upsilon^r(k)$ such that $\bar{u} = [u^\alpha] \xrightarrow{\gamma} [v^\eta] \xrightarrow{p} \bar{w}$ is a path in $A^{df(k)}$, and $u^\alpha \xrightarrow{\sigma} v^\eta$ is a $k$-index $r$-step sequence. The definition of the edge relation in $A^{df(k)}$ and $[v^\eta] \xrightarrow{p} \bar{w}$ shows that $v^\eta \xrightarrow{p} w^\beta$ for some $w^\beta$ such that $[w^\beta] = \bar{w}$ which concludes the proof.  

### B.2 Proof of Lemma 9

**Proof.** We start by proving a series of five facts.

(i) First, no production of $G$ has the form $(Y, v)$, where $Y \in \Xi_{\geq 3}$ and $v$ contains a symbol of $\Xi_{\leq 2}$. By contradiction, assume such a production exists where $Z \in \Xi_{\leq 3}$ is a nonterminal occurring in $v$. Because $Z \in \Xi_{\leq 3}$, $a_1$ occurs in some word of $L_Z(G)$ and $a_d$ occurs in some word of $L_Z(G)$. On the other hand, we have that either no word of $L_Y(G)$ contains $a_1$ or no word of $L_Y(G)$ contains $a_d$, since $Y \in \Xi_{\leq 3}$. Because $G$ is reduced, we have $\{ u \mid Y \xrightarrow{v} *= u \} \neq \emptyset$. We reach a contradiction, since $\{ u \mid Y \xrightarrow{v} *= u \}$ contains a word in which $a_1$ occurs and a word in which $a_d$ occurs, because $Z$ occurs in $v$.  

(ii) Define $Q(u,v)$ to be the following proposition:

$$\{u' \in (\Sigma \cup \mathcal{A})^* \mid u \Rightarrow^* u'\} \subseteq \{(a_1) \cup \Xi_{-\alpha}\}^*$$

and

$$\{v' \in (\Sigma \cup \mathcal{A})^* \mid v \Rightarrow^* v'\} \subseteq \{(a_2) \cup \Xi_{-\alpha}\}^*$$

We show that $Q(u,v)$ holds if $X_i \Rightarrow^* u X_j v$ with $X_i, X_j \in \Xi_{-\alpha}$. By contradiction, assume that there exists $u'$ such that $u \Rightarrow^* u'$ and $u' \notin \{(a_1) \cup \Xi_{-\alpha}\}^*$ (a similar argument holds for $v$). Then either (a) $u'$ contains a symbol $a_\ell$, for $\ell > 1$ or (b) $u'$ contains a nonterminal $Y$ with $\ell = 1$. Because $G$ is reduced, we have $\{u' \mid u \Rightarrow^* u'\} \notin \mathcal{B}$. In either case (a) or (b), there exists a step sequence $u' \Rightarrow^* u_1 a_\ell u_2 \in \mathcal{A}^*$ such that $\ell > 1$. Since $X_j \in \Xi_{-\alpha}$, we have that $X_j \Rightarrow^* a_1 u_3 \in \mathcal{A}^*$, hence that $X_j \Rightarrow^* u_1 a_\ell u_2 a_1 u_3$ and finally that $L_X(G) \subseteq \mathcal{B}$, since $G$ is reduced, a contradiction.

(iii) For every step sequence $X_i \Rightarrow^* \alpha$, where $X_i \in \Xi_{-\alpha}$, $\alpha$ cannot be of the form $u_1 X_{\ell} u_2 X_{\ell} u_3$ where $X_{\ell}, X_{\ell} \in \Xi_{-\alpha}$. In fact, take the decomposition $u = u_1$ and $v = u_2 X_{\ell} u_3$ (the case $u = u_1 X_{\ell} u_2$ and $v = u_3$ yields the same result). Because (ii) applies, we find that $Q(u,v)$ holds but $v \notin \{(a_2) \cup \mathcal{A} \cup \Xi_{-\alpha}\}^*$, hence a contradiction.

(iv) If $X \Rightarrow_G u X_i v$ is a step sequence of $G$, for some $X_i \in \Xi_{-\alpha}$, then $X \Rightarrow_G u X_i v$ is also a step sequence of $G^2$. The proof goes by induction on $n = |\gamma|$. Let $X = (X_0 \Rightarrow (X_1)(X_2)\ldots (X_n)) \Rightarrow_{G^2} u w_{n-1}$ and conclude that $X \Rightarrow_X w_{n-1}$ where $p \neq \ell$ and $\gamma = (Y, t)$ with $Y \in \Xi_{-\alpha}$ for otherwise $X \Rightarrow^*_G w_{n-1}$ contradicts (iii) (recall that both $(u_{n-1})\ell$ and $X$ belong to $\Xi_{-\alpha}$). Thus we have $\gamma_n \in \Delta^\ast$, hence $w_{n-1} \Rightarrow_{G^2} w_n$, and finally $X \Rightarrow_{G^2} u X_i v$.

(v) If $L_1, L_2 \subseteq \mathcal{B}$ and $L_1 \cdot L_2 \subseteq a_{\alpha}^+ \ldots a_{\alpha}^+$, for some $1 \leq \ell < r \leq d$, then there exists $\ell \leq q \leq r$ such that $L_1 \subseteq a_{\alpha}^q \ldots a_{\alpha}^q$ and $L_2 \subseteq a_{\alpha}^q \ldots a_{\alpha}^q$. Assume, by contradiction, that there is no such $q$. Then there exist words $w_1 = a_{\ell}^q \ldots a_{\ell}^q \in L_1$ and $w_2 = a_{r}^q \ldots a_{r}^q \in L_2$, two positions $p_1, p_2$ such that $\ell \leq p_2 < p_1 \leq r$ such that $i_{p_1} \neq 0, j_{p_2} \neq 0$. Because all $a_i$ are distinct, we conclude that $w_1 \cdot w_2 \notin a_{\alpha}^q \ldots a_{\alpha}^q$, hence a contradiction.

We continue with the proof of the five items of the lemma:

1. The derivation $X \Rightarrow_{df(K)} w$, where $[\gamma] = n$, has a unique corresponding r-step sequence

$$X^{(0)} = w_{\alpha}^\gamma \Rightarrow_{df(K)} w_{\alpha}^{\gamma_1} \cdots w_{\alpha}^{\gamma_n} = w_{\alpha}^n.$$  

Now, we define a parent relationship in that step sequence, denoted $\prec$, between r-annotated nonterminals: $Y^{(\alpha)} \prec Z^{(b)}$, where necessarily $a < b$, if there exists a step in the sequence that rewrites $Y^{(\alpha)}$ to $Z^{(b)}$, that is

$$Y^{(\alpha)} \Rightarrow_{df(K)} Z^{(b)} \text{ where } (w^{(\alpha)})_j = Y^{(\alpha)}, \text{ and } (w^{(b)})_j = Z^{(b)} \text{ for some } j \leq \ell \leq j + 1 + |\ell|.$$  

Let $[\gamma]_\ell = (X_\ell, yz)$ be the last occurrence, in $\gamma$, of a production with head $X_\ell \in \Xi_{-\alpha}$. Notice that such an occurrence always exists since $X \in \Xi_{-\alpha}$ and moreover we have that $y, z \in \Xi_{-\alpha} \cup \{\mathcal{A}\} \cup \{\varepsilon\}$. In fact, since $\gamma$ is a derivation, if $y \in \Xi_{-\alpha}$ or $z \in \Xi_{-\alpha}$ then $[\gamma]_\ell$ would clearly not be the last such occurrence. Let $X = X^{(0)} \prec X^{(1)} \prec \cdots \prec X^{(p)}$ be the sequence of (r-annotated) ancestors of $X_\ell$ in the r-step sequence, and $[\gamma]_{\ell_j} = (X_{\ell_j}, y_{m_j} X_{\ell_{j+1}}) \in \Delta$ (or, symmetrically $[\gamma]_{\ell_j} = (X_{\ell_j}, y_{m_j} X_{\ell_{j+1}}) \in \Delta$), for some $y_{m_j} \in \Xi \cup \mathcal{A} \cup \{\varepsilon\}$, be the productions introducing these nonterminals, for all $0 \leq j < p.$
If \( y_{m_j} \in \Xi \), let \( \tau_j \) be the subword of \( \gamma \) corresponding to the derivation \( y_{m_j} \Rightarrow w_{m_j} \), for some \( w_{m_j} \in A^* \). Notice that here \( y_{m_j} \) is not an ancestor of \( X_{i_k} \), and that \( y_{m_j} \Rightarrow w_{m_j} \) must be a depth-first derivation because \( X \Rightarrow w \) is. Otherwise, if \( y_{m_j} \in A \), let \( \tau_j = \varepsilon \).

Let \( \gamma^2 = (\gamma)_{\ell_0} \cdot \tau_0 \cdot (\gamma)_{\ell_1} \cdot \tau_1 \cdots (\gamma)_{\ell_{p-1}} \cdot \tau_{p-1} \). Observe that, since each \( y_{m_j} \Rightarrow w_{m_j} \) is a depth-first derivation, we have \( X_{i_{j+1}} y_{m_j} \Rightarrow X_{i_{j+1}} \gamma^2 w_{m_j} \) (or with \( X_{i_{j+1}} \) and \( y_{m_j} \) swapped) is a depth-first step sequence because \( y_{m_j} \) and \( X_{i_{j+1}} \) have the same rank \( b \). Clearly, \( \gamma^2 \) corresponds to a valid step sequence of \( G \) which, moreover, is depth first, since whenever \( (\gamma)_{\ell_j} \) fires, \( X_{i_{j+1}} \) is the only nonterminal left (and whose rank is therefore maximal). It follows from \( \varepsilon \) that because \( X \Rightarrow_G u X_{i_p} \) holds and \( X, X_{i_p} \in \Xi_{\gamma_{-}^\delta} \) then \( X \Rightarrow_G u X_{i_p} \) holds (notice the use of \( G^\delta \) instead of \( G \)). Moreover, the definition of \( \gamma^2 \) shows that \( X \Rightarrow_G u X_{i_p} \) is a depth-first step sequence and \( u, v \in A^* \).

Since \( X \Rightarrow_G w \) is a \( k \)-index derivation, each step sequence \( y_{m_j} \Rightarrow w_{m_j} \) are of index at most \( k \). Therefore the index of each step sequence \( X_{i_{j+1}} y_{m_j} \Rightarrow X_{i_{j+1}} w_{m_j} \) (or in reverse order) is at most \( k + 1 \). Also, when each \( (\gamma)_{\ell_j} \) fires, \( X_{i_{j+1}} \) is the only nonterminal left and so the index of the step is at most \( 2 \). Therefore we find that \( X \Rightarrow_G u X_{i_p} \), and finally that \( X \Rightarrow_G u X_{i_p} \) in \( G^2 \).

2. Assume that \( y, z \in \Xi_{\gamma_{-}^\delta} \) (the cases \( y \in A \cup \{\varepsilon\} \) or \( z \in A \cup \{\varepsilon\} \) are similar). Since \( \gamma \) of length \( n \) induces a \( k \)-index depth-first derivation, we have that \( y z \Rightarrow_G u y u z \in A^* \) can be split into two derivations of \( G \) as follows: \( y \Rightarrow_G u y \) and \( z \Rightarrow_G u z \) such that \( \max(k_y, k_y) \leq k \) and \( \min(k_y, k_y) \leq k - 1 \) (see Lem. 18 for a proof). Assume \( k_y \leq k - 1 \), the other case being symmetric. Since the only production in \( (\gamma)_{\ell_k} \cdots (\gamma)_{\ell_{n}} \), whose left hand side is a nonterminal from \( \Xi_{\gamma_{-}^\delta} \) is \( (\gamma)_{\ell_k} = (X_{i_y}, y z) \), which, moreover, occurs only in the first position, we have that \( \gamma_y \in G \Rightarrow_G u \) is a depth-first derivation, and \( \gamma_y \in G^\delta \Rightarrow_G u \) by the definition of \( G_{i_y, y z} \).

3. It suffices to notice that \( \gamma^2 = (\gamma)_{\ell_0} \cdots (\gamma)_{\ell_{n}} \) resulting from reordering the productions of \( \gamma \) and that ordering the productions of \( \gamma \) result into a step sequence producing the same word \( w = \gamma_{\ell_0} \cdots \gamma_{\ell_{n}} \) where \( L_X(G) \subseteq \mathbb{b} \) where \( \mathbb{b} \) is a strict \( d \)-letter bounded expression. That the resulting derivation has index \( k \) and is depth-first follow easily from [1] and [2].

4. Given that \( \Delta^\delta \subseteq \Delta \) we find that \( X \Rightarrow_{G^\delta} u X_i v \) implies \( X \Rightarrow_{G} u X_i v \), hence \( Q(u, v) \) holds by \( \lceil \triangledown \rceil \) and \( X, X_i \in \Xi_{\gamma_{-}^\delta} \). By the definition of \( Q(u, v) \), we have:

\[
\begin{align*}
\{ u' \in (\Xi \cup A)^* \mid u \Rightarrow u' \} & \subseteq (\{ a_1 \} \cup \Xi_{\gamma_{-}^\delta})^* \\
\{ v' \in (\Xi \cup A)^* \mid v \Rightarrow v' \} & \subseteq (\{ a_d \} \cup \Xi_{\gamma_{-}^\delta})^*
\end{align*}
\]

Since \( G \) is reduced, \( \{ u' \in A^* \mid u \Rightarrow u' \} \neq \emptyset \) and \( \{ v' \in A^* \mid v \Rightarrow v' \} \neq \emptyset \). But because \( X_i \in (\Xi_{\gamma_{-}^\delta} \) it must be the case that \( \{ u' \in A^* \mid u \Rightarrow u' \} \subseteq a_1^* \) and \( \{ v' \in A^* \mid v \Rightarrow v' \} \subseteq a_d^* \), otherwise we would contradict the fact that \( L_X(G) \subseteq \mathbb{b} \).

5. Since \( X \Rightarrow_{G^\delta} u X_{i_y} v \) and \( \gamma \) is a depth-first derivation, we have that \( \{ u' \in A^* \mid u \Rightarrow_{G^\delta} u' \} \) and \( \{ v' \in A^* \mid v \Rightarrow_{G^\delta} v' \} \) are depth-first. Moreover, \( L_X(G^\delta) \subseteq \mathbb{b} \), and thus \( L_{X_l}(G) \cdot L_{X_l}(G^\delta) \subseteq \mathbb{b} \). We consider only the case \( y, z \in \Xi_{\gamma_{-}^\delta} \) and \( X, X_i \in A \cup \{\varepsilon\} \) or \( z \in A \cup \{\varepsilon\} \) use similar arguments, and are left as an easy exercise. Hence, our proof falls into 4 cases:

(a) \( L_y(G_{i, w}) \cap (a_1 \cdot A^*) = \emptyset \) and \( L_{X_l}(G_{i, w}) \cap (a_1 \cdot A^*) = \emptyset \). Thus \( L_y(G_{i, w}) \cdot L_{X_l}(G_{i, w}) \subseteq \emptyset \)
Generating Bounded Languages Using Bounded Control Sets

\[ a_2^* \ldots a_d^* \] Then fact (v) for \( \ell = 2 \) and \( r = d \) concludes this case.

\[(b)\] \( L_y(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \) and \( L_z(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \). Thus \( L_y(G_{i,w}) \cdot L_z(G_{i,w}) \subseteq a_1^* \ldots a_{d-1}^* \). Then fact (v) for \( \ell = 1 \) and \( r = d - 1 \) concludes this case.

\[(c)\] \( L_y(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \) and \( L_z(G_{i,w}) \cap (a_1 \cdot A^*) = \emptyset \). Thus we have \( L_y(G_{i,w}) \subseteq a_1^* \ldots a_{d-1}^* \) and \( L_z(G_{i,w}) \subseteq a_2^* \ldots a_d^* \). By the fact (v) (with \( \ell = d \), \( r = d \)) there exists \( q, 1 < q < d \) such that \( L_y(G_{i,w}) \subseteq a_1^* \ldots a_{q-1}^* \) and \( L_z(G_{i,w}) \subseteq a_q^* \ldots a_d^* \). Next we show 1 < \( q < d \) holds. In fact, assume the inclusions hold for \( q = 1 \). Then they also hold for \( q = 2 \) since \( L_z(G_{i,w}) \subseteq a_2^* \ldots a_d^* \). A similar reasoning holds when \( q = d \) since \( L_y(G_{i,w}) \subseteq a_1^* \ldots a_{d-1}^* \).

\[(d)\] \( L_y(G_{i,w}) \cap (a_1 \cdot A^*) = \emptyset \) and \( L_z(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \). We first observe that it cannot be the case that \( L_y(G_{i,w}) \) contains some word where \( a_d \) occurs and \( L_z(G_{i,w}) \) contains some word where \( a_1 \) occurs for otherwise concatenating those two words shows \( L_y(G_{i,w}) \cdot L_z(G_{i,w}) \not\subseteq a_1^* \ldots a_d^* \). This leaves us with three cases: (a) If \( L_y(G_{i,w}) \cap (A^* \cdot a_d) \not\subseteq \emptyset \) we find that \( L_y(G_{i,w}) \subseteq a_d^* \), hence that \( L_y(G_{i,w}) \subseteq a_2^* \ldots a_d^* \) since \( L_y(G_{i,w}) \cap (a_1 \cdot A^*) = \emptyset \). (b) If \( L_z(G_{i,w}) \cap (a_1 \cdot A^*) \not\subseteq \emptyset \) we find that \( L_y(G_{i,w}) \subseteq a_1^* \), hence that \( L_y(G_{i,w}) \subseteq a_1^* \ldots a_{d-1}^* \) since \( L_z(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \). (c) Then \( L_y(G_{i,w}) \cap (A^* \cdot a_d) = \emptyset \) and \( L_z(G_{i,w}) \cap (a_1 \cdot A^*) = \emptyset \). Hence \( L_y(G_{i,w}) \cdot L_z(G_{i,w}) \subseteq a_2^* \ldots a_{d-1}^* \) and by the fact (v) for \( \ell = 2 \) and \( r = d - 1 \) there exists 1 < \( q < d \) such that \( L_y(G_{i,w}) \subseteq a_2^* \ldots a_q^* \) and \( L_z(G_{i,w}) \subseteq a_q^* \ldots a_{d-1}^* \).

B.3 Proof of Lemma 10

When \( L_{X,Y}(G) \subseteq \tilde{b} \), because \( \tilde{b} = a_1^* \ldots a_d^* \) is a strict \( s \)-letter-bounded expression with \( s \) a fixed constant, for every step sequence \( X \xrightarrow{\gamma} u \ Y \ v \), we have \( u \ v = \gamma_{i_1} \ldots \gamma_{i_s} \). Also remark that \( u \ v = a_{j_1} \ldots a_{j_r} \) for some \( (j_1, \ldots, j_r) \in \mathbb{N}^s \), hence that \( j_\ell = |\gamma_{i_\ell}| \) for each \( \ell, 1 \leq \ell \leq s \).

For convenience, given \( \gamma \in \Delta^* \), we denote \( \gamma|_{i_\ell} = \gamma_{i_\ell} \ldots \gamma_{i_s} \).

We recall the definition of the labeled graph \( A^{df(k)} = [\hat{T}^{(k)}], \Delta \to \) whose number of vertices we denote by \( N \). Due to Assumption 5, we can safely restrict \( [\hat{T}^{(k)}] \) to \( r \)-words with at most 2 nonterminals having the same rank, hence \( N \leq |G|^{2k} \). We recall also that \( \Omega(q) \) is the set of elementary cycles that start and end in a vertex \( q \in \hat{T}^{(k)} \).

> Proposition 19. Let \( G = (\Xi, \Sigma, \Delta) \) be a grammar, \( X \in \Xi \) be a nonterminal and \( \tilde{b} = a_1^* \ldots a_d^* \) be a \( s \)-letter-bounded expression, for some \( s \geq 0 \). For any two vertices \( q, q' \in [\hat{T}^{(k)}] \) of \( A^{df(k)} \), and any path \( \pi \in \Pi(q, q') \) there exists a path \( \pi' \in \Pi(q, q') \) such that \( |\pi| = |\pi'| \), \( \omega(\pi)|_{i_\ell} = \omega(\pi')|_{i_\ell} \) and \( \pi' \) is of the form \( c_1 \cdot \theta_1 \cdot \ldots \cdot \theta_k \cdot \cdot \cdot c_{k+1} \cdot \theta_{k+1} \), where \( c_1 \in \Pi(q, q_{i_1}) \), \( \in \Pi(q_{i_k}, q'_{i_k}) \) and \( j_\ell \in \Pi(q_{i_{\ell-1}}, q_{i_{\ell}}) \), for each \( 1 < j \leq k \), are acyclic paths, \( \theta_1 \in (\Omega(q_{i_1}))^* \ldots \theta_k \in (\Omega(q_{i_k}))^* \) are cycles, and \( k \leq |Q| \).

Proof. The proof goes along the lines of that of Lemma 7.3.2 in [12]. This proof is carried on graphs labeled with integer tuples, and addition, instead of concatenation. Since the only property of integer tuple addition, used in the proof of [12] Lemma 7.3.2, is commutativity, it suffices to observe that \( \omega(\pi)|_{i_\ell} = \omega(\pi')|_{i_\ell} \) whenever \( \omega(\pi) \) is a permutation of \( \omega(\pi') \).

Proof of Lemma 10 Given two step sequences \( X \xrightarrow{\gamma} u \ Y \ v \), \( X \xrightarrow{\gamma'} u' \ Y \ v' \), the following are equivalent:

\[ |\gamma_{i_\ell}| = |\gamma'_{i_\ell}| \quad \text{for all } \ell, 1 \leq \ell \leq s, \]

\[ \gamma|_{i_\ell} = \gamma'|_{i_\ell} \]

\[ u \ v = u' \ v' \]
Since \( L_{X,Y}(G) \subseteq \tilde{b} \) where \( \tilde{b} \) is a strict s-letter bounded expression, for every \( \pi \in \Omega(q) \) the induced word \( a^1_1 \ldots a^k_d = \omega(\pi)\downarrow_{\tilde{b}} \) is such that: \( \sum_{j=1}^{\kappa} k_j \leq 2N \), i.e. since \( G \) is in 2NF (Assumption 5), each production in \( \Delta \) issues at most 2 symbols from \( \{a_1, \ldots, a_s\} \), and each elementary cycle is of length at most \( N \). The nonnegative solutions of the inequation \( \sum_{j=1}^{\kappa} k_j \leq 2N \) are solutions to the equation \( \sum_{j=1}^{\kappa} k_j + y = 2N \), for a nonnegative slack variable \( y \geq 0 \). Since the number of nonnegative solutions to the latter equation is \( (s+2N)/s \), we have:

\[
[\omega(\pi)\downarrow_{\tilde{b}} \mid \pi \in \Omega(q)] = \left( \frac{s + 2N}{s} \right) = O(N^s) .
\]

For each vertex \( q \), we are interested in a set \( C_q \subseteq \Omega(q) \) such that \( \|C_q\| = O(N^s) \) and, moreover, for each \( \pi \in \Omega(q) \) there exists \( \pi' \in C_q \) such that \( \omega(\pi)\downarrow_{\tilde{b}} = \omega(\pi')\downarrow_{\tilde{b}} \) when \( \Pi(X, q) \neq \emptyset \) and \( \Pi(q, Y, (0)) \neq \emptyset \) holds.

For now we assume we have computed such sets \( \{C_q\}_{q \in \Upsilon(i)} \) (their effective computation will be described later). We are now ready to define the bounded expression \( \Gamma_{b} \). Given a finite set \( \Gamma = \{\gamma_1, \ldots, \gamma_s\} \subseteq \Delta^* \) of control words index by some total ordering (e.g. we assume a total order \( < \) on \( \Xi \cup A \), and define \( (X_1, w_1) <_{\Delta} (X_2, w_2) \iff X_1 \cdot w_1 <_{\text{lex}} X_2 \cdot w_2 \) in the lexicographical extension of \( < \), then extend \( <_{\Delta} \) to a lexicographical order \( <_{\Delta} \) on control words), we define the bounded expression: \( \text{concat}(\Gamma) = \gamma_1^* \cdots \gamma_s^* \). Let \( \Upsilon(i) = \{q_1, \ldots, q_N\} \) be the set of vertices of \( A^{\text{df}(k)} \), taken in some order. We define the set \( \{B_i\}_{i \geq 0} \) of bounded expressions as follows:

\[
\begin{align*}
B_0 &= \text{concat}(\{\omega(\pi) \mid \pi \in C_{q_1}\}) \cdots \text{concat}(\{\omega(\pi) \mid \pi \in C_{q_N}\}) \\
B_1 &= \text{concat}(\Delta)^{N-1} : B_0 \cdot \text{concat}(\Delta)^{N-1} \\
B_i &= \text{concat}(\Delta)^{N-1} : B_0 \cdot B_{i-1}, \text{ for all } i \geq 2
\end{align*}
\]

Finally, let:

\[ \Gamma_{b} = B_N . \]

Let us now prove the language inclusion.

It is easy to show that \( L_{X,Y}(G) = \hat{L}_{X,Y}(\Gamma_{b}^{\text{df}(k)}), G \) for every \( X \in \Xi, Y \in \Xi \cup \{\varepsilon\} \) and \( k > 0 \).

Then it suffices to show the following: given a k-index depth first step sequence \( X \xrightarrow{\text{df}(k)} u Y v \), there exists a control word \( \gamma' \in \Gamma_{b} \) such that \( X \xrightarrow{\gamma'} u' Y v' \) and \( u v = u' v' \).

Because Lemma 4 shows that each path \( \pi \in \Pi(X, Y, (0)) \) corresponds to a control word \( \omega(\pi) \) such that \( X \xrightarrow{\omega(\pi)} u Y v \), and because \( L_{X,Y}(G) \subseteq \tilde{b} \) where \( \tilde{b} \) is a strict s-letter bounded expression, it suffices to show that exists a path \( \rho \in \Pi(X, Y, (0)) \) such that \( \omega(\rho) \in \Gamma_{b} \) and \( \omega(\pi)\downarrow_{\tilde{b}} = \omega(\rho)\downarrow_{\tilde{b}} \). We apply the result from Prop. 19 which shows that there exists a path \( \rho \in \Pi(X, Y, (0)) \), such that \( |\rho| = |\pi|, \omega(\rho)\downarrow_{\tilde{b}} = \omega(\pi)\downarrow_{\tilde{b}} \) and \( \rho \) is of the form \( \varsigma_1 \cdot \theta_1 \cdots \varsigma_k \cdot \theta_k \cdot \varsigma_{k+1} \), where \( \varsigma_1 \in \Pi(X, q_{i_1}), \varsigma_{k+1} \in \Pi(q_{i_k}, Y, (0)) \), and \( \varsigma_j \in \Pi(q_{i_{j-1}}, q_{i_j}) \) for each \( 1 < j \leq k \) are acyclic paths, \( \theta_1 \in (\Omega(q_{i_1}))^* \), \( \ldots, \theta_k \in (\Omega(q_{i_k}))^* \) are cycles, \( q_{i_1}, \ldots, q_{i_k} \) are vertices, and \( k \leq |Q| \). Hence we conclude that \( \omega(\varsigma_j) \in \text{concat}(\Delta^{N-1}) \), for all \( 1 \leq j \leq k + 1 \).

---

8 The number of nonnegative solutions of an equation \( n = x_1 + \ldots + x_m \) is \( \binom{m+n-1}{m-1} \).

9 See Lemma 2 [13]
for each cycle \( \theta_j \in (\Omega(q_i))^* \), consisting of a concatenation of several elementary cycles \( \theta_1^j, \ldots, \theta_{\ell_j}^j \in \Omega(q_i) \), the cycle \( \theta_j^{\text{lex}} \) obtained by a lexicographic reordering of \( \theta_1^j, \ldots, \theta_{\ell_j}^j \) (based on the lexicographic order of their value in \( \Delta^* \)) belongs to \( B_{\theta_j} \), for all \( 1 \leq j \leq k \).

Second, it is easy to see that the words produced by \( \theta_j \) and \( \theta_j^{\text{lex}} \) are the same, since the order of productions labeling \( \theta_j (\theta_j^{\text{lex}}) \) is not important.

Let \( \pi' \) be the path \( q_1 \cdot \theta_1^{\text{lex}} \cdot \ldots \cdot q_k \cdot \theta_k^{\text{lex}} \cdot q_{k+1} \). By Prop. 19 we have that \( \omega(\pi) \upharpoonright B_\pi = \omega(\pi') \upharpoonright B_\pi \).

Moreover, \( \omega(\pi') \in B_N = \Gamma_\pi \). Since \( X \stackrel{\omega(\pi')}{ \longrightarrow } Y \) and \( X \stackrel{\omega(\pi)}{ \longrightarrow } Y \) are step sequences of \( G \), the previous equality implies \( u v = u' v' \).

Concerning the time needed to construct the bounded expression \( \Gamma_\pi \), the main ingredient in the previous, is the definition of the sets of cycles \( \{C_q\}_{q \in \Pi(k)} \), such that \( \|C_q\| = \mathcal{O}(N^k) \) and, moreover, for each \( \pi \in \Omega(q) \) there exists \( \pi' \in C_q \) such that \( \omega(\pi) \upharpoonright B_\pi = \omega(\pi') \upharpoonright B_\pi \) when \( \Pi(X^{(0)}, q) \neq \emptyset \) and \( \Pi(q, Y^{(0)}) \neq \emptyset \) holds. Below we describe the construction of such sets.

Define \( \text{Val} = \{a_1^{\ell_1} \ldots a_{s}^{\ell_s} \in B \mid \sum_{i=1}^{s} k_i \leq 2N\} \). Using previous arguments (i.e. equation (3)), it is routine to check that \( \|\text{Val}\| = \mathcal{O}(N^s) \). Consider the labeled graph \( H = (V, \Delta, \rightarrow) \), defined upon \( A^{\text{df}(k)} \), where:

\[
V = \{Y(k)\} \times \text{Val}, \quad \text{and} \quad \langle q', a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \xrightarrow{(Z, z)} \langle q', a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \text{ if } q' \xrightarrow{(Z, z)} q' \text{ and } a_i^{\ell_i} = a_i^{\ell_i} \cdot z_1 a_1 \text{ for each } \ell.
\]

First, observe that the number of vertices in this graph is \( |V| \leq N^{2k} \cdot (2N^s)^{k} = |G|^{O(k)} \). Second, it is routine to check (by induction on the length of a path) that given a path \( \pi \in \Pi_N((q, \varepsilon), \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle) \) for some \( i_1, \ldots, i_s \in \mathbb{N} \) we have \( \omega(\pi) \upharpoonright B_\pi = a_{i_1} \ldots a_{i_s} \). Next, for each \( q \in \{Y(k)\} \) define the set \( \mathcal{P}_q \) of paths of \( H \) consisting for each \( a_1^{\ell_1} \ldots a_{s}^{\ell_s} \in \text{Val} \) of a single path (one with the least number of edges) from \( \langle q, \varepsilon \rangle \) to \( \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \). By definition of \( \text{Val} \), we have that \( \|\mathcal{P}_q\| = \|\text{Val}\| = \mathcal{O}(N^s) \) and, moreover, for each \( \rho \in \Omega(q) \) (a path of \( A^{\text{df}(k)} \)) there exists a path \( \pi \in \mathcal{P}_q \) such that \( \omega(\rho) \upharpoonright B_\pi = \omega(\pi) \upharpoonright B_\pi = a_{i_1} \ldots a_{i_s} \) where \( \langle q, \varepsilon \rangle \) and \( \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \) are the endpoints of \( \pi \).

Hence, we define \( C_q \) to be the set of cycles in \( A^{\text{df}(k)} \) corresponding to the paths in \( \mathcal{P}_q \). The latter can be computed applying Dijkstra’s single source shortest path algorithm on \( H \), with source vertex \( \langle q, \varepsilon \rangle \), and assuming that the distance between adjacent vertices is always 1. The running time of the Dijkstra’s algorithm is \( \mathcal{O}(|V|^2) = |G|^{O(k)} \). Upon termination, one can reconstruct a shortest path \( \pi \) from \( \langle q, \varepsilon \rangle \) to each vertex \( \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \), and add the corresponding cycle of \( A^{\text{df}(k)} \) to \( C_q \). Since there are at most \( |G|^{O(k)} \) vertices \( \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \) in \( V \), and building a shortest path for each such vertex takes at most \( |G|^{O(k)} \) time, we can populate the set \( C_q \) in \( |G|^{O(k)} \) time. Once the sets \( C_q \) are built, it remains to compute the bounded expressions \( \text{concat}(\omega(\pi) \mid \pi \in C_q) \), \( \text{concat}(\Delta)^{N-1} \) and \( B_0, \ldots, B_N \). As shown below, they are all computable in time \( |G|^{O(k)} \).

Algorithm 1 gives the construction of \( \Gamma_\pi \). An upper bound on the time needed for building \( \Gamma_\pi \) can be derived by a close analysis of the running time of Algorithm 1. The input to the algorithm is a grammar \( G \), a strict \( s \)-letter bounded expression \( \bar{B} \) and an integer \( k > 0 \). First (lines 2–5), the algorithm builds the \( H \) graph, which takes time \( |G|^{O(k)} \). The loop on (lines 6–9) computes, for each vertex \( q \in \{Y(k)\} \), and each tuple \( v \in \text{Val} \), an elementary path from \( \langle q, \varepsilon \rangle \) to \( \langle q, a_1^{\ell_1} \ldots a_{s}^{\ell_s} \rangle \) in \( H \). For each \( q \), this set is kept in a variable \( \mathcal{P}_q \) (line 8). The variable \( B_\vartheta \) at the end of the loop contains the expression \( \text{concat}(\omega(\pi) \mid \pi \in \mathcal{P}_{q_1}) \ldots \text{concat}(\omega(\pi) \mid \pi \in \mathcal{P}_{q_N}) \).

Since both \( \|\{Y(k)\}\| = |G|^{O(k)} \) and \( \|\text{Val}\| = |G|^{O(k)} \), the loop at (lines 6–9) takes time \( |G|^{O(k)} \) as well.

The remaining part of the algorithm computes first an over-approximation of \( \text{concat}(\Delta)^{N-1} \) (lines 10–12) in the variable \( C \) – observe that the algorithm computes \( \text{concat}(\Delta)^{|G|^{2k}} - 1 \).
instead of $\text{concat}((\Delta)^{N-1})$. Finally, the control set $\Gamma_k$ with the needed property is produced by $|G|^{2k} \geq N$ repeated concatenations of the bounded expression $C : B_0$, at lines (13)–(15). Since both loops take time at most $|G|^{2k}$, we conclude that Algorithm 1 runs in time $|G|^{O(k)}$.

**B.4 Proof of Theorem 1**

**Proof.** We prove the theorem by induction on $d > 0$. If $d = 1, 2$, we obtain $\Gamma_b$ from Lemma 10, and time needed to compute $\Gamma_b^i$, using Algorithm 1, is $|G|^{O(k)}$. Moreover, $L_X^{(k)}(G) \subseteq L_X(\Gamma_b \cap \Gamma^{\text{df}(k+1)}, G) \subseteq L_X(\Gamma_b \cap \Gamma^{\text{df}(k+1)}, G)$.

For the induction step, assume $d \geq 3$. W.l.o.g. we assume that $G$ is reduced for $X$, and that $a_1^* \ldots a_d^*$ is the minimal bounded expression such that $L_X(G) \subseteq a_1^* \ldots a_d^*$. Consider the partition $\Xi_{\geq d} \cup \Xi_{< d} = \Xi$, defined in the previous. Since $G$ is reduced for $X$, then $X \in \Xi_{\geq d}$.

Let $\Delta_{\text{pivot}} = \{(X_i, yz) \in \Delta \mid X_i \in \Xi_{\geq d} \text{ and } y, z \in \Xi_{< d} \cup A \cup \{\varepsilon\}\}$. By Lemma 10 for each $X_i \in \Xi$, such that $L_{X_i}(G) \subseteq a_1^*a_d^*$, there exists a bounded expression $\Gamma_{X_i}^{(k)}$ such that $L_{X_i}(\Gamma_{X_i}^{(k)}) \subseteq \hat{L}_{X_i}(\Gamma_{X_i,k'} \cap \Gamma^{\text{df}(k+1)}, G)$. Moreover, by the induction hypothesis, for each $\ell, m, r$ such that $1 \leq \ell \leq m \leq r \leq d$, $d - \ell < d - 1$, and $r - m < d - 1$, and for each $Y, Z \in \Xi$ such that $L_Y(G) \subseteq a_1^* \ldots a_m^*$ and $L_Z(G) \subseteq a_m^* \ldots a_r^*$, there exist bounded expressions $\Gamma^Y_{\ell..m}, \Gamma^Z_{\ell..m}$ over $\Delta_{\text{pivot}}$ such that $\hat{L}^Y_{\ell..m}(G) \subseteq \hat{L}_{Y}(\Gamma^Y_{\ell..m} \cap \Gamma^{\text{df}(k+1)}, G)$ and $L^Z_{\ell..m}(G) \subseteq L_{Z}(\Gamma^Z_{\ell..m} \cap \Gamma^{\text{df}(k+1)}, G)$. We extend this notation to symbols from $A \cup \{\varepsilon\}$, and assume that $\Gamma_{\ell..m} = \{\varepsilon\}$ in case $y \in A \cup \{\varepsilon\}$. We define:

\[
\begin{align*}
\text{IH} &= \{(\ell, m, r) \mid 1 \leq \ell \leq m \leq r \leq d, \ d - \ell < d - 1 \text{ and } r - m < d - 1\} \text{ and the set } \\
\mathcal{S}_b &= \left\{ \Gamma_{X_i}^{(k)}, (X_i, yz)^*, \Gamma^Y_{\ell..m}, \Gamma^Z_{\ell..m} \mid (X_i, yz) \in \Delta_{\text{pivot}} \cap L_{X_i}(G) \subseteq a_1^*a_d^* \wedge (\ell, m, r) \in \text{IH} \right\}
\end{align*}
\]

Observe that, since each member of the $\mathcal{S}_b$ set is a bounded expression, the union of their languages is captured by the concatenation (the order is irrelevant) of these expressions. As noted earlier, the resulting expression, defined as $\Gamma_{\geq d}$, is again a bounded expression.

First, let us prove that $L_{X_i}(G) \subseteq \hat{L}_{X_i}(\Gamma_{\geq d} \cap \Gamma^{\text{df}(k+1)}, G)$. Let $w \in L_{X_i}(G)$ be a word, and $X \overset{\gamma}{\Rightarrow} w$ be a $k$-index depth first derivation of $w$ in $G$—since $w \in L_{X_i}(G)$, such a derivation is guaranteed to exist. By Lemma 9 there exists a production $(X_i, yz) \in \Delta_{\text{pivot}}$, and control words $\gamma^y \in (\Delta^k)^*$, $\gamma_y \gamma_z \in (\Delta_{i,yz})^*$, such that $\gamma^y \cdot (X_i, yz) \cdot \gamma_y \gamma_z$ is a permutation of $\gamma$, and:

\[
\begin{align*}
X \overset{\gamma^y}{\Rightarrow} u y v &\text{ a step sequence of } G^2 \text{ with } u, v \in A^*; \\
y \overset{\gamma_y}{\Rightarrow} u_y \text{ and } z \overset{\gamma_z}{\Rightarrow} u_z \text{ are derivations of } G_{i,yz} \text{ (hence } u_y u_z \in A^*) \text{, max}(k_y, k_z) < k \\
\text{ and } \min(k_y, k_z) < k - 1; \\
X \overset{\gamma^y}{\Rightarrow} w &\text{ is a derivation of } G^2 \text{ if } y \overset{\gamma_y}{\Rightarrow} u_y \text{ is a derivation of } G_{i,yz}; \\
X \overset{\gamma^y}{\Rightarrow} w &\text{ is a derivation of } G^2 \text{ if } z \overset{\gamma_z}{\Rightarrow} u_z \text{ is a derivation of } G_{i,yz}; \\
L_{X_i}(G^2) &\subseteq a_1^*a_d^*; \\
L_y(G_{i,u}) &\subseteq a_1^* \ldots a_m^* \text{ if } y \in \Xi_{< d}; \\
L_z(G_{i,u}) &\subseteq a_m^* \ldots a_r^* \text{ if } z \in \Xi_{< d}, \text{ with } 1 \leq \ell \leq m \leq r \leq d, \text{ such that } m - \ell < d - 1 \text{ and } r - m < d - 1.
\end{align*}
\]

Let us consider the case where $y, z \in \Xi$ (the other cases of $y \in A \cup \{\varepsilon\}$ or $z \in A \cup \{\varepsilon\}$ being similar, are left to the reader). We also assume $k_y < k - 1$ the other case being symmetric. Therefore, by the induction hypothesis there exist control words $\gamma^y \in \Gamma^Y_{\ell..m}$ and $\gamma_z \in \Gamma^Z_{\ell..m}$ such that $y \overset{\gamma^y}{\Rightarrow} u_y$ and $z \overset{\gamma_z}{\Rightarrow} u_z$. If $L_{X_i}(G^2) \subseteq a_1^*a_d^*$, by Lemma 10 there exists
a control word $\gamma^2 \in \Gamma_{\text{df}(k+1)}$ such that $X \xrightarrow{\gamma^2 \text{df}(k+1)} uX_i v$ is a $(k+1)$-index depth first step sequence in $G^\gamma$. It follows that:

$$X \xrightarrow{\gamma^2 \text{df}(k+1)} uX_i v \xrightarrow{(X_i,wz) \text{df}(k+1)} uy z v \xrightarrow{\gamma'^r \text{df}(k+2)} uy_g z v \xrightarrow{\gamma'^r \text{df}(k+1)} u u_g u_z v = w.$$ 

Observe that $u y z v \xrightarrow{\gamma'^r \text{df}(k+1)} u u_g u_z v$ because $u, v \in A^*$, $z \in \Sigma$ and $y \xrightarrow{\gamma'^r \text{df}(k+2)} u_g$. Since $k_y \leq k - 1$ and $k_z \leq k$, we find that $k_y + 2 \leq k + 1$ and $k_z + 1 \leq k + 1$, respectively. Hence the overall index of the foregoing derivation with control word $(\gamma^2, (X_i, yz), \gamma'^r, \gamma'^r) \in \Gamma_{\text{df}(k+1)}$ is at most $k + 1$. Since it is also a depth-first derivation, we finally find that $w \in L_X(\Gamma_{\text{df}(k+1)} \cap \Gamma_{\text{df}(k+1)}(G), G)$.

Finally, we address the time complexity of the construction of $\Gamma_{\text{df}}$. We refer to Algorihm [2] in the following. Notice first that both the $\text{minimizeExpression}$ and $\text{partitionNonterminals}$ functions take time $O(|G|)$, because emptiness of the intersection between a context-free and a finite automaton of constant size is linear in the size of the grammar (see Section 5 [1] for details). Moreover, the inclusion check on (line 12) is possible also in time $O(|G|)$ (by a similar argument). By Lemma 10, a call to $\text{ConstantBoundedControlSet}(G, b, k)$ will take time $|G|^{O(k)}$. Lemma 9 shows that the sizes of the minimal bounded expression considered at lines 11 and 19 in a recursive call, sum up to the size of the minimal bounded expression for the current call. Thus the total number of recursive calls is at most $d$. We thus let $T(d)$ denote the time needed for the top-level call of the function $\text{LetterBoundedControlSet}(G, X, a_1^* \ldots a_n^*, k)$ to complete. Since the loop on (lines 11-21) will be taken at most $\|\Delta\| \leq |G|$ times, we obtain:

$$T(d) = |G|^{O(k)} + |G|(|G| + 2T(d-1))$$

where $2T(d-1)$ is the time needed for the two recursive calls at lines 16 and 19 to complete. Because $T(0) = O(|G|) + |G|^{O(k)}$, we find that: $T(d) = |G|^{O(k)+d}$.

**B.5 Proof of Theorem 12**

**Proof.** Let $A = \{a_1, \ldots, a_d\}$ be an alphabet disjoint from $\Sigma$, and $h : A \rightarrow \Sigma^*$ be the homomorphism defined as $h(a_i) = w_i$, for all $i = 1, \ldots, d$. By Lemma [7] there exists a grammar $G^w = (\Sigma^*, A, \Delta^w)$, a set $V_X = \{Y_1, \ldots, Y_n\} \subseteq \Delta^w$, and a mapping $\xi : \Delta^w \rightarrow \Delta$, such that the following hold:

- $L_Y^k(G^w) \subseteq \hat{b} = a_1^* \ldots a_d^*$,
- for any control set $\Gamma \subseteq \Gamma_{\text{df}(k)}(G^w)$, with the property $L_Y^k(G^w) \subseteq L_Y(\Gamma, G^w)$, for all $Y \in V_X$, it is the case that $L_Y^k(\Gamma) \subseteq \hat{L}_X(\xi(\Gamma), G)$.

Let $Y \in V_X$, since $L_Y^k(G^w) \subseteq \hat{b}$, by Theorem [11] there exists a bounded expression $\Gamma_Y^k$ over $\Delta^w$ such that $L_Y^k(\Gamma_Y^k) \subseteq \hat{L}_Y(\Gamma_Y, G^w)$. Define $\Gamma_Y^b = \Gamma_Y^1 \ldots \Gamma_Y^n$. Since $\Gamma_Y^k \subseteq \Gamma_Y^b$, for all $Y \in V_X$, then clearly $L_Y^k(G^w) \subseteq \hat{L}_Y(\Gamma_Y^b) \cap \Gamma_{\text{df}(k+1)}(G^w)$. Since $Y$ was chosen arbitrarily, the latter holds for any $Y \in V_X$. By Lemma [7] we have:

- $L_Y^k(\Gamma) \subseteq \hat{L}_X(\xi(\Gamma^b) \cap \Gamma_{\text{df}(k+1)}(G^w), G)$,
- $\hat{L}_X(\xi(\Gamma^b) \cap \xi(\Gamma_{\text{df}(k+1)}(G^w), G))$,
- $\hat{L}_X(\xi(\Gamma^b) \cap \Gamma_{\text{df}(k+1)}(G, G))$. 


The last inclusion is due to \( \xi(\Gamma_{\text{df}}^{(k+1)}(G^\infty)) \subseteq \Gamma_{\text{df}}^{(k+1)}(G) \), which is an easy check, by the construction of \( G^\infty \). We define thus \( \Gamma_b = \xi(\Gamma_b') \).

Finally, since \( \Gamma_b \) can be built in time \( |G^\infty|^{O(k)+d} \) (Theorem 11) and \( G^\infty \) can be built in \( O(|b|^3 \cdot |G|) \) (Lem. 7), then \( \Gamma_b \) can be built in \( |b| \cdot |G|^{O(k)+d} \).

### B.6 Proof of Lemma 13

**Proof.** For each \( i \in \{1, \ldots, n\} \), let \( p_i \) be the production \( X_i \rightarrow X_{i-1} X_i \) of \( G_n \), and let \( p_0 \) be \( X_0 \rightarrow a \). It is easy to see that, because the derivation is depth-first, the control word \( \gamma \) generating \( a^{2^k} \) from \( X_k \) is unique.

Now suppose that there exists \( \Gamma_b = w_1^* \ldots w_d^* \) such that \( \gamma = w_1^{i_1} \ldots w_d^{i_d} \), for some \( i_1, \ldots, i_d \geq 0 \). What we show next is that no \( i_j \) can be larger than 2.

We first make this crucial observation, since the derivation tree is binary and its traversal is depth-first, we have that for every \( p_i \), every three consecutive occurrences \( \ell_1 < \ell_2 < \ell_3 \) of \( p_i - (\gamma)_\ell = (\gamma)_\ell = (\gamma)_\ell = p_i \) implies that there exists a position \( \ell \) between \( \ell_1 \) and \( \ell_3 \) such that \( (\gamma)_\ell = p_{i+1} \). Otherwise that would imply that the derivation tree has a node \( X_{i+1} \) with three \( X_i \) children; or that the tree was not traversed in depth-first.

Take an arbitrary \( w_j \) in \( \Gamma_b \) and let \( g \) be the greatest production index occurring in \( w_j \). The number \( i_j \) of repetitions of \( w_j \) cannot be greater than two for otherwise \( p_g \) contradicts the previous fact. So this concludes that no \( i_j \) can be larger than 2.

Now, since the only string of \( L_{\chi_k}(G) \) has length \( 2^k \) and that no rule produces more than one terminal then necessarily \( |\gamma| \geq 2^k \). So we show that \( |\Gamma_b| \) has to be at least \( 2^{k-1} \). By contradiction, suppose \( |\Gamma_b| \leq (2^{k-1} - 1) \), then since in order to capture \( \gamma \) no word of \( \Gamma_b \) can occur more than twice, the longest control word that \( \Gamma_b \) can capture is \( 2 \cdot (2^{k-1} - 1) = 2^k - 2 \) which is shorter than \( 2^k = |\gamma| \), hence a contradiction.

\[ \star \]