Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 83-94

Boolean algebras of unambiguous context-free languages

Didier Caucal
Institut Gaspard Monge, CNRS – Université Paris-Est
caucal@univ-mlv.fr

ABSTRACT. Several recent works have studied subfamilies of deterministic context-free languages with good closure properties, for instance the families of input-driven or visibly pushdown languages, or more generally families of languages accepted by pushdown automata whose stack height can be uniquely determined by the input word read so far. These ideas can be described as a notion of synchronization. In this paper we present an extension of synchronization to all context-free languages using graph grammars. This generalization allows one to define boolean algebras of non-deterministic but unambiguous context-free languages containing regular languages.

1 Introduction

Several restrictions of pushdown automata were recently studied in order to define classes of languages which generalize regular languages while retaining some of their closure properties, namely closure under boolean operations, concatenation and its iteration. All of these approaches consist in defining a notion of synchronization between pushdown automata [AM 04, Ca 06, NS 07] (see also [LMM 08] for complexity results). An approach which also avoids a special treatment of the ε-moves, is to define the synchronization at graph level [CH 08]. More precisely, the transition graph of any pushdown automaton A can be generated by a (deterministic graph) grammar R [MS 85, Ca 07] using infinite parallel rewritings. The stack height of a configuration of A is replaced by its weight, which is the minimal number of steps of parallel rewriting by R necessary to produce it.

The notion of synchronization can be defined for all graph grammars. A grammar G is synchronized by a grammar H if for any accepting path λ of (the graph generated by) G, there exists an accepting path μ of H with the same label u such that λ and μ are synchronized: for every prefix v of u, the prefixes of λ and μ labelled by v lead to vertices of the same weight. By extending usual constructions from finite automata to grammars generating deterministic graphs, we have shown that the languages recognized by all grammars synchronized with a given grammar form a boolean algebra lying between regular languages and deterministic context-free languages [CH 08].

In this paper, we apply the notion of synchronization to graph grammars recognizing unambiguous context-free languages, which are the languages generated by context-free grammars with at most one derivation tree for each word. Although these languages form a natural generalization of deterministic context-free languages, their equivalence problem remains a challenge in formal language theory [Gi 66]. Recent developments can be found in [Wi 04]. We present two classes of graph grammars, called unambiguous and level-unambiguous, recognizing all unambiguous context-free languages. A grammar is unambiguous if two accepting paths in the generated graph have distinct labels. More

FSTTCS 2008
IARCS Annual Conference on
Foundations of Software Technology and Theoretical Computer Science
http://drops.dagstuhl.de/opus/volltexte/2008/1743

generally, a grammar is level-unambiguous if two accepting paths with the same label are synchronized. We show that the languages recognized by grammars synchronized with a fixed level-unambiguous grammar form a boolean algebra containing the regular languages (where the complement operation is relative to the language of the synchronizing grammar). A direct consequence is the decidability of the inclusion problem between languages recognized by two level-unambiguous grammars synchronized by a third one.

The paper is structured as follows: after recalling some notations and definitions in Sections 2 and 3, we present the notion of synchronization of arbitrary grammars in Section 4. We then focus on the closure properties of level-unambiguous grammars in Section 5.

2 Notations

Let $\mathbb{N}$ be the set of natural numbers. For a set E, we write $|E|$ its cardinality, 2^E its powerset and for every $n \geq 0$, $E^n = \{(e_1, \ldots, e_n) \mid e_1, \ldots, e_n \in E\}$ is the set of n-tuples of elements of E. Thus $E^* = \bigcup_{n\geq 0} E^n$ is the free monoid generated by E for the *concatenation*: $(e_1, \ldots, e_m)\cdot(e'_1, \ldots, e'_n) = (e_1, \ldots, e_m, e'_1, \ldots, e'_n)$, whose neutral element is the 0-tuple $()$. A finite set E of symbols is an *alphabet* of *letters*, and E^* is the set of *words* over E. Any word $u \in E^n$ is of *length* $|u| = n$ and is also represented by a mapping from $[n] = \{1, \ldots, n\}$ into E, or by the juxtaposition of its letters: $u = u(1)\ldots u(|u|)$. The neutral element is the word of length 0 called the *empty word* and denoted by ε. We denote by $[0, n] = \{0, \ldots, n\}$ for any $n \in \mathbb{N}$. For any binary relation R, we also write xRy for $(x, y) \in R$; as usual $Dom(R) = \{x \mid \exists y, xRy\}$ and $Im(R) = \{y \mid \exists x, xRy\}$ are the *domain* and the *image* of R.

Let F be a set of symbols called *labels*, ranked by a mapping $\varrho : F \longrightarrow \mathbb{N}$ associating to each label f its *arity* $\varrho(f) \geq 0$, and such that $F_n := \{f \in F \mid \varrho(f) = n\}$ is countable for every $n \geq 0$. We consider simple, oriented and labelled hypergraphs: a *hypergraph* G is a subset of $\bigcup_{n\geq 0} F_n V^n$, where V is an arbitrary set, such that

- its *vertex* set $V_G := \{v \in V \mid FV^*vV^* \cap G \neq \emptyset\}$ is finite or countable,
- its *label* set $F_G := \{f \in F \mid fV^* \cap G \neq \emptyset\}$ is finite.

Any $fv_1\ldots v_{\varrho(f)} \in G$ is a *hyperarc* labelled by f and of successive vertices $v_1, \ldots, v_{\varrho(f)}$; it is depicted according to the arity of f as follows:

- for $\varrho(f) \geq 2$, as an arrow labelled f and successively linking $v_1, \ldots, v_{\varrho(f)}$;
- for $\varrho(f) = 1$, as a label f on vertex v_1 (f is called a *colour* of v_1);
- for $\varrho(f) = 0$, as an isolated label f called a *constant*.

This is illustrated in the next figures. Note that a vertex v is depicted by a dot named (v) where parentheses are used to differentiate a vertex name from a vertex label (a colour).

For a subset $E \subseteq F$ of labels, we write $V_{G,E} := \{v \in V \mid EV^*vV^* \cap G \neq \emptyset\} = V_{G\cap EV_G^*}$ the set of vertices of G linked by a hyperarc labelled in E. A *graph* G is a hypergraph whose labels are only of arity 1 or 2: $F_G \subset F_1 \cup F_2$. Hence a graph G is a set of *arcs* av_1v_2 identified with the labelled transition $v_1 \xrightarrow[G]{a} v_2$ or directly $v_1 \xrightarrow{a} v_2$ if G is understood, plus a set of coloured vertices fv.

A tuple $(v_0, a_1, v_1, \ldots, a_n, v_n)$ with $n \geq 0$ and $v_0 \xrightarrow[G]{a_1} v_1 \ldots v_{n-1} \xrightarrow[G]{a_n} v_n$ is called a *path* from v_0 to v_n labelled by $u = a_1 \ldots a_n$; we write $v_0 \xRightarrow[G]{u} v_n$ or directly $v_0 \xRightarrow{u} v_n$ if G is understood. For $P, Q \subseteq V_G$ and $u \in F_2^*$, we write $P \xRightarrow[G]{u} Q$ if $p \xRightarrow[G]{u} q$ for some $p \in P$ and $q \in Q$ and

$L(G,P,Q) := \{u \mid P \xRightarrow[G]{u} Q\}$ is the language recognized by G from P to Q. In these notations, we can replace P (and/or Q) by a colour $\circ$ to designate the subset $V_{G,\circ}$. In particular $\circ \xRightarrow[G]{u} Q$ means that there is a path labelled by u from a vertex coloured by $\circ$ to a vertex in Q, and $L(G,\circ,\bullet)$ is the label set of the paths from a vertex coloured by $\circ$ to a vertex coloured by $\bullet$.

In this paper, we use two colours $\circ, \bullet \in F_1$ to mark respectively the initial vertices and the final vertices. To depict an initial or final vertex, the dot is replaced by its colour, and $\circledcirc$ represents a vertex which is initial and final. For any graph G, we denote by $L(G) := L(G,\circ,\bullet)$ the *language recognized* by G. Recall that the *regular languages* over an alphabet $T \subset F_2$ form the set $Reg(T^*) := \{L(G) \mid G \text{ finite } \wedge\, F_G \subseteq T \cup \{\circ,\bullet\}\}$.

3 Graph grammars

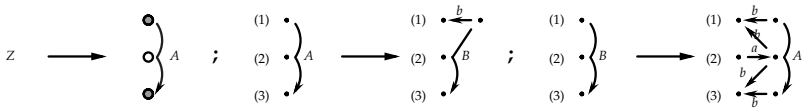
In this section, we recall the definition of deterministic graph grammars, together with the family of graphs they generate (called regular graphs). Using initial and final vertices, they can be viewed as infinite automata, generalizing finite automata. We also define two restricted classes of grammars recognizing all unambiguous context-free languages.

A graph *grammar* R is a finite set of rules of the form $fx_1 \ldots x_{\varrho(f)} \longrightarrow H$ where $fx_1 \ldots x_{\varrho(f)}$ is a hyperarc joining pairwise distinct vertices $x_1 \neq \ldots \neq x_{\varrho(f)}$ and H is a finite hypergraph; we denote by $N_R := \{f \in F \mid \exists x_1, \ldots, x_{\varrho(f)},\ fx_1 \ldots x_{\varrho(f)} \in Dom(R)\}$ the *non-terminals* of R (the labels of the left hand sides), by $T_R := \{f \in F - N_R \mid \exists H \in Im(R), V_{H,f} \neq \emptyset\}$ the *terminals* of R (the labels of R which are not non-terminals), and by $F_R := N_R \cup T_R$ the labels of R. We use grammars to generate graphs. Hence in the following, we assume that any terminal is of arity 1 or 2: $T_R \subset F_1 \cup F_2$.

Like a context-free grammar (on words), a graph grammar has an axiom, which is an initial finite hypergraph. To specify this axiom, we assume that any grammar R has a constant non-terminal $Z \in N_R \cap F_0$ which does not appear in any right hand side; the *axiom* of R is the right hand side H of the rule corresponding to Z: $Z \longrightarrow H \wedge Z \notin F_K$ for any $K \in Im(R)$.

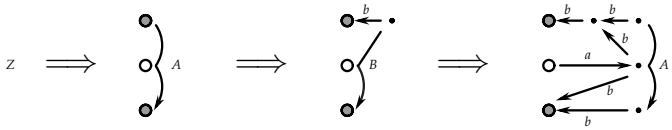
Starting from the axiom, we want R to generate a unique graph up to isomorphism. So we finally assume that any grammar R is *deterministic*, meaning that there is only one rule per non-terminal: $(X,H),(Y,K) \in R \wedge X(1) = Y(1) \Longrightarrow (X,H) = (Y,K)$. For any rule $X \longrightarrow H$, we say that $V_X \cap V_H$ are the *inputs* of H and $\bigcup\{V_Y \mid Y \in H \wedge Y(1) \in N_R\}$ are the *outputs* of H. For convenience and without loss of generality, it is simpler to assume that any grammar R is *terminal-outside* [Ca 07], meaning that there should be at least one non-input vertex in the support of any terminal arc or colour in a right hand side: $H \cap (T_R V_X V_X \cup T_R V_X) = \emptyset$ for any rule $(X,H) \in R$. We use upper-case letters $A, B, C, \ldots$ to denote non-terminals and lower-case letters $a, b, c \ldots$ for terminals.

The next figure shows an example of a (deterministic graph) grammar *Double* with non-terminals Z, A, B, terminals $a, b, \circ, \bullet$ and rule inputs $1, 2, 3$ (except for the axiom rule which has no input).

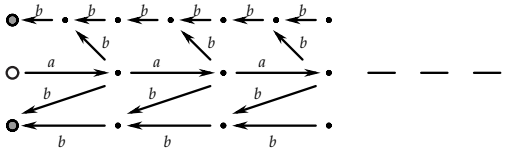


Given a grammar R, the *rewriting* $\underset{R}{\longrightarrow}$ is the binary relation between hypergraphs defined as follows: M rewrites into N, written $M \underset{R}{\longrightarrow} N$, if we can choose a non-terminal hyperarc $X = As_1 \dots s_p$ in M and a rule $Ax_1 \dots x_p \longrightarrow H$ in R such that N can be obtained by replacing X by H in M: $N = (M - X) \cup h(H)$ for some function h mapping each x_i to s_i, and the other vertices of H injectively to vertices outside of M; this rewriting is denoted by $M \underset{R,X}{\longrightarrow} N$. The rewriting $\underset{R,X}{\longrightarrow}$ of a hyperarc X is extended in an obvious way to the rewriting $\underset{R,E}{\longrightarrow}$ of any set E of non-terminal hyperarcs.

The *complete parallel rewriting* $\underset{R}{\Longrightarrow}$ is a simultaneous rewriting according to the set of all non-terminal hyperarcs: $M \underset{R}{\Longrightarrow} N$ if $M \underset{R,E}{\longrightarrow} N$ where E is the set of all non-terminal hyperarcs of M. We depict below the first three steps of the parallel derivation of the previous grammar *Double* from its constant non-terminal Z:



Given a deterministic grammar R and a hypergraph H, we denote by $[H] := H \cap T_R V_H^* = H \cap (T_R V_H V_H \cup T_R V_H)$ the set of terminal arcs and of terminal coloured vertices of H. A graph G is *generated* by R (from its axiom) if G belongs to the set of isomorphic graphs $R^\omega := \{\bigcup_{n \geq 0} [H_n] \mid Z \underset{R}{\longrightarrow} H_0 \underset{R}{\Longrightarrow} \dots H_n \underset{R}{\Longrightarrow} H_{n+1} \dots\}$. For instance by indefinitely iterating the previous derivation, we get the following infinite graph:



We call *regular* a graph generated by a (deterministic graph) grammar. Given a (regular) graph $G = \bigcup_{n \geq 0} [H_n]$ generated by a grammar R, with $Z \underset{R}{\longrightarrow} H_0 \underset{R}{\Longrightarrow} \dots H_n \underset{R}{\Longrightarrow} H_{n+1} \dots$, we define the *level* $\ell(s)$ of a vertex $s \in V_G$, denoted also $\ell_G^R(s)$ to specify G and R, as the minimal number of rewritings applied from the axiom to obtain s: $\ell(s) := \min\{n \mid s \in V_{H_n}\}$. The previous graph is represented by vertices of increasing level: vertices of the same level are vertically aligned for clarity. For any grammar R and for $G \in R^\omega$, we denote by $L(R) := L(G)$ the *language recognized* by R, which is well-defined since all graphs generated by a grammar are isomorphic. For instance, the grammar *Double* above recognizes the language $L(Double) = \{a^n b^n \mid n > 0\} \cup \{a^n b^{2n} \mid n > 0\}$.

A graph G is *deterministic* if $\circ$ colours a unique vertex, and two arcs with the same source have distinct labels: $r \underset{G}{\overset{a}{\longrightarrow}} s \wedge r \underset{G}{\overset{a}{\longrightarrow}} t \implies s = t$. Deterministic graph grammars recognize the family of context-free languages. The restriction to grammars generating a deterministic graph yields the family of deterministic context-free languages [Ca 07]. A grammar R is *unambiguous* if any pair of accepting paths have distinct labels: for $G \in R^\omega$,

$$s_0 \underset{G}{\overset{a_1}{\longrightarrow}} s_1 \dots \underset{G}{\overset{a_n}{\longrightarrow}} s_n \ \wedge \ t_0 \underset{G}{\overset{a_1}{\longrightarrow}} t_1 \dots \underset{G}{\overset{a_n}{\longrightarrow}} t_n \ \wedge \ \circ s_0, \circ t_0, \bullet s_n, \bullet t_n \in G \implies s_i = t_i \ \forall i \in [0, n].$$

Note that the previous grammar is unambiguous. Any grammar generating a deterministic graph is unambiguous. However, unambiguous grammars recognize strictly more languages than deterministic ones.

PROPOSITION 1. *Unambiguous grammars recognize the family of unambiguous context-free languages.*

Recall that there exist context-free languages which are not unambiguous *i.e.* which cannot be generated by an unambiguous context-free grammar; they are called inherently *ambiguous* context-free languages. An example of an ambiguous context-free language is $\{a^m b^m a^n b^n \mid m,n \geq 0\} \cup \{a^m b^n a^n b^m \mid m,n \geq 0\}$.

The synchronization relation we will soon define requires a slight generalization of unambiguous grammars. A grammar R is called *level-unambiguous* if for any pair of accepting paths λ, μ with the same label u and for every prefix v of u, the prefixes of λ and μ labelled by v lead to vertices of the same level. Formally, for (any) $G \in R^\omega$,

$$s_0 \xrightarrow[G]{a_1} s_1 \ldots \xrightarrow[G]{a_n} s_n \ \wedge\ t_0 \xrightarrow[G]{a_1} t_1 \ldots \xrightarrow[G]{a_n} t_n \ \wedge\ \circ s_0, \circ t_0, \bullet s_n, \bullet t_n \in G \implies \ell^R_G(s_i) = \ell^R_G(t_i)\ \ \forall i \in [0,n].$$

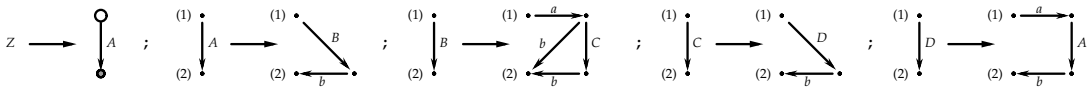
Note that any unambiguous grammar is also level-unambiguous. One can prove (Cf. Lemmas 12 and 13) that even though they are slightly more general, level-unambiguous grammars do not recognize more languages than unambiguous ones.

PROPOSITION 2. *Level-unambiguous grammars recognize the family of unambiguous context-free languages.*

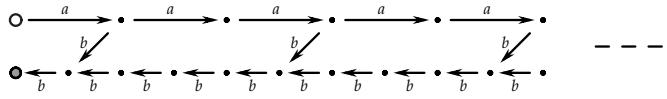
4 Synchronization of grammars

The notion of synchronization was defined in earlier work as a binary relation between grammars generating deterministic graphs [CH 08]. In this section, we extend it to all grammars. To each grammar R, we associate the family $\mathrm{Sync}(R)$ of languages recognized by grammars synchronized by R. We give closure properties of $\mathrm{Sync}(R)$ and show that this family is independent of the way to generate R^ω.

A grammar R *synchronizes* a grammar S, and we write $R \rhd S$ or $S \lhd R$ if for (any) $G \in R^\omega$ and (any) $H \in S^\omega$, whenever there exists a path $t_0 \xrightarrow[H]{a_1} t_1 \ldots \xrightarrow[H]{a_n} t_n$ with $\circ t_0, \bullet t_n \in H$, then there exists $s_0 \xrightarrow[G]{a_1} s_1 \ldots \xrightarrow[G]{a_n} s_n$ with $\circ s_0, \bullet s_n \in G$ and $\ell^R_G(s_i) = \ell^S_H(t_i)\ \forall i \in [0,n]$, meaning that for any accepting path μ labelled by u in the graph generated by S, there must be an accepting path λ label by u in the graph generated by R such that for every prefix v of u, the prefixes of λ and μ labelled by v lead to vertices of the same level.
For instance the grammar *Double* of the previous section synchronizes the following grammar S:



whose generated graph is represented by vertices of increasing level as follows:



and whose accepted language is $L(S) = \{a^{2n+1}b^{4n+2} \mid n \geq 0\}$.

Note in particular that $S \lhd R \implies L(S) \subseteq L(R)$. The relation $\rhd$ is reflexive and transitive but not antisymmetric. We denote by $\bowtie$ the *bi-synchronization* relation: $R \bowtie S$ if $R \rhd S$ and $S \rhd R$. The following lemma states that level-unambiguity is preserved for synchronized grammars.

LEMMA 3. *For any level-unambiguous grammar R:*
a) $S \lhd R \implies$ *S is level-unambiguous;*
b) $S \bowtie R \iff S \lhd R$ *and* $L(S) = L(R)$.

A useful transformation preserving bi-synchronization is to restrict to vertices accessible from $\circ$ and co-accessible from $\bullet$. The *restriction* $G_{|P}$ of a graph G to a subset $P \subseteq V_G$ of vertices is the subgraph of G induced by P:

$$G_{|P} := \{s \xrightarrow{a} t \mid s \xrightarrow[G]{a} t \wedge s, t \in P\} \cup \{cs \mid cs \in G \wedge s \in P\}.$$

We write $R^\omega_{\circ,\bullet} := \{G_{|\{s \mid \circ \underset{G}{\Longrightarrow} s \underset{G}{\Longrightarrow} \bullet\}} \mid G \in R^\omega\}$ the restriction of R^ω by accessibility from $\circ$ and co-accessibility from $\bullet$. We can restrict synchronization to grammars generating graphs accessible from their initial vertices and co-accessible from their final vertices.

LEMMA 4. *Any grammar R can be transformed into a grammar S such that $S \bowtie R$ and $S^\omega = R^\omega_{\circ,\bullet}$.*

Another basic transformation, given in Lemma 6.1 of [Ca 07] allows us to restrict ourselves to grammars with colours $\circ$ and $\bullet$ only in the axiom (i.e. whose generated graph only contains initial and final vertices at level 0). We say that a grammar R is *initial* when this is the case, *i.e.* when $(X, H) \in R \wedge X \neq Z \implies V_{H,\circ} = \emptyset = V_{H,\bullet}$.

This transformation works as follows. Let R be any grammar. We consider two arity 2 new symbols $i, f \in F_2$ such that $i, f \notin F_R$ and i, f are not vertices of R. To any non-terminal $A \in N_R - \{Z\}$, we associate a new symbol $A_{i,f}$ of arity $\varrho(A) + 2$. We consider the grammar:

$$\begin{aligned}[R, i, f] \;:=\; & \{(Z, H_{i,f} \cup \{\circ i, \bullet f\}) \mid (Z, H) \in R\} \cup \{(A_{i,f} X i f, H_{i,f}) \mid (AX, H) \in R \wedge A \neq Z\} \\ \text{where} \quad H_{i,f} \;:=\; & ([H] - \{\circ, \bullet\}V_H) \cup \{A_{i,f} X i f \mid AX \in H \wedge A \in N_R\} \\ & \cup \{i \xrightarrow{i} s \mid \circ s \in H\} \cup \{s \xrightarrow{f} f \mid \bullet s \in H\}.\end{aligned}$$

This grammar $[R, i, f]$ is an initial grammar such that, for any $G \in R^\omega$ with $i, f \notin V_G$, $G_{i,f} \cup \{\circ i, \bullet f\} \in [R, i, f]^\omega$. In particular $L([R, i, f]) = iL(R)f$. Moreover,

$S \lhd R \iff [S, i, f] \lhd [R, i, f]$ and $[R, i, f]$ is (level-)unambiguous if and only if R is.

Note that if R^ω has an infinite number of initial (resp. final) vertices then the initial (resp. final) vertex of $[R, i, f]^\omega$ is of infinite out-degree (resp. in-degree).

To any grammar R, we associate a family of *synchronized languages*

$$\mathrm{Sync}(R) \;:=\; \{L(S) \mid S \lhd R\}$$

which are thelanguages accepted by the grammars synchronized by R. Observe in particular that $R \bowtie S \implies \mathrm{Sync}(R) = \mathrm{Sync}(S)$, and $\mathrm{Sync}([R,i,f]) = \{iLf \mid L \in \mathrm{Sync}(R)\}$.

For any alphabet $T \subset F_2$, all the regular languages in T^* can be synchronized by the grammar Reg defined as the unique axiom rule $Z \longrightarrow \{0 \xrightarrow{a} 0 \mid a \in T\} \cup \{\circ 0, \bullet 0\}$ (in other words, $\mathrm{Sync}(\mathrm{Reg}) = Reg(T^*)$). Also note that any grammar R synchronizes any grammar without colour $\circ$ or $\bullet$, thus $\emptyset \in \mathrm{Sync}(R)$. Let us generalize this fact.

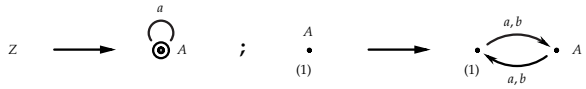
PROPOSITION 5. *For any grammar R, the family $\mathrm{Sync}(R)$ is closed under union, and contains $L(R) \cap M$ for any regular language M.*

PROOF. Closure under union will not be detailed here, but is straightforward. Containment of all regular languages inside $L(R)$ is done by synchronization product of R with a finite automaton K [CH 08]. Let $\{q_1, \ldots, q_n\} = V_K$ be the vertex set of K. To each $A \in N_R$, we associate a new symbol A' of arity $n \times \varrho(A)$, and to each hyperarc $Ar_1 \ldots r_m$ with $m = \varrho(A)$, we associate the hyperarc $(Ar_1 \ldots r_m)' := A'(r_1,q_1) \ldots (r_1,q_n) \ldots (r_m,q_1) \ldots (r_m,q_n)$. As an exception, we assimilate Z' to Z. We then define the grammar $R \times K$, which associates to each rule $(X,H) \in R$ the rule:

$$X' \longrightarrow \{(s,p) \xrightarrow{a} (t,q) \mid s \xrightarrow[H]{a} t \wedge p \xrightarrow[K]{a} q\} \cup \{(BU)' \mid BU \in H \wedge B \in N_R\}$$
$$\cup \{\circ(s,p) \mid \circ s \in H \wedge \circ p \in K\} \cup \{\bullet(s,p) \mid \bullet s \in H \wedge \bullet p \in K\}.$$

It is easily shown that $R \times K \lhd R$ and $L(R \times K) = L(R) \cap L(K)$. ∎

For any grammar R, the family $\mathrm{Sync}(R)$ is in general not closed under intersection, hence not closed under complement with respect to $L(R)$, since $L \cap M = L(R) - [(L(R) - L) \cup (L(R) - M)]$ for any $L, M \subseteq L(R)$. For instance the following grammar:

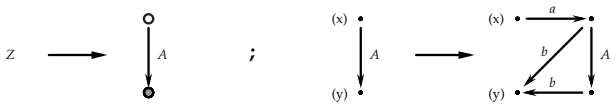


is not level-unambiguous, and for $L = \{a^m b^m a^n \mid m,n \geq 0\}$ and $M = \{a^m b^n a^n \mid m,n \geq 0\}$, we have $L, M \in \mathrm{Sync}(R)$ but $L \cap M = \{a^n b^n a^n \mid n \geq 0\} \notin \mathrm{Sync}(R)$.

For R^ω deterministic, $\mathrm{Sync}(R)$ coincides with the family of synchronized languages defined in [CH 08].

PROPOSITION 6. *For any grammar R such that R^ω is deterministic,*
$$\mathrm{Sync}(R) = \{L(S) \mid S \lhd R \wedge S^\omega \text{ deterministic}\}.$$

As a corollary of Proposition 6, $\mathrm{Sync}(R)$ is a boolean algebra when R^ω is deterministic [CH 08]. For instance, let *Single* be the following grammar:



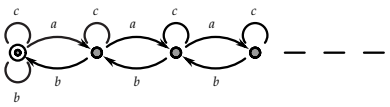
We have $L(Single) = \{a^n b^n \mid n > 0\}$ and $\mathrm{Sync}(Single) = \{L(G_{m,n}, I) \mid m \geq 0 \wedge n > 0\}$, where $L(G_{m,n}, I)$ is the language generated from I by the linear context-free grammar $G_{m,n}$:

$$I = P + a^m A b^m \quad \text{with} \quad P \subseteq \{ab, \ldots, a^m b^m\}$$
$$A = Q + a^n A b^n \quad \text{with} \quad Q \subseteq \{ab, \ldots, a^n b^n\}.$$

We conclude this section with a fundamental result concerning grammar synchronization, which states that $\mathrm{Sync}(R)$ is independent of the way the graph R^ω is generated.

THEOREM 7. *For any grammars R and S, $R^\omega = S^\omega \implies \mathrm{Sync}(R) = \mathrm{Sync}(S)$.*

This theorem allows to transfer the concept of grammar synchronization to the level of graphs: for any regular graph G, we can define $\mathrm{Sync}(G)$ as $\mathrm{Sync}(R)$ for any grammar R generating G. For instance, the following regular graph:



defines by synchronization the family of visibly pushdown languages (with a pushing, b popping and c internal) [AM 04].

5 Synchronization of level-unambiguous grammars

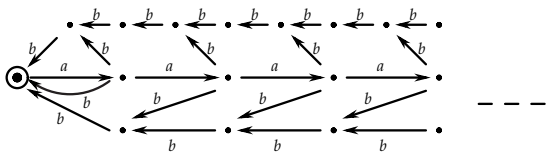
As previously stated, for any grammar R generating a deterministic graph, $\mathrm{Sync}(R)$ is an effective boolean algebra. In this section, we show that this remains true when R is level-unambiguous.

THEOREM 8. *For any level-unambiguous grammar R, the family $\mathrm{Sync}(R)$ is an effective boolean algebra with respect to $L(R)$, containing all the regular languages included in $L(R)$.*

For instance, let us consider the initial and unambiguous grammar *Double* of Section 3. We have $\mathrm{Sync}(Double) = \{L(G_{m,n}, I) \cup L(H_{m',n'}, I) \mid m, m' \geq 0 \wedge n, n' > 0\}$ where $G_{m,n}$ is defined above and $H_{m,n}$ is the following linear context-free grammar:

$$I = P + a^m A b^{2m} \quad \text{with} \quad P \subseteq \{abb, \ldots, a^m b^{2m}\}$$
$$A = Q + a^n A b^{2n} \quad \text{with} \quad Q \subseteq \{abb, \ldots, a^n b^{2n}\}.$$

This is indeed a boolean algebra with respect to $L(Double)$. Finally for the regular graph G



the family $\mathrm{Sync}(G)$ is the regular closure of $\mathrm{Sync}(Double)$.

A particular consequence of Theorem 8 is that we can decide the inclusion $L(S) \subseteq L(S')$ for two grammars S and S' synchronized by a common level-unambiguous grammar. Recall that the inclusion problem is undecidable for the so-called *simple* languages [Fr 77].

The constructions from [CH 08] cannot be trivially extended because level-unambiguity is a global property of accepted words and not a local property like graph determinism. However we can still work locally thanks to the notions of synchronization and level-unambiguity, which both only require to work level by level.

Closure under union was already stated in Proposition 5. We now proceed to prove the closures under intersection (Lemma 9) and complement (Lemma 14).

5.1 Closure under intersection

We will use other colours in addition to $\circ$ and $\bullet$. For any set of colours $C \subseteq F_1 - \{\circ\}$ and any grammar R, we denote R_C the grammar obtained from R by colouring every C-coloured vertex with $\bullet$ and removing $\bullet$ on all other vertices:

$$R_C := \{\big(X, (H - \{\bullet\}V_H) \cup \{\bullet p \mid \exists c \in C, cp \in H\}\big) \mid (X,H) \in R\}.$$

We define a level-preserving version of the grammar synchronization product. Let $\bullet_1, \bullet_2$ be new colours. Let R and S be two grammars, $G \in R^\omega$ and $H \in S^\omega$ two graphs they generate, and let $W := \{(s,p) \in V_G \times V_H \mid \ell^R_G(s) = \ell^S_H(p)\}$, the *level synchronization product* $G \times H$ is

$$\begin{aligned} G \times H := \{(s,p) \xrightarrow{a} (t,q) \mid s \xrightarrow[G]{a} t \wedge p \xrightarrow[H]{a} q \wedge (s,p),(t,q) \in W\} \\ \cup \{\circ(s,p) \mid \circ s \in G \wedge \circ p \in H \wedge (s,p) \in W\} \cup \{\bullet(s,p) \mid \bullet s \in G \wedge \bullet p \in H \wedge (s,p) \in W\} \\ \cup \{\bullet_1(s,p) \mid \bullet s \in G \wedge \bullet p \notin H \wedge (s,p) \in W\} \cup \{\bullet_2(s,p) \mid \bullet s \notin G \wedge \bullet p \in H \wedge (s,p) \in W\}. \end{aligned}$$

We then simply define $R^\omega \times S^\omega$ as $\{G \times H \mid G \in R^\omega \wedge H \in S^\omega\}$. The standard synchronization product of two regular graphs can be non regular, but the level synchronization product $R^\omega \times S^\omega$ can be generated by a grammar $R \times S$ that we define.

Let $(A,B) \in N_R \times N_S$ be any pair of non-terminals, we consider binary relations E over inputs such that $\forall i,j \in [\varrho(A)], E(i) \cap E(j) \neq \emptyset \implies E(i) = E(j)$, where $E(i) = \{j \mid (i,j) \in E\}$ denotes the *image* of $i \in [\varrho(A)]$. To any such A, B and E, we associate a new symbol $[A,B,E]$ of arity $|E|$ (where $[Z,Z,\emptyset]$ is assimilated to Z). To each non-terminal hyperarc $Ar_1 \ldots r_m$ of R ($A \in N_R$ and $m = \varrho(A)$) and each non-terminal hyperarc $Bs_1 \ldots s_n$ of S ($B \in N_S$ and $n = \varrho(B)$), we associate the hyperarc $[Ar_1 \ldots r_m, Bs_1 \ldots s_n, E] := [A,B,E](r_1,s_1)_E \ldots (r_1,s_n)_E \ldots (r_m,s_1)_E \ldots (r_m,s_n)_E$ with $(r_i,s_j)_E := (r_i,s_j)$ if $(i,j) \in E$, and ε otherwise. The grammar $R \times S$ is then defined as the set of rules

$$[AX, BY, E] \longrightarrow \big([P] \times [Q]\big)_{|\overline{E}} \cup \{[CU, DV, E'] \mid CU \in P \wedge C \in N_R \wedge DV \in Q \wedge D \in N_S\}$$

for each $(AX,P) \in R$, each $(BY,Q) \in S$, and each $E \subseteq [\varrho(A)] \times [\varrho(B)]$ with

$$\begin{aligned} \overline{E} &:= \{(X(i), Y(j)) \mid (i,j) \in E\} \cup \big(V_P - V_X\big) \times \big(V_Q - V_Y\big) \\ E' &:= \{(i,j) \in [\varrho(C)] \times [\varrho(D)] \mid (U(i), V(j)) \in \overline{E}\} \end{aligned}$$

and where the level synchronization product $[P] \times [Q]$ is defined according to

$$\ell(s) = \begin{cases} 0 & \text{if} \quad s \in V_X \\ 1 & \text{if} \quad s \in V_P - V_X \end{cases} \qquad \ell(t) = \begin{cases} 0 & \text{if} \quad t \in V_Y \\ 1 & \text{if} \quad t \in V_Q - V_Y. \end{cases}$$

Finally we restrict $R \times S$ to the non-terminals accessible from Z. This grammar indeed generates the level synchronization product $(R \times S)^\omega = R^\omega \times S^\omega$ of their generated graphs, and also satisfies the following properties:

$$(R \times S)_{\bullet,\bullet_1} \lhd R \quad ; \quad (R \times S)_{\bullet,\bullet_2} \lhd S \quad ; \quad S \lhd R \implies R \times S \bowtie S.$$

This implies that for any level-unambiguous R, $\mathrm{Sync}(R)$ is closed under intersection.

LEMMA 9. *For any $S, S' \lhd R$ with R level-unambiguous, $L(S \times S') = L(S) \cap L(S')$.*

5.2 Level-wise determinization

Before proving the closure under complement of $\mathrm{Sync}(R)$ in the next subsection, we need to define a suitable notion of level-wise determinism, and show that any level-unambiguous grammar is equivalent, in terms of synchronised languages, to one generating a level-wise deterministic graph. We say that a grammar R is *level-deterministic* if for any $G \in R^\omega$, there is at most one initial vertex per level, and the targets of any pair of arcs with the same source and label have distinct levels: $\circ s, \circ t \in G \vee (r \xrightarrow[G]{a} s \wedge r \xrightarrow[G]{a} t) \implies s = t \vee \ell_G(s) \neq \ell_G(t)$.

In other words, R is level-deterministic if and only if there exists no pair of level-synchronized initial paths in $R^\omega_\circ$. So any grammar generating a deterministic graph is level-deterministic. We state another property oflevel-deterministic grammars.

LEMMA 10. *Any level-deterministic and level-unambiguous grammar is unambiguous.*

Another advantage of level-deterministic grammars is that the synchronization relation is recursive when the synchronizer is level-deterministic (this is proved using the generalized grammar synchronization product defined in the next section).

LEMMA 11. *We can decide whether $R \rhd S$ for R level-deterministic.*

Similarly to way level synchronization is done, we perform the standard powerset construction only level by level.

For R a grammar generating G, let $\Pi := \{P \mid \emptyset \neq P \subseteq V_G \wedge \forall p, q \in P, \ell(p) = \ell(q)\}$ be the set of subsets of vertices with same level, and let $\mathrm{Succ}_a(P)$ be the set of successors of vertices in $P \in \Pi$ by $a \in F_G \cap F_2$: $\mathrm{Succ}_a(P) := \{q \mid \exists p \in P(p \xrightarrow[G]{a} q)\}$. The *level-determinization* of any grammar R is defined as $\mathrm{Det}(R^\omega) := \{\mathrm{Det}(G) \mid G \in R^\omega\}$, where $\mathrm{Det}(G)$ is:

$$\begin{aligned}\mathrm{Det}(G) :=& \{P \xrightarrow{a} Q \mid P, Q \in \Pi \wedge Q \subseteq \mathrm{Succ}_a(P) \wedge \forall q \in \mathrm{Succ}_a(P) - Q, Q \cup \{q\} \notin \Pi\} \\ &\cup \{\circ P \mid P \in \Pi \wedge \forall p \in P(\circ p \in G) \wedge \forall q(\circ q \in G \wedge q \notin P \implies P \cup \{q\} \notin \Pi)\} \\ &\cup \{cP \mid P \in \Pi \wedge c \in F_1 - \{\circ\} \wedge \exists p \in P(cp \in G)\}\end{aligned}$$

restricted to the vertices accessible from $\circ$.

Contrary to the level synchronization product, Det does not preserve regularity. However $\mathrm{Det}(R^\omega)$ can be generated by a grammar when R is in a certain normal form which preserves synchronised languages.

Let us define an *arc grammar* R as an initial grammar whose rules (except the axiom rule) are all of the form $A12 \longrightarrow H_A$ where H_A is a finite graph with no terminal arc of target 1, or of source 2, or of source 1 and target 2 : $s \xrightarrow[[H_A]]{a} t \implies s \neq 2 \wedge t \neq 1 \wedge (s,t) \neq (1,2)$. We tranform a grammar into an arc grammar by splitting non-terminal hyperarcs into non-terminal arcs of arity 2 (hence the name).

LEMMA 12. *Any initial grammar can be transformed into a bi-synchronized arc grammar, while preserving unambiguity.*

This lemma allows to prove Proposition 1 by translating any unambiguous arc grammar R into an unambiguous context-free grammar generating $L(R)$, and conversely.

For any arc grammar R, $\mathrm{Det}(R^\omega)$ can be generated by a grammar $\mathrm{Det}(R)$ that we define. Let R be an arc grammar generating a graph accessible from $\circ$. To any $A \in N_R - \{Z\}$, we

associate a new symbol $\overline{A}$ of arity 2 and we define the grammar $\overline{R}$ obtained from R by adding the rules $\overline{A}12 \longrightarrow H_A$ for all $A \in N_R - \{Z\}$, and then by replacing in the right hand sides any non-terminal arc $s \xrightarrow{B} 2$ by $s \xrightarrow{\overline{B}} 2$:

$$\begin{aligned}\overline{R} := \{(Z, H_Z)\} \cup \{\big(A12, (H_A - N_R V_{H_A} 2) \cup \{\overline{B}s2 \mid B \in N_R \wedge Bs2 \in H_A\}\big) \mid A \in N_R - \{Z\}\} \\ \cup \{\big(\overline{A}12, (H_A - N_R V_{H_A} 2) \cup \{\overline{B}s2 \mid B \in N_R \wedge Bs2 \in H_A\}\big) \mid A \in N_R - \{Z\}\}.\end{aligned}$$

Let $<$ be a linear order over $2^{N_{\overline{R}} - \{Z\}}$ of smallest element $\emptyset$. For each $P \subseteq N_{\overline{R}} - \{Z\}$, $P \neq \emptyset$, we take a new symbol P' of arity $2^{|P|}$ and a hyperarc $<P> = P' p_1 \dots p_m$ with $\{p_1, \dots, p_m\} = 2^P$ and $p_1 < \dots < p_m$, and we define a graph H_P such that $\{Z \xrightarrow{A} A \mid A \in P\} \cup \{\circ Z\} \underset{\overline{R}}{\Longrightarrow} H_P$. In the special case where $P = \emptyset$, we let $<\emptyset> = Z$ and $H_\emptyset = H_Z$.

For every $P \subseteq N_{\overline{R}} - \{Z\}$, we apply to H_P the level-determinization procedure described above to get the graph $H'_P := \mathrm{Det}(H_P)[\emptyset/\{Z\}] - \{\circ\emptyset\}$ whose vertex level mapping ℓ is defined by $\ell(A) = 0$ for all $A \in P - N_R$, $\ell(A) = 1$ for all $A \in P \cap N_R$ and $\ell(s) = 2$ for all $s \in V_{H_P} - (P \cup \{Z\})$. Note that the level $\ell(Z)$ of Z is not significant because there is no arc of target Z in H_P. We define grammar $Det(R)$ by associating to each $P \subseteq N_{\overline{R}} - \{Z\}$ the rule:

$$<P> \quad \longrightarrow \quad [H'_P] \cup \{<Q>[s/\emptyset][\cup_{e \in E} s_e / E]_{\emptyset \neq E \subseteq Q} \mid s \in V_{H'_P} \wedge Q \neq \emptyset\}$$

with $Q := \{A \in N_{\overline{R}} \mid s \xrightarrow[H'_P]{A}\}$ and $s \xrightarrow[H'_P]{A} s_A$ for any $A \in Q$. Note that when R is unambiguous, we can restrict $<P> = P' p_1 \dots p_m$ to $\{p_1, \dots, p_m\} = P$.

LEMMA 13. *For any arc grammar R, $(\mathrm{Det}(R))^\omega = \mathrm{Det}(R^\omega)$, $\mathrm{Det}(R) \bowtie R$ and $Det(R)$ is level-deterministic, hence $\mathrm{Det}(R)$ is unambiguous for R level-unambiguous.*

5.3 Closure under complement

We now consider the closure under complement of $\mathrm{Sync}(R)$ for R level-unambiguous.

First we have to extend the level synchronization product $R \times S$ of any grammars R and S in order to retain a path for all the words accepted by R. We take new colours $\bullet^1, \bullet^2$ and a fresh symbol $\perp$. For any grammars R and S, the *generalized level synchronization product* of their generated graphs is $R^\omega \otimes S^\omega := \{G \otimes H \mid G \in R^\omega \wedge H \in S^\omega\}$, where $G \otimes H$ is defined as:

$$\begin{aligned}G \otimes H := & G \times H \cup \{\bullet^1(s, \perp) \mid \bullet s \in G\} \cup \{\bullet^2(\perp, p) \mid \bullet p \in H\} \\ & \cup \{(s,p) \xrightarrow{a} (t, \perp) \mid s \xrightarrow[G]{a} t \wedge ((s,p) \in V_{G \times H} \vee p = \perp) \wedge \forall q (p \xrightarrow[H]{a} q \Longrightarrow \ell(q) \neq \ell(t))\} \\ & \cup \{(s,p) \xrightarrow{a} (\perp, q) \mid p \xrightarrow[H]{a} q \wedge ((s,p) \in V_{G \times H} \vee s = \perp) \wedge \forall t (s \xrightarrow[G]{a} t \Longrightarrow \ell(t) \neq \ell(q))\} \\ & \cup \{\circ(s, \perp) \mid \circ s \in G \wedge \forall p (\circ p \in H \Longrightarrow \ell(p) \neq \ell(s))\} \\ & \cup \{\circ(\perp, p) \mid \circ p \in H \wedge \forall s (\circ s \in G \Longrightarrow \ell(s) \neq \ell(p))\}.\end{aligned}$$

The definition of $R \times S$ from the previous section is extended to define a grammar $R \otimes S$. The symbol $[A, B, E]$ is now of arity $|E| + \varrho(A) + \varrho(B)$ with the definition

$$[Ar_1 \dots r_m, Bs_1 \dots s_n, E] := [A, B, E](r_1, s_1)_E \dots (r_m, s_n)_E (r_1, \perp) \dots (r_m, \perp)(\perp, s_1) \dots (\perp, s_n)$$

and we replace $\big([P]\times[Q]\big)_{|\overline{E}}$ by $\big([P]\otimes[Q]\big)_{|\overline{E}\,\cup\, V_P\times\{\bot\}\,\cup\,\{\bot\}\times V_Q}$ in the right hand side of the rule of $[AX, BY, E]$. The grammar $R\otimes S$ generates $R^\omega\otimes S^\omega$, and satisfies:

$$(R\otimes S)_{\bullet,\bullet_1,\bullet^1}\bowtie R \quad ; \quad (R\otimes S)_{\bullet,\bullet_2,\bullet^2}\bowtie S \quad ; \quad \forall f\in\{\bullet,\bullet_1,\bullet_2\}, (R\otimes S)_f\bowtie(R\times S)_f.$$

The language $L(R)-L(S)$ for $S\lhd R$ is the set of non accepting words labelling initial paths in $R\otimes S$ which end in a vertex coloured by $\bullet_1$ or $\bullet^1$:

$$L(R)-L(S)=L(R)-(L(R)\cap L(S))=L((R\otimes S)_{\bullet,\bullet_1,\bullet^1})-L(R\otimes S)=L((R\otimes S)_{\bullet_1,\bullet^1})-L(R\otimes S)$$

When $(R\otimes S)_{\bullet,\bullet_1,\bullet^1}$ is unambiguous, the language $L((R\otimes S)_{\bullet_1,\bullet^1})-L(R\otimes S)$ is the set of words which label paths ending in non final vertices coloured by $\bullet_1$ or $\bullet^1$. As $(R\otimes S)_{\bullet,\bullet_1,\bullet^1}$ is level-unambiguous when R is, we get the closure under complement of $\mathrm{Sync}(R)$ using Lemmas 12 and 13.

LEMMA 14. *For R level-unambiguous and $S\lhd R$, $L(R)-L(S)\ \in\mathrm{Sync}(R)$.*

6 Conclusion

For lack of space, we had to omit from this paper a condition on grammars ensuring that their synchronized languages are closed under concatenation and Kleene star. Many other examples of grammars and their families of synchronized languages also have to be studied.

Acknowledgements. Many thanks to Antoine Meyer for helping me prepare the final version of this paper.

References

[AM 04] R. ALUR and P. MADHUSUDAN *Visibly pushdown languages*, 36^{th} STOC, ACM Proceedings, L. Babai (Ed.), 202–211 (2004).

[Ca 07] D. CAUCAL *Deterministic graph grammars*, Texts in Logic and Games 2, Amsterdam University Press, J. Flum, E. Grädel, T. Wilke (Eds.), 169–250 (2007).

[Ca 06] D. CAUCAL *Synchronization of pushdown automata*, 10^{th} DLT, LNCS 4036, O. Ibarra, Z. Dang (Eds.), 120-132 (2006).

[CH 08] D. CAUCAL and S. HASSEN *Synchronization of grammars*, 3^{rd} CSR, LNCS 5010, E. Hirsch, A. Razborov, A. Semenov, A. Slissenko (Eds.), 110–121 (2008).

[Fr 77] E. FRIEDMAN *Equivalence problems for deterministic context-free languages and monadic recursion schemes*, JCSS 14, 344–359 (1977).

[Gi 66] S. GINSBURG *The mathematical theory of context free languages*, McGraw-Hill (1966).

[LMM 08] N. LIMAYE, M. MAHAJAN and A. MEYER *On the complexity of membership and counting in height-deterministic pushdown automata*, 3^{rd} CSR, LNCS 5010, E. Hirsch, A. Razborov, A. Semenov, A. Slissenko (Eds.), 240–251 (2008).

[MS 85] D. MULLER and P. SCHUPP *The theory of ends, pushdown automata, and second-order logic*, Theoretical Computer Science 37, 51–75 (1985).

[NS 07] D. NOWOTKA and J. SRBA *Height-deterministic pushdown automata*, 32^{nd} MFCS, LNCS 4708, L. Kucera, A. Kucera (Eds.), 125–134 (2007).

[Wi 04] K. WICH *Ambiguity functions of context-free grammars and languages*, PhD Thesis, Universität Stuttgart (2004).