Deterministic graph grammars

Didier Caucal

Deterministic graph grammars

Didier Caucal[1]

[1] IGM–CNRS, university of Paris-Est
5 Bd Descartes
77454 Marne-la-Vallée, France
caucal@univ-mlv.fr

1 Introduction

Context-free grammars are one of the most classical and fundamental notions in computer science textbooks, in both theoretical and applied settings. As characterizations of the well-known class of context-free languages, they are a very prominent tool in the field of language theory. Since context-free grammars are powerful enough to express most programming languages, they also play an important role in compilation, where they form the basis of many efficient parsing algorithms.
A similar notion can be adapted to the more general setting of grammars generating graphs instead of words. In this case, grammar rules no longer express the replacement of a non-terminal letter by a string of terminal and non-terminal letters, but that of a non-terminal arc (or more generally hyperarc) by a finite graph (or hypergraph) possibly containing new non-terminals, thus generating larger and larger graphs. It is still relevant to call such grammars context-free, since the replacement of a given non-terminal is independent of the context in which it is performed, *i.e.* the remainder of the graph it is applied to, which is left unchanged. Also, whenever two non-terminals can be replaced, the corresponding derivation steps are independant. Consequently, starting from a given graph, it is possible to describe any sequence of productions (a derivation) as a derivation tree. This intuitively explains why many notions suitable for the study of context-free word grammars extend to context-free (also called hyperedge-replacement) graph grammars (see for instance [Ro 97]).

In this paper, we are concerned with the specific setting where the considered sets of grammar rules are deterministic, meaning that there is only one production rule for every non-terminal hyperarc. Consequently, from a given axiom, a grammar does not generate a set of graphs (which could be called a 'context-free' graph language), but a unique graph up to isomorphism called a regular graph. This is an important restriction, which entails another crucial conceptual difference with word grammars. Note

that grammars generating a unique finite graph are trivial: they are equivalent to grammars containing a unique rule, or even no rule if any finite graph is allowed as an axiom. As a result and contrary to the case of words, we are not interested in graphs generated after a finite derivation sequence, but in graphs generated 'at the limit' *i.e.* after an infinite number of steps (see Figure 2.8 and Figure 2.9).
These deterministic graph grammars correspond to the finite systems of equations over graph operators originally defined by Courcelle [Co 89], and whose least solutions, called equational graphs, are the regular graphs. This kind of graphs was first considered by Muller and Schupp [MS 85] : they showed that the connected components of the transition graphs of pushdown automata are the connected graphs of finite degree whose decomposition by distance from a vertex yields finitely many non-isomorphic connected components. These graphs are exactly the connected regular graphs of finite degree [Ca 90] (see also Section 5).

This work is a first attempt at a general survey of deterministic graph grammars and the class of graphs they generate. We focus on providing some of the basic tools to reason about deterministic graph grammars, and on a structural study of their generated graphs.
First, Section 2 presents the necessary definitions as well as some examples of grammars and their generated graphs. We also define a canonical representant of the set of isomorphic graphs generated by a given grammar.
Second, as is the case for word grammars, we need to provide a collection of normal forms before being able to conveniently write more involved proofs. This is a slightly tedious but necessary task, which is addressed in Section 3, where in particular the notions of reduced, proper and connected grammars are defined. We provide a way to cleanly separate input and output vertices in grammar rules. We also show that considering multi-hypergraphs does not improve expressiveness. All these results are obtained via fixed-point computations. This allows us, as a first application, to derive some structural properties of regular graphs, namely that they only have a finite number of non-isomorphic connected components, and that the sets of possible vertex degrees in such graphs are finite.
A problematic feature of regular graphs is that any given such graph can be generated by infinitely many different graph grammars. In Section 4, we investigate systematic ways to generate regular graphs, for instance according to the length of their vertex names for pushdown graphs, or more generally, by increasing distance from the vertices having a given colour. This yields a notion of a canonical graph grammar associated to any regular graph (which will prove useful in the following section). It also allows us to establish the closure of the class of regular graphs under various vertex colouring operations.

Section 5 builds up on all the notions and results presented in the previous sections to establish a characterization of regular graphs of bounded degree, either in a general way by the suffix transition graphs of labelled word rewriting systems, or in a restrictive way by the transition graphs of pushdown automata in a weak form.
Finally in Section 6, we present a simple and strong connection between deterministic graph grammars and context-free grammars over words, and hence also context-free word languages: even though regular graphs may in general have an infinite degree, the set of path labels between two regular sets of vertices in a regular graph remains a context-free language. In this respect, deterministic graph grammars provide a natural and powerful tool to reason about context-free languages, and indeed several classical results in the theory of context-free languages can be reassessed in this framework.
To summarize, deterministic graph grammars are not only finite representations of infinite graphs whose structure is regular (*i.e.* which have a finite decomposition by distance), they are also to context-free languages what finite automata are to regular languages.

Contents

2 Regular graphs

In this section, we introduce the notion of deterministic graph grammar (Section 2.2) together with the family of graphs they generate: the regular graphs (Section 2.3). We conclude by presenting several examples of regular graphs. But first, we introduce basic notations on graphs and hypergraphs (Section 2.1).

2.1 Graphs

Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{N}_+ = \mathbb{N} - \{0\}$. A set in bijection with $\mathbb{N}$ is called *countable*. For a set E, we write $|E|$ its cardinal, 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*: $(e_1, \ldots, e_m) \cdot (e'_1, \ldots, e'_n) = (e_1, \ldots, e_m, e'_1, \ldots, e'_n)$, and 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.

A *multi-subset* M of E is a mapping from E into $\mathbb{N}$ where for any $e \in E$, the integer $M(e)$ is its *multiplicity* (the number of occurrences of e in M). A multi-subset M of E is also represented by the functional subset $\{ (e, M(e)) \mid e \in E \wedge M(e) \neq 0 \}$ of $E \times \mathbb{N}_+$: if $(e, m), (e, n) \in M$ then $m = n$. The *cardinal* of M is $|M| = \sum_{e \in E} M(e)$, and M is said to be finite if its *support* $\widehat{M} := \{ e \in E \mid M(e) \neq 0 \}$ is finite. By extension we write $e \in M$ for $e \in \widehat{M}$. A finite multi-subset M can also be described by a subset of E where each $e \in E$ appears $M(e)$ times. For instance the multi-subset defined by $a \mapsto 3$, $b \mapsto 1$, $x \mapsto 0$ otherwise, is represented by $\{(a, 3), (b, 1)\}$ or directly by $\{a, a, a, b\}$. For instance $\{2, 2, 2, 5\}$ is the multi-subset of the decomposition into prime factors of 40. A subset $P \subseteq E$ corresponds to the multi-subset $\{ (e, 1) \mid e \in P \}$ and vice-versa.

Given multi-subsets M and N, we define the multi-subset

sum	$M+N$	by $(M+N)(e) := M(e)+N(e)$,
difference	$M-N$	by $(M-N)(e) := max\{M(e)-N(e),0\}$,
union	$M \cup N$	by $(M \cup N)(e) := max\{M(e),N(e)\}$,
intersection	$M \cap N$	by $(M \cap N)(e) := min\{M(e),N(e)\}$,
restriction	$M_{\mid P}$	to $P \subseteq E$ by $M_{\mid P}(e) := \begin{cases} M(e) & \text{if } e \in P, \\ 0 & \text{otherwise;} \end{cases}$ we will also write $M_{\mid -P}$ for $M_{\mid E-P}$.

The *inclusion* $M \subseteq N$ means that $M(e) \leq N(e)$ for every $e \in E$.
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* $\varrho(f)$, and such that

$$F_n := \{\ f \in F \mid \varrho(f) = n\ \} \text{ is countable for every } n \geq 0.$$

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

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

This is illustrated in the next figure.

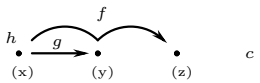


Figure 2.1. The hypergraph $\{fxyz\,,\, gxy\,,\, hx\,,\, c\}$.

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).
Note that a hyperarc X is a word whose first letter $X(1)$ is its label, and for $1 < i \leq |X|$, the i-th letter $X(i)$ is its $(i-1)$-th vertex; to avoid such a shift, we also write a hyperarc as the word fY where f is its label and Y is its vertex word.
Observe that a hypergraph is finite if and only if it has a finite vertex set.
The transformation of a hypergraph G by a function h from V_G into a set V is the following hypergraph:

$$h(G) := \{\ fh(v_1)\ldots h(v_{\varrho(f)}) \mid fv_1 \ldots v_{\varrho(f)} \in G\ \}.$$

An *isomorphism* h from a hypergraph G to a hypergraph H is a bijection from V_G to V_H such that $h(G) = H$, and we write $G \overset{h}{\sim} H$ or $G \sim H$ if we do not specify the bijection.
The *restriction* of a hypergraph G to a subset $P \subseteq V_G$ is the sub-hypergraph of G induced by P:

$$G_{|P} \;:=\; G \cap FP^* \,.$$

So $G_{|P} = \mathrm{Id}_P(G)$ where $\mathrm{Id}_P := \{\, (v,v) \mid v \in P \,\}$ is the *identity* on P.
For a hypergraph G, the *edge relation* $\underset{G}{\longleftrightarrow}$ is the binary relation on the vertex set V_G defined by

$$X(i) \underset{G}{\longleftrightarrow} X(j) \quad \text{for any} \quad X \in G \;\text{ and }\; i \neq j \in \{2,\ldots,|X|\}\,.$$

We denote by $\underset{G}{\longleftrightarrow}^n$ with $n \geq 0$ the n-fold composition of $\underset{G}{\longleftrightarrow}$, with $\underset{G}{\longleftrightarrow}^0 := \mathrm{Id}_{V_G}$ the identity on V_G, and by $\underset{G}{\longleftrightarrow}^* := \bigcup_{n\geq 0} \underset{G}{\longleftrightarrow}^n$ the reflexive and transitive closure of $\underset{G}{\longleftrightarrow}$. As usual s and t are *connected vertices* in G if $s \underset{G}{\longleftrightarrow}^* t$, and G is a *connected hypergraph* if the vertices of G are connected.
The *degree* of a vertex s of a hypergraph G is

$$d_G(s) \;:=\; |\{\, (X,i) \mid X \in G - F_1\, V_G \;\wedge\; 2 \leq i \leq |X| \;\wedge\; X(i) = s \,\}|\,.$$

Note that the colouring does not affect the degree.
We say that a hypergraph G is of *finite degree* (or *locally finite*) if $d_G(s) < \omega$ for any vertex $s \in V_G$, and G is of *bounded degree* (or *globally finite*) if $max\{\, d_G(s) \mid s \in V_G \,\} < \omega$.
For a subset $E \subseteq F$ of labels, we write

$$V_{G,E} \;:=\; \{\, v \in V \mid EV^*vV^* \cap G \neq \varnothing \,\} \;=\; V_G \cap EV_G^*$$

the set of vertices of G linked by a hyperarc labelled in E.
A *graph* G is a hypergraph without constants and without labels of arity strictly greater than 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 $v_1 \underset{G}{\overset{a}{\longrightarrow}} v_2$ or directly $v_1 \overset{a}{\longrightarrow} v_2$ if G is understood, plus a set of coloured vertices $f\,v$. For instance, the finite graph:

$$\{r \overset{b}{\longrightarrow} p\,,\; p \overset{a}{\longrightarrow} s\,,\; p \overset{b}{\longrightarrow} q\,,\; q \overset{a}{\longrightarrow} p\,,\; q \overset{b}{\longrightarrow} s\,,\; i\,r\,,\; g\,p\,,\; h\,p\,,\; f\,s\,,\; f\,t\}$$

has vertices p,q,r,s,t, colours f,g,h,i and arc labels a,b, and is represented below; we omit the names of the vertices to give a representation up to isomorphism.

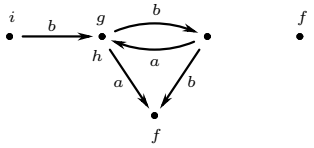


Figure 2.2. A finite graph.

A tuple $(v_0, a_1, v_1, \ldots, a_n, v_n)$ for $n \geq 0$ and $v_0 \underset{G}{\overset{a_1}{\longrightarrow}} v_1 \;\ldots\; v_{n-1} \underset{G}{\overset{a_n}{\longrightarrow}} v_n$ is a *path* from v_0 to v_n labelled by $u = a_1 \ldots a_n$; we write $v_0 \underset{G}{\overset{u}{\Longrightarrow}} v_n$ or directly $v_0 \overset{u}{\Longrightarrow} v_n$ if G is understood. For $E \subseteq F_2^*$, we write $v \underset{G}{\overset{E}{\Longrightarrow}} v'$ if $v \underset{G}{\overset{u}{\Longrightarrow}} v'$ for some $u \in E$.

Given a graph G and vertex sets $P, Q \subseteq V_G$, we write $L(G,P,Q)$ the language of path labels from a vertex in P to a vertex in Q:

$$L(G,P,Q) \;:=\; \{\; u \in F_2^* \mid \exists\, p \in P\ \exists\, q \in Q \;\; p \underset{G}{\overset{u}{\Longrightarrow}} q \;\}\,.$$

Given colours $i, f \in F_1$, we define $L(G,i,f) \;:=\; L(G, V_{G,i}, V_{G,f})$ as the path labels from the set $V_{G,i}$ of vertices coloured by i to the set $V_{G,f}$ of vertices coloured by f.
For instance taking the previous graph, its path labels from i to f is $b(ba)^*(a+bb)$.
Hence a finite graph G with two colours i and f is a *finite automaton* recognizing the language $L(G,i,f)$. For any (finite) alphabet $T \subset F_2$, the family

$$\mathrm{Rat}(T^*) \;:=\; \{\; L(G,i,f) \mid |G| < \omega \;\wedge\; F_G \cap F_2 \subseteq T \;\wedge\; i,f \in F_1 \;\}$$

of languages over T recognized by the finite automata coincides with the family of *regular languages* over T.
A graph G without vertex label *i.e.* such that $F_G \subset F_2$ is called an *uncoloured graph*.

The family of hypergraphs ordered by inclusion $\subseteq$ forms a complete partial order: its least element is the empty graph $\varnothing$ and any sequence $(G_n)_{n\,\geq\,0}$ (not necessarily increasing) with a finite label set $\bigcup_{n\geq 0} F_{G_n}$ has a least upper bound $\bigcup_{n\geq 0} G_n$.
If we fix a finite or countable set V of vertices and a finite set $E \subset F$ of labels, the family $\mathcal{G}(V,E)$ of subsets of $\bigcup_{n\geq 0} E_n V^n$ with $E_n = E \cap F_n$ for any $n \geq 0$, is the set of hypergraphs G with $V_G \subseteq V$ and $F_G \subseteq E$. Such a set $\mathcal{G}(V,E)$ is a complete lattice: $\varnothing$ is the least element, $\bigcup_{n\geq 0} E_n V^n$ is the greatest element, and every subset $\mathcal{H} \subseteq \mathcal{G}(V,E)$ has a supremum $\bigcup \mathcal{H}$ and an infimum $\bigcap \mathcal{H}$.

A *multi-hypergraph* G is a multi-subset of $\bigcup_{n\geq 0} F_n V^n$ where V is an arbitrary set; each hyperarc $X \in G$ is depicted $G(X)$ times.

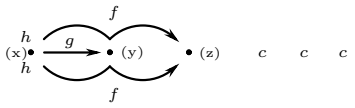


Figure 2.3. The multi-hypergraph $\{fxyz\,,\ fxyz\,,\ gxy\,,\ hx\,,\ hx\,,\ c\,,\ c\,,\ c\}$.

The vertex set V_G and the label set F_G of a multi-hypergraph G are the sets defined on its support $\widehat{G}$ *i.e.* $V_G{:=}V_{\widehat{G}}$ and $F_G{:=}F_{\widehat{G}}$.
The transformation of any multi-graph G by any function h from V_G into a set is extended in a natural way:

$$h(G)(X) := \textstyle\sum_{h(Y)=X} G(Y) \quad \text{for any hyperarc } X$$

assuming that the sum is always finite.
Given $f \in F_1$ and $v \in V$, the sequence $\{(fv,n)\}_{n\geq 1}$ is increasing for the

inclusion but it has no least upper bound because an infinite multiplicity is not allowed.

2.2 Graph grammars

A hypergraph *grammar* R is a finite set of rules of the form $fx_1 \dots x_{\varrho(f)} \longrightarrow H$ where $fx_1 \dots x_{\varrho(f)}$ is a hyperarc joining pairwise distinct vertices $x_1 \neq \dots \neq x_{\varrho(f)}$ and H is a finite multi-hypergraph; we denote by

$$\begin{array}{rcl} N_R & := & \{\ f \in F \mid fX \in Dom(R)\ \} \text{ the } \textit{non-terminals} \text{ of } R\text{:} \\ & & \quad \text{the labels of the left hand sides,} \\ T_R & := & \{\ f \in F - N_R \mid \exists\ P \in Im(R),\ fX \in P\ \} \text{ the } \textit{terminals} \text{ of } R, \\ & & \quad \text{the labels of } R \text{ which are not non-terminals,} \\ F_R & := & N_R \cup T_R \text{ the labels of } R, \\ \varrho(R) & := & \max\{\ \varrho(A) \mid A \in N_R\ \} \text{ the } \textit{arity} \text{ of } R, \\ & & \quad \text{the maximal arity of its non-terminals.} \end{array}$$

We use grammars to generate simple graphs (without multiplicity). Hence in the following, we may assume that any terminal hyperarc of any right hand side is of multiplicity 1, otherwise we replace R by

$$\{\ (X, <H>) \mid (X, H) \in R\ \}$$

where $<H>$ is obtained from H by removing the multiplicity of the terminal hyperarcs:

$$<H> \ := \ H_{|N_R V_H^*} \cup (H \cap T_R V_H^*)\,.$$

Remark that multiplicities of non-terminal hyperarcs are usually not introduced when working with graph grammars. As explained in the next subsection, they are in all generality necessary to ensure the unicity of the graph generated (see also Figure 2.6). In the next section, we will see that any graph grammar can be transformed into an equivalent grammar where multiplicities do not need to be taken into account.

Starting from any hypergraph, we want a grammar to generate a unique hypergraph up to isomorphism. So we restrict ourselves to *deterministic* grammars, meaning that there is only one rule per non-terminal:

$$(X, H)\,, (Y, K) \in R \ \wedge \ X(1) = Y(1) \quad \Longrightarrow \quad (X, H) = (Y, K).$$

For any rule $X \longrightarrow H$, we say that

$$\begin{array}{rcl} & V_X \cap V_H & \text{are the } \textit{inputs} \text{ of } H \\ \text{and} & \bigcup\{\ V_Y \mid Y \in H \ \wedge \ Y(1) \in N_R\ \} & \text{are the } \textit{outputs} \text{ of } H. \end{array}$$

We will use upper-case letters $A, B, C, \dots$ for non-terminals and lower-case letters $a, b, c \dots$ for terminals. We say that R is a *graph grammar* if the terminals are of arity 1 or 2. An example is given below.

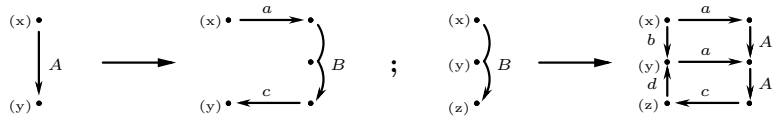


Figure 2.4. A (deterministic) graph grammar.

For the grammar R of Figure 2.4, we have

$$N_R = \{A, B\}\ ,\ T_R = \{a, b, c, d\}\ ,\ \varrho(R) = 3,$$

and the inputs of the first (resp. second) rule are x, y (resp. x, y, z).

Given a grammar R, the *rewriting* $\underset{R}{\longrightarrow}$ is the binary relation between multi-hypergraphs defined as follows: M rewrites into N, written $M \underset{R}{\longrightarrow} N$, if we can choose a non-terminal hyperarc $X = As_1 \ldots s_p$ in M and a rule $Ax_1 \ldots x_p \longrightarrow H$ in R such that N can be obtained by replacing X by H in M and by removing the multiplicity of terminal hyperarcs:

$$N = <(M - X) + h(H)>$$

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 multi-subset E of non-terminal hyperarcs. A *complete parallel rewriting* $\underset{R}{\Longrightarrow}$ is the rewriting according to the multi-subset of all non-terminal hyperarcs: $M \underset{R}{\Longrightarrow} N$ if $M \underset{R,\,E}{\longrightarrow} N$ where E is the multi-subset of all non-terminal hyperarcs of M.

For instance, the first three steps of the parallel derivation from the graph $\{Axy\,,\,1x\,,\,2y\}$ according to the grammar of Figure 2.4 are depicted in the figure below.

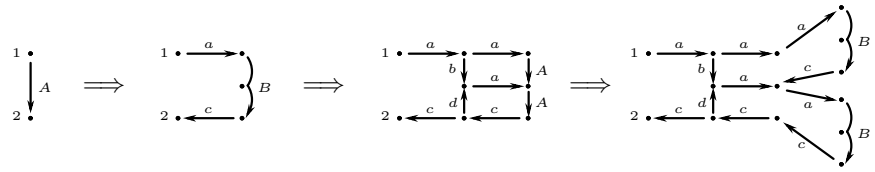


Figure 2.5. Parallel rewritings according to the grammar of Figure 2.4 .

The *derivation* $\underset{R}{\Longrightarrow}^*$ is the reflexive and transitive closure for composition of the parallel rewriting $\underset{R}{\Longrightarrow}$ (*i.e.* $G \underset{R}{\Longrightarrow}^* H$ if H is obtained from G by a consecutive sequence of parallel rewritings).

We can now define the graphs generated by deterministic graph grammars.

2.3 Regular graphs

Intuitively the graph (up to isomorphism) generated by a deterministic graph grammar R from a finite graph G_0 is the limit of any infinite se-

quence of rewritings starting from G_0 where every non-terminal is eventually rewritten. Formally to a sequence $\big(G_i\big)_{i\geq 0}$ of finite multi-hypergraphs such that for all $i \geq 0$, $G_i \underset{R,\, X_i}{\longrightarrow} G_{i+1}$ and
for all $X \in G_i$ with $X(1) \in N_R$, there exists $j \geq i$ such that $X = X_j$, we associate the limit $\bigcup_{i\geq 0}[G_i]$ where for M a hypergraph,

$$[M] \;:=\; M \cap T_R V_M^*$$

designates the (simple) set of terminal hyperarcs of M.
Note that the sequence $\big(G_i\big)_{i\geq 0}$ can be not increasing contrary to the sequence $\big([G_i]\big)_{i\geq 0}$; in particular, even if $\bigcup_{i\geq 0}[G_i]$ is finite, the sequence $\big(G_i\big)_{i\geq 0}$ is not necessarily ultimately constant.
It is easy to check that this limit does not depend on the order of the rewriting. In particular, we can use the parallel rewriting $\underset{R}{\Longrightarrow}$ which provides a canonical rewriting order similar to the leftmost rewriting for context-free grammars.
The example below illustrates that without multiplicities, the unicity of the limit graph no longer holds.

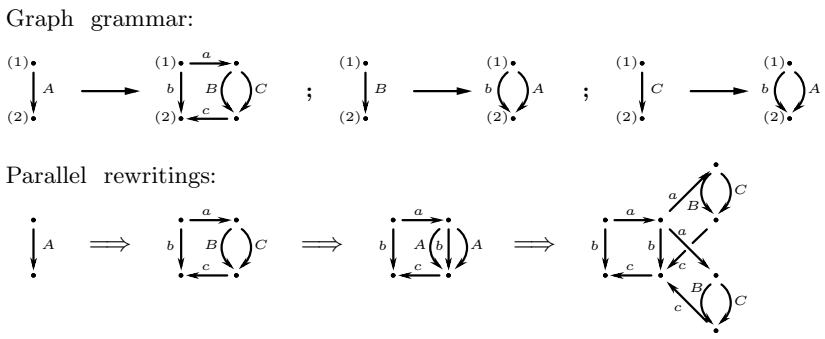


Figure 2.6. Parallel rewritings producing multiplicity.

We will see in next section that, though multiplicities are crucial in ensuring the unicity of the generated graph, they can be omitted provided that the grammar respects a certain normal form (see Subsection 3.2).
A hypergraph G is *generated by a grammar* R from a hypergraph H if G is isomorphic to a hypergraph in the following set:

$$R^\omega(H) \;:=\; \{\, \textstyle\bigcup_{n\geq 0}[H_n] \mid H_0 = H \;\wedge\; \forall\, n \geq 0,\; H_n \underset{R}{\Longrightarrow} H_{n+1} \,\};$$

note that the vertices of H appear in any hypergraph of $R^\omega(H)$.
For instance by continuing infinitely the derivation of Figure 2.5, we get a graph depicted in the next figure.

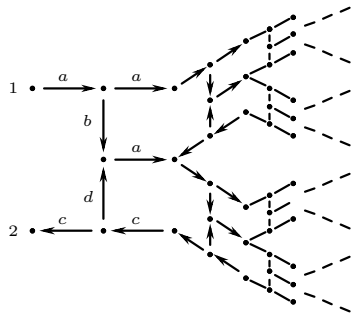


Figure 2.7. Graph generated by the grammar of Figure 2.4 .

Remark that the definition of $R^\omega(H)$ does not fix a particular naming of the vertices of the graph generated by R. A canonical naming is provided in Section 3.5 .

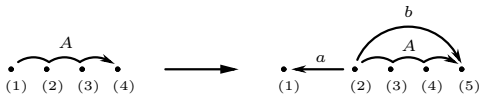
A *regular hypergraph* is a hypergraph generated by a (deterministic) grammar from a finite hypergraph.

The regular hypergraphs are the *hyperedge replacement equational hypergraphs* in the sense of [Co 89], which are defined as the smallest solutions of finite systems of equations involving a set of hypergraph operators.
A *regular graph* is a regular hypergraph which is a graph: it is generated by a graph grammar from a finite hypergraph. We give below some examples of regular graphs.

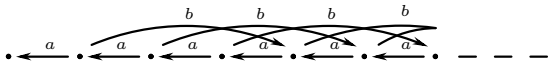
The grammar R reduced to the unique rule

$$A1234 \longrightarrow \{a21\,,\, b25\,,\, A2345\}$$

and represented below:



generates from its left hand side the following regular graph:



which can be drawn without crossing edges as the following regular graph:

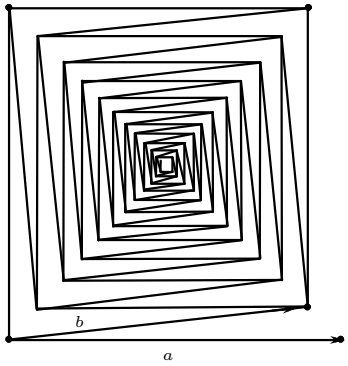
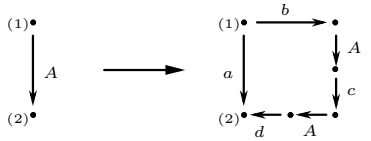


Figure 2.8. A regular graph.

Another example of graph grammar is the grammar reduced to the following rule:



generating from its left hand side the following regular graph:

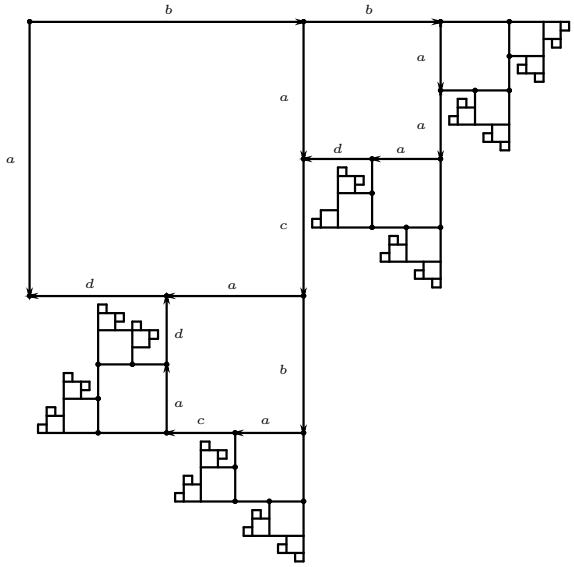
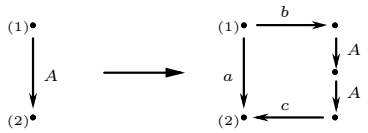
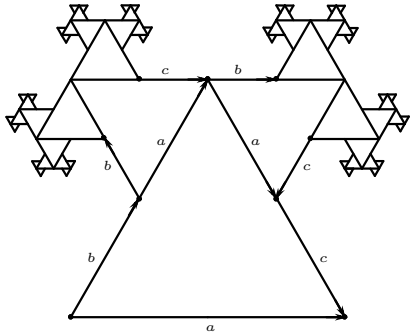


Figure 2.9. Another regular graph.

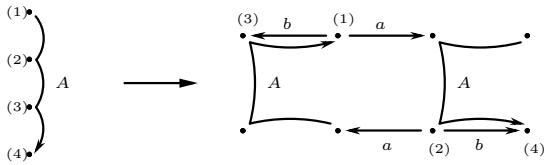
The grammar reduced to the following rule:



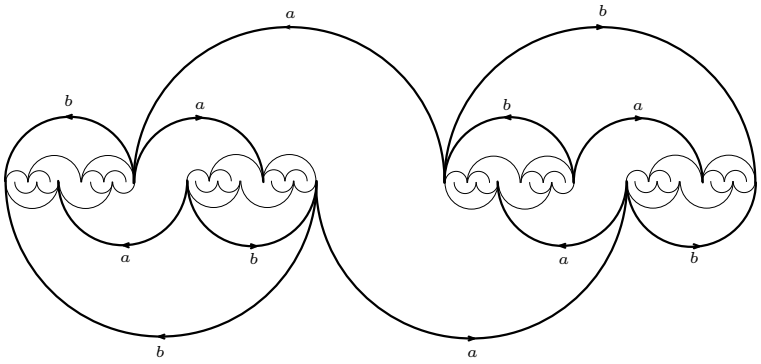
generates from its left hand side the following regular graph:



Finally the graph grammar reduced to the following rule:



generates from its left hand side the regular graph below, where each vertex is of infinite in-degree:



3 Normalizations of graph grammars

In this section, we present several elementary transformations to normalize hypergraph grammars. The first normalization gives an equivalent grammar with a constant axiom such that each non-terminal is accessible and generates a non empty graph which is connected except for the axiom and possibly another constant (cf. Proposition 3.5).
This normalization is extended to get ride of multiplicities both in the definition of the graph grammar and in its derivation relation. To ensure that multiplicities are not needed, we ask that any non-terminal hyperarc appearing in a right hand side of a rule contains a vertex which is not an input (cf. Proposition 3.10).
Finally we extend this second normalization by separating as much as possible for each right hand side the inputs and the outputs (cf. Theorem 3.12).
All these basic transformations are expressed in a powerful and natural way as fixed point computations. These normalizations are used to derive properties on the generated graphs: any regular graph has a finite number of non-isomorphic connected components, and a finite number of vertex degrees (cf. Propositions 3.4 and 3.13).
Finally we give a canonical vertex naming for the regular graphs (cf. Subsection 3.5).

3.1 Reduced and connected form

We begin by transforming any grammar into a reduced form.
We say that a grammar R is *reduced* if $R = \varnothing$ or there exists a constant non-terminal $Z \in Dom(R) \cap F_0$ called the *axiom* such that the following three conditions are satisfied:

(i) for all $H \in Im(R)$, $Z \notin F_H$

(ii) for all $A \in N_R$ there exists H such that $Z \underset{R}{\Longrightarrow}^* H$ and $A \in F_H$

(iii) $R^\omega(X) \neq \{\varnothing\}$ for every $X \in Dom(R)$;

the axiom is a non-terminal constant which by condition (i) does not appear in the right hand sides of the rules, condition (ii) means that each non-terminal is accessible from the axiom, and condition (iii) expresses that R generates a non empty hypergraph from any non-terminal hyperarc.
By condition (iii), the grammar $\varnothing$ (with no rule) is the unique reduced grammar generating the empty graph $\varnothing$.
By conditions (i) and (ii), a non empty reduced grammar has a unique axiom.
For instance the grammar of Figure 2.4 is not reduced, but it becomes reduced by adding the rule $Z \longrightarrow Axy$.
We say that a hypergraph G is *generated by a reduced grammar* R if $R = G = \varnothing$ or if the reduced grammar R is non empty and generates

G from its axiom.
Any regular graph can be generated by a reduced grammar.

Lemma 3.1. *Any regular hypergraph can be generated in an effective way by a reduced grammar.*

Proof. Let G be a hypergraph generated by a deterministic grammar R from a finite multi-hypergraph G_0.
We take a new constant $Z \in F_0 - F_R$ and we complete R into

$$\overline{R} := R \cup \{(Z, G_0)\}.$$

The set

$$E := \bigcup\{ F_K \cap N_{\overline{R}} \mid Z \underset{\overline{R}}{\Longrightarrow}^* K \}$$

of *accessible* non-terminals from Z is the least fixed point of the following equation:

$$E = \{Z\} \cup \{ Y(1) \in N_R \mid \exists (X,H) \in \overline{R},\ X(1) \in E \wedge Y \in H \}.$$

The set

$$\overline{E} := \{ X(1) \in E \mid \overline{R}^{\omega}(X) \neq \{\varnothing\} \}$$

of *productive* accessible non-terminals is the least fixed point of the following equation:

$$\overline{E} = \{ X(1) \in E \mid \exists P,\ (X,P) \in \overline{R} \wedge P \cap (\overline{E} \cup T_{\overline{R}})V_P^* \neq \varnothing \}.$$

The following grammar:

$$S := \{ (X, P_{|(\overline{E} \cup T_{\overline{R}})V_P^*}) \mid (X,P) \in \overline{R} \wedge X(1) \in \overline{E} \}$$

is reduced and generates G. Q.E.D. (Lemma 3.1)

The 'standard' construction in the proof of Lemma 3.1 is illustrated below.

Axiom: 

Grammar:

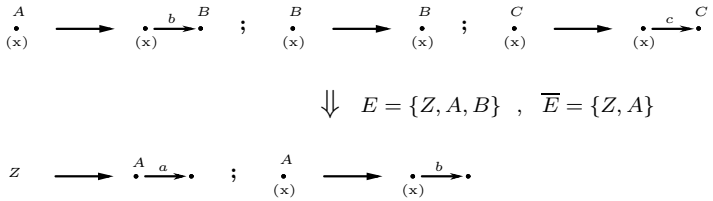


Figure 3.1. Reduction of a grammar.

Another form of grammar is to be *proper*:

$$V_X \subseteq V_G \text{ for all } X \in Dom(R) \text{ and for all } G \in R^{\omega}(X)$$

meaning that any vertex of the left hand side X of any rule, is a vertex of its right hand side and is a vertex of a terminal hyperarc of any graph

obtained by a derivation from X. For instance the grammar of Figure 2.4 is proper but the following grammar:

$$Axy \longrightarrow \{axz\,,\, Azy\}$$

is not proper because the vertex y of Axy does not belong to any graph of $R^\omega(Axy)$.

Lemma 3.2. *Any regular hypergraph can be generated in an effective way by a proper and reduced grammar.*

Proof. Let G be a regular hypergraph. We may assume $G \neq \varnothing$ because $\varnothing$ is a proper and reduced grammar generating $\varnothing$. By Lemma 3.1, G is generated by a reduced grammar R from its axiom Z.
For every rule $Ax_1 \ldots x_{\varrho(A)} \longrightarrow P_A$ in R, we define the set $Keep(A)$ of indices $1 \leq i \leq \varrho(A)$ such that x_i is *useful*: $x_i \in V_G$ for all $G \in R^\omega(Ax_1 \ldots x_{\varrho(A)})$.
This collection of sets $Keep(A)$ is the least fixed point of the following recursive system:

$$\begin{aligned} Keep(A) \quad := \quad & \{\, i \in [\varrho(A)] \mid P_A \cap T_R\, V_{P_A}^*\, x_i\, V_{P_A}^* \neq \varnothing \ \vee\ \exists\, BY \in P_A\,, \\ & B \in N_R \ \wedge\ \exists\, 1 \leq j \leq |Y|\,,\ Y(j) = x_i \ \wedge\ j \in Keep(B)\ \}\,. \end{aligned}$$

To each $A \in N_R$, we associate a new symbol A' of arity $|Keep(A)|$.
To each non-terminal hyperarc $Ay_1 \ldots y_{\varrho(A)}$ (with $A \in N_R$), we associate the following hyperarc:

$$h(Ay_1 \ldots y_{\varrho(A)}) \ := \ A' y_{i_1} \ldots y_{i_p}$$

with $\{i_1, \ldots, i_p\} = Keep(A)$ and $i_1 < \ldots < i_p$.
We complete h by the identity: $h(X){:=}X$ for any terminal hyperarc X.
Finally we extend h by union to any multi-hypergraph H: $h(H) := \{\, h(X) \mid X \in H\ \}$. We define the following grammar:

$$h(R) \ := \ \{\, (h(X), h(H)) \mid (X, H) \in R\ \}\,.$$

The grammar $h(R)$ is proper, reduced and generates G from its axiom $h(Z) = Z'$. Q.E.D. (Lemma 3.2)

The construction in the proof of Lemma 3.2 is illustrated in the figure below.

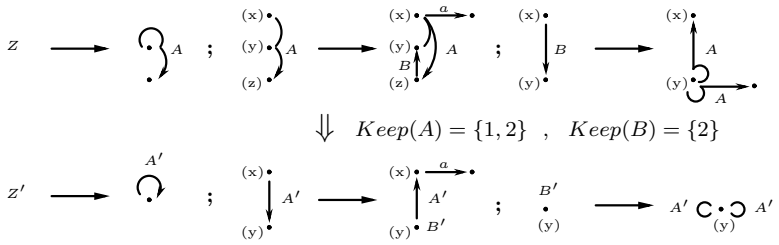


Figure 3.2. Transformation of a grammar into a proper grammar.

We now want to generate regular hypergraphs using grammars in two parts: a set of rules producing only connected graphs, and a set of rules whose left hand sides are constants.
Note that for any reduced grammar generating a connected hypergraph, the axiom is the unique non-terminal of null arity.
A *connected grammar* R is a proper grammar such that

for all $X \in Dom(R) - F_0$ and all $G \in R^\omega(X)$, G is connected.

In particular for every rule $(X, H) \in R$ with $X \notin F_0$, we have $H \cap F_0 = \varnothing$.
Let us extend Lemma 3.2.

Lemma 3.3. *Any regular hypergraph can be generated in an effective way by a connected and reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph.
By Lemma 3.2, G is generated by a proper and reduced grammar R from its axiom Z.
For every rule $Ax_1 \ldots x_{\varrho(A)} \longrightarrow H_A$ in R and for every $1 \leq i \leq \varrho(A)$, we associate the set $Con(A, i)$ of vertices in H_A which are connected to x_i in $R^\omega(Ax_1 \ldots x_{\varrho(A)})$.
This collection of sets $Con(A, i)$ is the least fixed point of the following recursive system:

$$\begin{aligned} Con(A,i) \quad = \quad & \{x_i\} \cup \bigcup\{\, V_X \mid X \in H_A \ \wedge\ X(1) \in T_R \\ & \qquad \wedge\ V_X \cap Con(A,i) \neq \varnothing \,\} \\ \cup \quad & \{\, X(j) \mid \exists\, Y \in Dom(R),\ Y(1)X \in H_A \\ & \qquad \wedge\ \exists\, k,\ X(k) \in Con(A,i) \ \wedge\ x_j \in Con(Y(1),k) \,\}. \end{aligned}$$

We complete these sets by defining for any $A \in N_R$ the set

$$Con(A) \ := \ \{\, Con(A,i) \mid 1 \leq i \leq \varrho(A) \,\} \cup \{\varnothing\}.$$

To each non-terminal hyperarc $X \in Dom(R)$ and to each $P \in Con(X(1))$, we associate a new symbol $X(1)_P$ of arity $|P \cap \{x_1, \ldots, x_{\varrho(X(1))}\}|$, and the hyperarc

$$X_P \ := \ X(1)_P x_{i_1} \ldots x_{i_p}$$

with $\{x_{i_1}, \ldots, x_{i_p}\} = P \cap \{x_1, \ldots, x_{\varrho(X(1))}\}$ and $i_1 < \ldots < i_p$.
In particular $X_\varnothing = X(1)_\varnothing$ is a constant.
This permits to define the following grammar:

$$I \ := \ \{\, (\, X\, ,\, \{\, X_P \mid P \in Con(X(1)) \,\}\,) \mid X \in Dom(R) \,\}$$

which splits each $X \in Dom(R)$ into hyperarcs according to $Con(X(1))$.
For each rule (X, H) of R, there is a unique hypergraph K_X such that $H \underset{I}{\Longrightarrow} K_X$, and we denote

$$\ll X \gg \ := \ V_{K_X} - \bigcup Con(X(1))$$

the set of vertices of H_X which are not connected to an input (a vertex in V_X). The following grammar:

$$S \;:=\; \{\, (X_P\,,\,(K_X)_{|P} - F_0\,) \mid X \in Dom(R) \;\wedge\; P \in Con(X(1)) - \{\varnothing\}\,\}$$
$$\cup\; \{\, (X_\varnothing\,,\,(K_X)_{|\ll X \gg} \mid X \in Dom(R)\,\}$$

generates from X_P the connected component of $R^\omega(X)$ containing $P \neq \varnothing$, and S generates from $X_\varnothing$ the remaining part of $R^\omega(X)$.
In particular $G \in S^\omega(Z_\varnothing)$.
The grammar S is connected but it is not necessarily reduced.
However by applying Lemma 3.1, we get an equivalent connected and reduced grammar of axiom $Z_\varnothing$. Q.E.D. (Lemma 3.3)

The transformation of Lemma 3.3 is illustrated below.

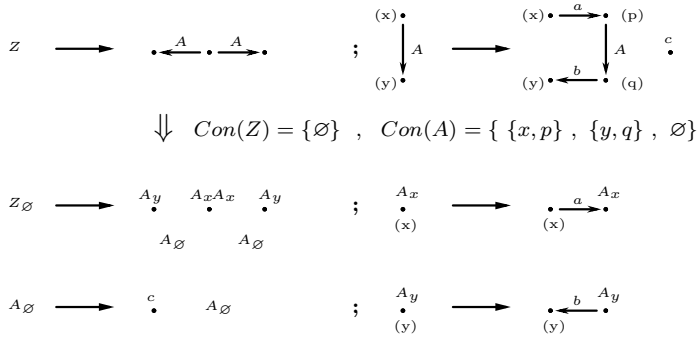
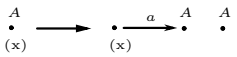


Figure 3.3. From a proper grammar to a connected grammar.

A regular graph can have an infinite number of connected components as shown in the figure below.

Grammar:



Graph generated from its non-terminal:

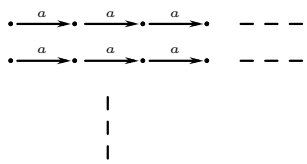


Figure 3.4. A non connected regular graph.

An even simpler example is given by the graph grammar reduced to the unique rule $Z \longrightarrow \{axy\,,\,Z\}$ which generates from the constant Z the infinite repetition of an a-arc.
However these two regular graphs have only a unique connected component up to isomorphism. Let us generalize this property.

Proposition 3.4. *A regular hypergraph has a finite number of non-isomorphic connected components.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph. By Lemma 3.3, G is generated by a connected and reduced grammar R from its axiom Z.
We restrict R to the following grammar:

$$S \ := \ \{ \ (X,H) \in R \mid X \notin F_0 \ \} \, .$$

The grammar S preserves the connectivity:

$S^\omega(K)$ is connected for any connected hypergraph $K \notin N_R \cap F_0$.

Any connected component of G is isomorphic to a hypergraph of the following set:

$$\bigcup \{ \ S^\omega(K) \mid \exists \ H \in Im(R), \ K \ \text{ conn. comp. of } \ H \ - \ (N_R \cap F_0) \ \}$$

which has a finite number of non-isomorphic hypergraphs. Q.E.D. (Prop. 3.4)

Let us normalize the constant rules.
A grammar R is *strongly reduced* if R is a reduced grammar with at most two non-terminal constants (i.e. $|N_R \cap F_0| \leq 2$), and such that

$$(X,H) \in R \ \wedge \ X \notin F_0 \quad \Longrightarrow \quad H \cap F_0 = \varnothing \, .$$

Note that this last condition is already satisfied if R is connected. Let us extend Lemma 3.3.

Proposition 3.5. *Any regular hypergraph can be generated in an effective way by a connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph. By Lemma 3.3, G is generated by a connected and reduced grammar R from its axiom Z.
We extract in R the following constant rules:

$$R_0 \ := \ \{ \ (X,Y) \mid \exists \ H, \ (X,H) \in R \ \wedge \ X \in F_0 \ \wedge \ Y \in F_H \cap N_R \cap F_0 \ \}$$

in order to determine the following subset of 'non-repetitive' constant non-terminals:

$$NRep \ := \ \{ \ A \in N_R \cap F_0 \mid (A,A) \notin R_0^+ \ \} \ - \ \{Z\}.$$

First we restrict R to the rules of its non-repetitives constant non-terminals:

$$I \ := \ \{ \ (X,H) \in R \mid X \in NRep \ \} \ .$$

To each $X \in NRep$, we derive a hypergraph H_X such that

$$X \underset{I}{\Longrightarrow}^* H_X \ \wedge \ F_{H_X} \cap NRep \ = \ \varnothing$$

and we define the following grammar:

$$I' \ := \ \{ \ (X, H_X) \mid X \in NRep \ \} \, .$$

By rewriting according to I', we remove the non-repetitive constant non-terminals in R.
For each $X \in F_0 - NRep$, we associate a hypergraph H'_X such that

$$X \ \ R \circ \underset{I'}{\Longrightarrow} H'_X$$

with $V_{H'_X} \cap V_{H'_Y} = \varnothing$ for every $X \neq Y$ in $F_0 - NRep$.
The following grammar:

$$S \ := \ \{ \ (X,H) \in R \mid X \notin F_0 \ \} \ \cup \ \{ \ (X, H'_X) \mid X \in F_0 - NRep \ \} \, .$$

remains connected, reduced and generates G from its axiom Z.
The set of 'repetitive' constant non-terminals is

$$Rep := \{ A \mid (A,A) \in R_0^+ \} = (N_R \cap F_0) - (NRep \cup \{Z\}) .$$

If $Rep = \varnothing$ then S suits with $N_R \cap F_0 = \{Z\}$.
Otherwise we take a new constant $Y \neq Z$ and the following graphs:

$$K_0 := (H'_Z)_{|-F_0} \quad \text{the image of } Z \text{ in } S \text{ without constants,}$$
$$\text{and} \quad K := \bigcup\{ (H'_X)_{|-F_0} \mid X \in Rep \} .$$

The following grammar:

$$S' := \{ (X,H) \in S \mid X \notin F_0 \} \cup \{ (Z , K_0 \cup \{Y\}) , (Y , K \cup \{Y\}) \}$$

remains connected and S' generates G from Z.
By restriction to the accessible non-terminals from Z using Lemma 3.1, we get an equivalent grammar which is strongly reduced. Q.E.D. (Proposition 3.5)

The transformation of Proposition 3.5 is illustrated below.

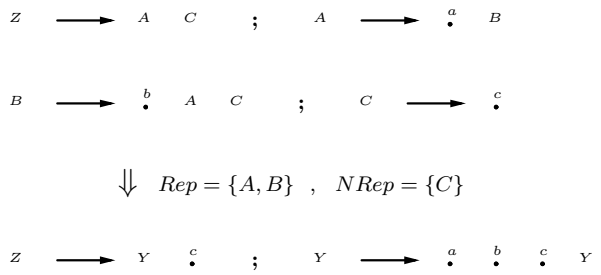


Figure 3.5. From a reduced grammar into a strongly reduced one.

3.2 Discarding the multiplicity

In this section, we construct for any reduced grammar a finite graph of its output dependencies by rewritings from the axiom. This permits to decide whether every derivation from the axiom is only on simple hypergraphs (without multiplicity). Then we present a normal form that allows to get ride of multiplicity. In a first time in Lemma 3.8, we show that any grammar is equivalent to one where right hand sides are hypergraphs and not multi-hypergraphs. In a second time, we show in Proposition 3.10 that any grammar is equivalent to a grammar where each non-terminal hyperarc appearing in a right hand side contains a vertex which is not an input. For a grammar in this normal form, the generated graph can be defined using only hypergraphs and not multi-hypergraphs.

Let R be any reduced grammar.
An *output link* C of R is a multi-hypergraph of at most two hyperarcs which are non-terminals and with a common vertex:

$$|C| \leq 2 \ \wedge \ F_C \subseteq N_R \ \wedge \ (X,Y \in C \implies V_X \cap V_Y \neq \varnothing) ;$$

we denote $[C]_\sim := \{\ D \mid C \ \sim \ D\ \}$ the closure of C by isomorphism.
The *output dependency graph* $Out(R)$ of R is

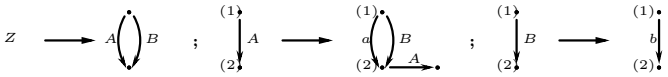
$$Out(R) \ := \ G_{|\{\ s\ \mid\ [Z]_\sim \ \underset{G}{\longrightarrow}^*\ s\ \}}$$

the graph G below and restricted to its vertices accessible from $[Z]_\sim$:

$$G \ := \ \{\ [C]_\sim \longrightarrow [D]_\sim \mid C, D \text{ output links } \wedge\, \exists\ H,\ C \underset{R}{\longrightarrow} H$$
$$\wedge\ D \subseteq H \ \wedge\ (|D| = 1 \ \Rightarrow \ D \text{ conn. comp. of } H - [H])\ \}\,.$$

We give below the output dependency graph of a reduced grammar.

Grammar R:



Output dependency graph Out(R):

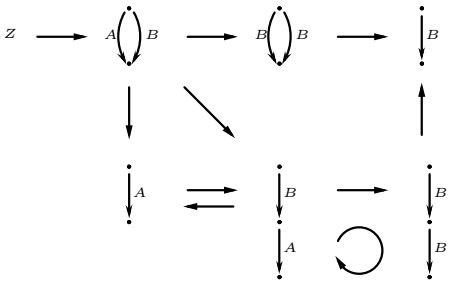


Figure 3.6. Output dependency graph of a grammar.

We say that a grammar R is *without multiplicity* if R is reduced and every vertex of $Out(R)$ is a simple hypergraph. Thus

$$R \text{ is without multiplicity} \quad \Longleftrightarrow \quad (\forall\ H,\ Z \underset{R}{\overset{*}{\longrightarrow}} H \quad \Longrightarrow \quad H \text{ simple}).$$

In particular, any grammar without multiplicity is simple.
We now want to transform any grammar into an equivalent grammar without multiplicity. We start with preliminary normal forms presented in Lemma 3.6 and 3.7.
We say that a grammar R is *growing* if $R = \varnothing$ or R generates an infinite hypergraph from each left hand side, except possibly from its axiom Z:

for all $X \in Dom(R) - \{Z\}$ and $G \in R^\omega(X)$, we have $|G| = \omega$.

Lemma 3.6. *Any regular hypergraph can be generated in an effective way by a growing, connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph.
By Proposition 3.5, G is generated by a connected and strongly reduced grammar R from its axiom Z.

We define two binary relations R_0 and R_1 on the non-terminal set N_R as follows:

$$R_0 \;:=\; \{\ (X(1)\,,Y(1)) \mid \exists\, H,\ (X,H) \in R \ \wedge\ Y \in H \ \wedge\ Y(1) \in N_R\ \}$$

$$R_1 \;:=\; \{\ (X(1)\,,Y(1)) \mid \exists\, H,\ (X,H) \in R \ \wedge\ Y \in H \ \wedge\ Y(1) \in N_R \ \wedge\ V_Y - V_X \neq \varnothing\ \}$$

Then the set

$$E \;:=\; \{\ A \mid \exists\, B,\ (A,B) \in R_0^* \ \wedge\ (B,B) \in R_1^+\ \}$$

is the set of non-terminals $X(1)$ with $X \in Dom(R)$ such that the graphs of $R^\omega(X)$ are infinite. We begin with the grammar:

$$I_0 \;:=\; \{\ (X,\varnothing) \mid X \in Dom(R) \ \wedge\ X(1) \in N_R - E\ \}$$

and having constructed a grammar I_n for $n \geq 0$, we define a deterministic grammar I_{n+1} with $Dom(I_{n+1}) = Dom(I_0)$ and

$$I_{n+1} \;\subseteq\; \{\ (X,H) \mid X\ R \circ \underset{I_n}{\Longrightarrow} H\ \}\,.$$

Note that the right hand sides of the grammars I_n do not contain any non-terminal hyperarc.

We stop with the grammar $I = I_m$ for $m = min\{\ n \mid I_n = I_{n+1}\ \}$.

Thus I is a grammar with $Dom(I) = \{\ X \in Dom(R) \mid X(1) \in N_R - E\ \}$ and for every $(X,H) \in I$, H is finite and $H \in R^\omega(X)$.

From I, we construct a deterministic grammar S such that

$$S \;\subseteq\; \{\ (X,H) \mid X\ R \circ \underset{I}{\Longrightarrow} H \ \wedge\ X(1) \in E\ \}.$$

This grammar S is growing, connected and by restriction to the accessible non-terminals from Z, it is strongly reduced. Q.E.D. (Lemma 3.6)

We say that a grammar R is *strict* if

$$V_H - V_X \neq \varnothing \quad \text{for any} \quad (X,H) \in R$$

any rule has at least one non-input vertex in its right hand side.

Starting from a growing grammar, it is enough to write every right hand side until the grammar is strict.

Lemma 3.7. *Any regular hypergraph can be generated in an effective way by a strict, connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph.

By Lemma 3.6, G is generated by a growing, connected and strongly reduced grammar R from its axiom Z.

As R is growing, we derive each right hand side of S until we get a non input vertex. We define

$$S_0 \;:=\; \{\ (X,H) \in R \mid V_X \neq V_H\ \}$$

and having defined S_n, we construct a maximal deterministic grammar

$$S_{n+1} \;\subseteq\; S_n \,\cup\, \{\ (X,H) \mid X \in Dom(R) - Dom(S_n) \ \wedge\ X\ R \circ \underset{S_n}{\Longrightarrow} H \ \wedge\ V_X \neq V_H\ \}$$

to get $S := S_m$ for $m = min\{\ n \mid \forall\ (X,H) \in S_n\ ,\ V_X \neq V_H\ \}$.
This grammar S is strict and generates G from its axiom Z.
Furthermore S remains connected and becomes strongly reduced by restriction to the accessible non-terminals from Z. Q.E.D. (Lemma 3.7)

The transformations of Lemma 3.6 and Lemma 3.7 are illustrated below.

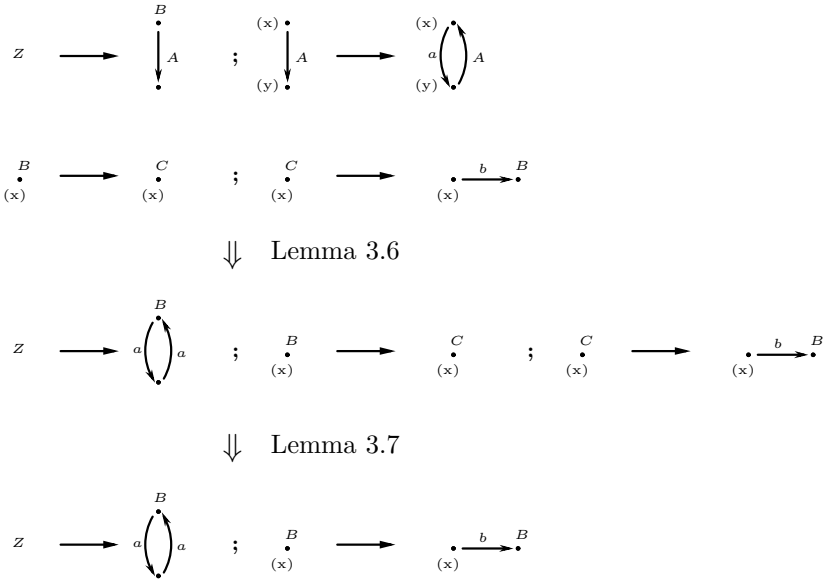


Figure 3.7. Transformation of a grammar into a strict grammar.

To generate a regular (simple) hypergraph, we can avoid multiplicity in the grammar.
Precisely, a *simple grammar* is a grammar where each right hand side is a (simple) hypergraph.

Lemma 3.8. *Any regular hypergraph can be generated in an effective way by a simple, strict, connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph. By Lemma 3.7, G is generated by a strict, connected and strongly reduced grammar R from its axiom Z.
To each non-terminal $A \in N_R - \{Z\}$, we associate its maximal multiplicity:

$$m(A) := max\{\ H(X) \mid H \in Im(R)\ \wedge\ X \in H\ \wedge\ X(1) = A\ \},$$

and we take new non-terminals $A_1, \ldots, A_{m(A)}$ of arity $\varrho(A)$.
This allows us to replace each right hand side $H \in Im(R)$ by the following simple hypergraph:

$$\begin{aligned} H' \quad &:= \quad \{\ X \mid X \in H\ \wedge\ X(1) \in T_R\ \} \\ &\cup \quad \{\ X(1)_i X(2) \ldots X(|X|) \mid X \in H \cap N_R V_H^*\ \wedge\ 1 \leq i \leq H(X)\ \}. \end{aligned}$$

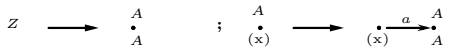
We obtain the following grammar:

$$\begin{aligned} S \quad &:= \quad \{\ (Z, H') \mid (Z, H) \in R\ \} \\ &\cup \quad \{\ (X(1)_i X(2)\ldots X(|X|), H') \mid (X, H) \in R \ \wedge\ 1 \leq i \leq m(X(1))\ \}\,. \end{aligned}$$

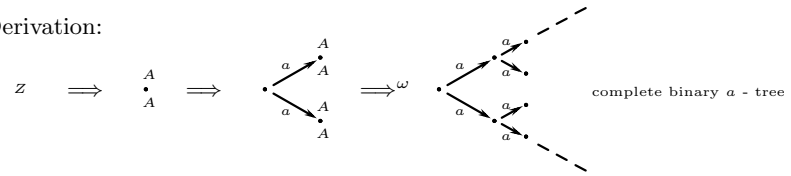
This grammar S is simple, strict, connected, strongly reduced and generates G from its axiom Z. Q.E.D. (Lemma 3.8)

The transformation of Lemma 3.8 is illustrated below.

Grammar:



Derivation:



Equivalent simple grammar:

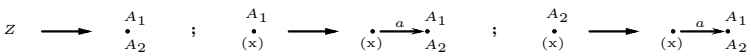
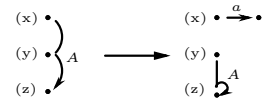


Figure 3.8. Transformation of a grammar into a simple grammar.

To generate a regular hypergraph, we also want to reduce the rewriting steps to (simple) hypergraphs. This is not possible in general as shown in Figure 2.6 and in the figure below.

Simple grammar:



Derivation:

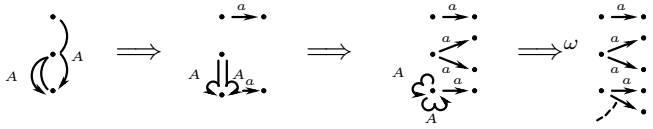


Figure 3.9. Multiplicity by parallel rewritings.

However any regular hypergraph can be generated by a simple hypergraph grammar whose rewriting steps are restricted to simple hypergraphs.
A grammar R is *non-terminal outside* if for any rule $X \longrightarrow H$, any non-terminal hyperarc $Y \in H$ with $Y(1) \in N_R$ has a vertex which is not an

input: $V_Y - V_X \neq \varnothing$.
The grammar of Figure 3.4 is non-terminal outside and the grammar of Figure 3.9 is not.
With the strongly reduced property, we have removed the multiplicity by parallel rewritings for constants. The non-terminal outside property removes the multiplicity by parallel rewritings for non-constant non-terminals.

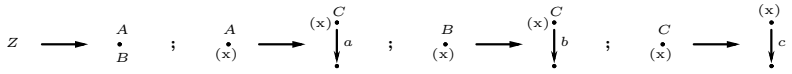
Lemma 3.9. *Any non-terminal outside, simple and strongly reduced grammar is without multiplicity.*

Proof. Let R be any non-terminal outside, simple and strongly reduced grammar.
By induction, we verify that the rewriting $\underset{R}{\longrightarrow}$ preserves the property $P(H)$ of a hypergraph H to be simple and with at most one non-terminal constant:

$$P(H) \ \wedge \ H \underset{R}{\longrightarrow} K \quad \Longrightarrow \quad P(K).$$

Let X be the left hand side of the applied rule. If X is a constant then the implication is due to R being simple and strongly reduced.
If X is not a constant then the implication is due to R being simple, strongly reduced and non-terminal outside. Q.E.D. (Lemma 3.9)

We give below another simpler grammar which is not non-terminal outside and for which the generated graph is obtained by parallel rewritings with multiplicity.



Derivation:

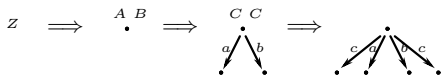


Figure 3.10. A graph grammar which is not non-terminal outside.

We can transform any grammar into an equivalent simple non-terminal outside grammar.

Proposition 3.10. *Any regular hypergraph can be generated in an effective way by a non-terminal outside, simple, connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph.
By Lemma 3.8, G is generated by a simple, strict, connected and strongly reduced grammar R from its axiom Z.

Recall that a connected grammar is proper.
We transform R by incrementing the arity of non constant non-terminal hyperarcs.
For each non-terminal $A \in N_R - F_0$, we take a new symbol A' of arity $\varrho(A') = \varrho(A) + 1$.
For each $(X, H) \in R$ with $X \notin F_0$, there exists a vertex $x_X \in V_H - V_X$ because R is strict, and we define the hyperarc:

$$X' := X(1)'X(2)\ldots X(|X|)x_X .$$

For each $H \in Im(R)$ and for each $Y \in H$, we define the following hyperarc:

$$Y' := \begin{cases} Y & \text{if } Y(1) \notin N_R - F_0 \\ Y(1)'Y(2)\ldots Y(|Y|)y_Y & \text{if } Y(1) \in N_R - F_0 \,; \end{cases}$$

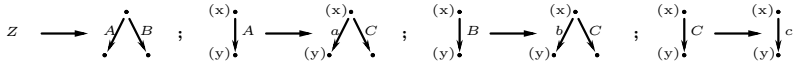
where y_Y is a new vertex (not in R with $y_Y \neq y_Z$ for $Y \neq Z$).
By union, we extend to $H' := \{\, Y' \mid Y \in H \,\}$.
It remains to take

$$\begin{aligned} S \quad &:= \quad \{\, (X, H') \in R \mid (X, H) \in R \ \wedge\ X \in F_0 \,\} \\ &\cup \quad \{\, (X', H') \mid (X, H) \in R \ \wedge\ X \notin F_0 \,\} . \end{aligned}$$

The grammar S remains simple, connected and strongly reduced of axiom Z. And S is non-terminal outside and generates G. Q.E.D. (Proposition 3.10)

The transformation of Proposition 3.10 is illustrated below.

Grammar:



Derivation:

Figure 3.11. Transformation of the grammar of Figure 3.10.

3.3 Separating the inputs with the outputs

We want to extend Proposition 3.10 by separating as much as possible in every right hand side of the grammar input and output vertices. However we can observe that if a vertex of a left hand side X is of infinite degree in $R^\omega(X)$ then it must be also an output. We will show that a grammar can be transformed into an equivalent one such that the non-output vertices of every left hand side X are the inputs of finite degree in $R^\omega(X)$.
A grammar R is *terminal outside* if for any rule $X \longrightarrow H$, any terminal hyperarc $Y \in H$ with $Y(1) \in T_R$ has a vertex which is not an input: $V_Y - V_X \neq \varnothing$.

An *outside grammar* is a terminal outside and non-terminal outside grammar.

Lemma 3.11. *Any regular hypergraph can be generated in an effective way by an outside, simple, connected and strongly reduced grammar.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph.
By Lemma 3.1, G is generated by a reduced grammar R from its axiom Z.
By least fixed point, we define the grammar I such that $Dom(I) = Dom(R)$ and

$$I \;=\; \{\, (X\,,\, H \cap T_R V_X^*) \mid X \; R \circ \underset{I}{\Longrightarrow} H \,\}\,.$$

We define the following grammar:

$$J \;:=\; \{\, (X\,,\, H \cup \{X\}) \mid (X,H) \in I \;\wedge\; X \neq Z \,\}$$

and the following grammar:

$$S \;:=\; \{\, (Z,H) \mid Z \; R \circ \underset{J}{\Longrightarrow} H \,\} \;\cup\; \{\, (X, H_{|- T_R V_X^*}) \mid X \; R \circ \underset{J}{\Longrightarrow} H \,\}\,.$$

For any $X \in Dom(R) - \{Z\}$,

$$S^\omega(X) \;=\; \{\, K - T_R V_X^* \mid K \in R^\omega(X) \,\}$$

hence $S^\omega(Z) = R^\omega(Z)$.
Furthermore S is terminal outside but not necessary reduced (due to condition (iii)).
By applying the previous constructions, the obtained grammar remains terminal outside and becomes non-terminal outside, simple, connected and strongly reduced. Q.E.D. (Lemma 3.11)

Let us apply the construction of the proof of Lemma 3.11 to the grammar of Figure 2.4 completed with the rule $Z \longrightarrow Axy$.

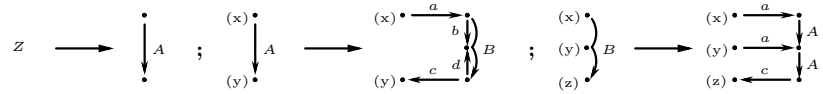
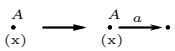


Figure 3.12. From the grammar of Figure 2.4 to a terminal outside one.

In the last figure of Section 2 and in Figure 3.9, we have regular graphs with vertices of infinite degree. We give below another regular graph of infinite degree.

Grammar:



Graph generated from its unique non-terminal:

Figure 3.13. A regular graph of infinite degree.

We will see that there is no regular hypergraph of finite degree which is not of bounded degree. To compute the vertex degrees of a hypergraph, we separate in the right hand sides of a grammar the outputs from the inputs of finite degree.
A *degree-outside* grammar R is a grammar such that the vertices of any right hand side which are inputs and outputs are the input vertices of infinite degree in the generated graph:
$\forall\,(X,H)\in R,\ \forall\,Y\in H\cap N_R\,V_H^*\,,\ V_X\cap V_Y\ \subseteq\ \{\,s\in V_X\mid d_{R^\omega(X)}(s)=\omega\,\}\,.$
The grammar of Figure 3.13 is degree-outside but the grammar below is not: x is both an input and an output but is of finite degree in the generated graph.

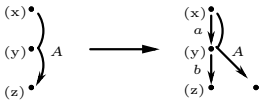


Figure 3.14. A grammar which is not degree-outside.

A degree-outside and reduced grammar generating a hypergraph of finite degree is called an *input-separated grammar*: for each right hand side, any input is not an output.
A grammar which is outside and degree-outside is a *complete outside grammar*.
Any regular hypergraph can be generated by a complete outside grammar.

Theorem 3.12. *Any regular hypergraph can be generated in an effective way by a complete outside, simple, connected and strongly reduced grammar.*

Proof. Let $G\neq\varnothing$ be a regular hypergraph.
By Lemma 3.11, G is generated by an outside, simple, connected and strongly reduced grammar R from its axiom Z.
For any hypergraph H and any $P\subseteq V_H$, we denote
$[H,P]\ :=\ |\{\,(Y,i)\mid Y\in H\ \wedge\ Y(1)\in N_R\ \wedge\ 2\leq i\leq|Y|\ \wedge\ Y(i)\in P\,\}|$
the number of non-terminal links in H on vertices in P.
To get from R a degree-outside grammar, we derive each right hand side until we cannot separate outputs from inputs. We begin with the initial grammar:
$$S_0\ :=\ R$$
and having constructed a grammar S_n with $n\geq 0$, we associate to each rule $(X,H)\in S_n$ a hypergraph K_X such that
$$H\underset{S_n}{\longrightarrow}K_X\quad\wedge\quad[K_X,V_X]<[H,V_X]$$
if such a hypergraph exists, otherwise $K_X=H$; and we define the grammar

$$S_{n+1} \;:=\; \{\ (X, K_X) \mid X \in Dom(R)\ \}\,.$$

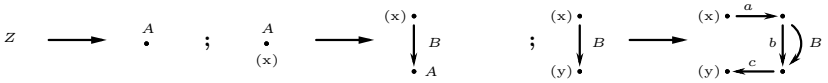
We stop with the grammar:

$$S \;:=\; S_m \quad \text{for} \quad m \;=\; min\{\ n \mid S_n = S_{n+1}\ \}\,.$$

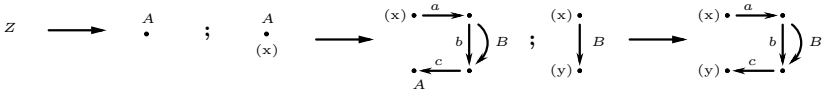
This grammar S is complete outside, simple, connected and generates G from Z. And S becomes strongly reduced by restriction to the accessible non-terminals from Z. Q.E.D. (Theorem 3.12)

Note that the transformation of Theorem 3.12 applied directly to the grammar of Figure 3.14 which is not terminal outside, and completed with the rule $Z \longrightarrow \{A123\}$, leaves the grammar unchanged. Let us apply the transformation of Theorem 3.12 to a suitable grammar.

Outside, simple, connected and strongly reduced grammar:



Equivalent complete outside grammar:



Generated graph:

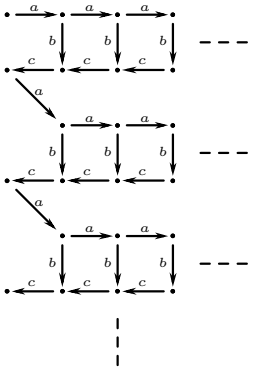
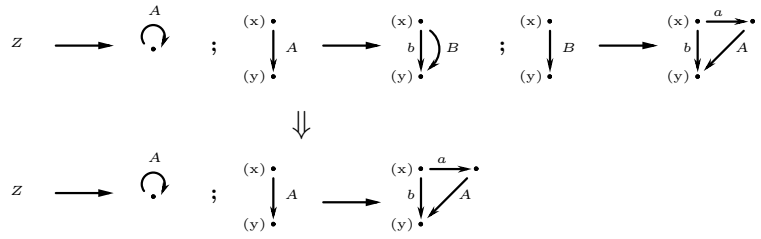


Figure 3.15. Transformation of Theorem 3.12.

The transformation of Theorem 3.12 is also illustrated below.



generating the graph:

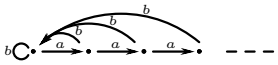


Figure 3.16. Transformation of a grammar into a degree-outside one.

The regular graph of Figure 3.16 has only two possible vertex degrees: 3 and ω. Let us generalize this property.

Proposition 3.13. a) *Any regular hypergraph has a finite number of vertex degrees, hence is either of infinite degree or of bounded degree.*
b) *The class of regular hypergraphs is closed under colouring of vertices whose degree belongs to a given subset of* $\mathbb{N} \cup \{\omega\}$.

Proof. Let $G \neq \varnothing$ be a regular hypergraph. By Theorem 3.12, G is generated by a complete outside, simple, connected and strongly reduced grammar R from its axiom Z.
We can assume that the non-input vertices of the right hand sides are distinct:
$(V_H - V_X) \cap (V_K - V_Y) = \varnothing \quad \forall\ (X,H), (Y,K) \in R$ with $X \neq Y$,
and we denote by E the finite set of non-input vertices in R:
$$E := \bigcup\{\ V_H - V_X \mid (X,H) \in R\ \}.$$
Let us prove Property (a).
For each rule $(X,H) \in R$, we take a hypergraph K such that $H \underset{R}{\Longrightarrow} K$ and for every vertex $s \in V_H - V_X$, we define

$$d(s) := \begin{cases} \omega & \text{if } \exists\ Y \in K,\ Y(1) \in N_R \ \wedge\ s \in V_Y \\ d_{[K]}(s) & \text{otherwise.} \end{cases}$$

The vertex degrees of G form the set $\{\ d(s) \mid s \in E\ \}$ which is finite and computable.
Let us prove Property (b). Let $P \subseteq \mathbb{N} \cup \{\omega\}$ and $\#$ a colour.
We want to construct a grammar generating
$$G_P := G \cup \{\ \#\, s \mid s \in V_G \ \wedge\ d(s) \in P\ \}.$$
To each rule $(X,H) \in R$, we associate the hypergraph:
$$H' := H \cup \{\ \#\, s \mid s \in V_H - V_X \ \wedge\ d_{R^\omega(H)}(s) \in P\ \}.$$

So the grammar $\{ (X, H') \mid (X, H) \in R \}$ generates G_P from Z.
Q.E.D. (Proposition 3.13)

3.4 Separating the outputs

This last normalization subsection permits to get grammars separating for each right hand side the vertices of the non-terminals.
A grammar R is *output-separated* if for any right hand side, distinct non-terminal hyperarcs have no common vertex and any non-terminal hyperarc has distinct vertices: for any $H \in Im(R)$ and any $X, Y \in H \cap N_R V_H^*$,

$$|V_X| = \varrho(X(1)) \quad \wedge \quad (X \neq Y \;\Rightarrow\; V_X \cap V_Y = \varnothing).$$

Note that any output-separated grammar is without multiplicity.
Theorem 3.12 cannot be extended to get grammars which are also output-separated.

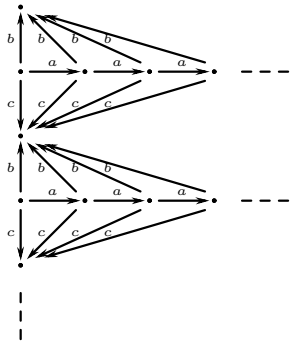


Figure 3.17. Regular graph not given by an output-separated grammar.

However we give a general sufficient condition on any reduced grammar R that allows to transform R into an equivalent output-separated grammar.
To any hypergraph H labelled in $N_R \cup T_R$, we denote

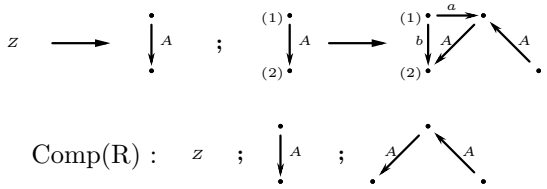
$$Comp(H) := \{ [C]_\sim \mid C \text{ connected component of } H_{|-T_R V_H^*} \}$$

the family of the connected components (up to isomorphism) of the set of non-terminal hyperarcs of H, and

$$Comp(R) := \bigcup \{ Comp(H) \mid Z \underset{R}{\Longrightarrow}^* H \}.$$

We say that R is *output-separable* if $Comp(R)$ is finite.
This notion is illustrated below.

Output-separable grammar R:



Non output-separable grammar:

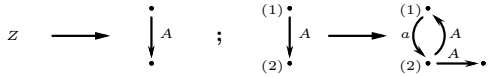


Figure 3.18. Output separation for grammars.

Any input-separated grammar R (reduced and degree-outside grammar with $\mathrm{Gen}(R)$ of finite degree) is output-separable:

$$Comp(R) \quad = \quad \{\,\{Z\}\,\} \cup \{\,[C]_\sim \mid \exists\, H \in Im(R),\ C \text{ connected component of } H_{|-T_R V_H^*}\,\}.$$

Any graph generated by an output-separable grammar can be generated by an output-separated grammar.

Lemma 3.14. *Any output-separable grammar can be transformed into an equivalent output-separated grammar.*

Proof. Let R be any output-separable grammar: R is reduced and $Comp(R)$ is finite. Denoting m the cardinality of $Comp(R)$, we take hypergraphs $H_1, \ldots, H_m$ such that

$$\{[H_1]_\sim, \ldots, [H_m]_\sim\} \quad = \quad Comp(R).$$

The axiom Z of R satisfies $[Z]_\sim = \{Z\}$ hence $Z \in \{H_1, \ldots, H_m\}$.
For each $1 \leq i \leq m$, we take a new symbol A_i of arity $\varrho(A_i) = |V_{H_i}|$, we denote

$$\{s_{i,1}, \ldots, s_{i,\varrho(A_i)}\} \ = \ V_{H_i}$$

we take a hypergraph K_i such that $H_i \underset{R}{\Longrightarrow} K_i$ and let

$$C_{i,1}, \ldots, C_{i,n_i} \text{ be the connected components of } (K_i)_{|-T_R V_{K_i}^*}.$$

By definition of $Comp(R)$ and for every $1 \leq i \leq m$ and $1 \leq j \leq n_i$, there is a unique $1 \leq i_j \leq m$ such that $C_{i,j}$ is isomorphic to H_{i_j}, and we take an isomorphism $h_{i,j}$ from H_{i_j} to $C_{i,j}$: $H_{i_j} \overset{h_{i,j}}{\sim} C_{i,j}$.
We define the grammar S having for each $1 \leq i \leq m$, the following rule:
$A_i s_{i,1} \ldots s_{i,\varrho(A_i)} \longrightarrow [K_i] \cup \{\, A_{i_j} h_{i,j}(s_{i_j,1}) \ldots h_{i,j}(s_{i_j,\varrho(A_{i_j})}) \mid 1 \leq j \leq n_i \,\}$.
So S is output-separated and $S^\omega(Z) = R^\omega(Z)$. Q.E.D. (Lemma 3.14)

The construction of the proof of Lemma 3.14 is illustrated below.

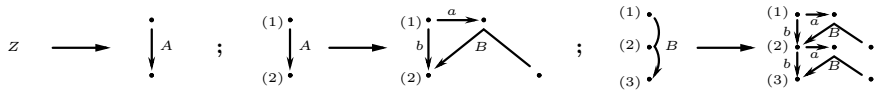


Figure 3.19. Output-separated grammar from the first grammar of Figure 3.18.

Lemma 3.14 permits to extend Theorem 3.12 to any regular graph of finite degree.

Corollary 3.15. *Any regular hypergraph of finite degree can be generated in an effective way by a grammar which is input and output separated, connected and strongly reduced.*

By generation by distance from a vertex of any connected regular graph of finite degree, we will get in next section a grammar normal form stronger than in Corollary 3.15 (cf. Theorem 4.6). The condition of a grammar to be output-separable is effective.

Lemma 3.16. *We can decide whether a reduced grammar is output-separable.*

The proof of Lemma 3.16 is left as a simple exercise on grammars.

Henceforth and considering Proposition 3.10, we assume that any grammar is reduced, proper and without multiplicity.

3.5 Canonical regular graphs

A grammar R generates from a hypergraph K a family $R^\omega(K)$ of isomorphic hypergraphs. We present here a canonical way to extract a representant $\mathrm{Gen}(R,K)$ in this family. A vertex s of $\mathrm{Gen}(R,K)$ is the word of the path of the non-terminals plus the non-input vertex which are used to get s by rewritings.
Up to a label renaming with adding rules, we assume that K and each right hand side of R has no two non-terminal hyperarcs with the same label: for every $H \in \{K\} \cup Im(R)$,

$$Y, Y' \in H \ \wedge\ Y \neq Y' \ \wedge\ Y(1), Y'(1) \in N_R \quad \Longrightarrow \quad Y(1) \neq Y'(1).$$

We denote by V_R the vertex set of K plus the set of non input vertices of the right hand sides of R:

$$V_R \ := \ V_K \ \cup \ \bigcup\{\ V_H - V_X \mid (X,H) \in R\ \}\,.$$

To each word $u \in N_R^*$, we associate for each non-terminal $A \in N_R$ a new symbol A_u of arity $\varrho(A)$, and for each hyperarc X, we define

$$X_u := \begin{cases} X & \text{if } X(1) \notin N_R \\ X(1)_u X(2) \ldots X(|X|) & \text{if } X(1) \in N_R\,, \end{cases}$$

that we extend by union to any hypergraph H: $H_u := \{\ X_u \mid X \in H\ \}$.

To each rule $(X,H) \in R$ and hyperarc Y with $Y(1) = X(1)_u$, $u \in N_R^*$ and $V_Y \subset N_R^* V_R$, we associate the finite hypergraph:

$$\widehat{Y} \;:=\; \big(h(H)\big)_{uX(1)}$$

where h is the function defined for every vertex $r \in V_H$ by

$$h(r) := \begin{cases} Y(i) & \text{if } \; r = X(i) \;\; \text{for some } \; 2 \le i \le |X| \\ u\,X(1)r & \text{otherwise.} \end{cases}$$

Beginning with the hypergraph $H_0 = K_\varepsilon$ and having defined H_n for $n \ge 0$, we construct

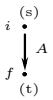
$$H_{n+1} \;:=\; [H_n] \cup \{\, \widehat{Y} \mid Y \in H_n \;\wedge\; \exists\, u \in N_R^*,\; Y(1) \in (N_R)_u \,\}$$

in order to define the following terminal hypergraph:

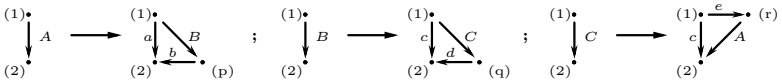
$$\mathrm{Gen}(R,K) \;:=\; \textstyle\bigcup_{n\ge 0}[H_n]\,.$$

Such a hypergraph is generated by R from K: $\mathrm{Gen}(R,K) \in R^\omega(K)$.
We illustrate below the previous construction.

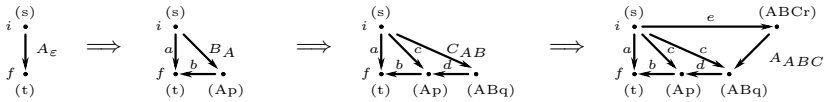
Axiom K :



Grammar R :



Graphs H_0 , ... , H_3 :



Canonical graph $\mathrm{Gen}(R,K)$ represented by vertices of increasing length:

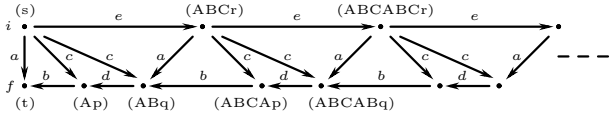


Figure 3.20. Canonical graph generated by a grammar.

The vertex set of $\mathrm{Gen}(R,K)$ is regular because

$$V_{\mathrm{Gen}(R,K)} \;=\; V_K \;\cup\; \bigcup\{\, L_A \mid A \in F_K \cap N_R \,\}$$

where the family of languages L_A for all $A \in N_R$ is the least fixed point of the following system: for each $(X,H) \in N_R$,

$$L_{X(1)} \;=\; X(1).\big((V_H - V_X) \;\cup\; \bigcup\{\, L_A \mid A \in F_H \cap N_R \,\}\big)\,.$$

For the grammar R of Example 3.20, we have

$L_A = A(\{p\} \cup L_B)$; $L_B = B(\{q\} \cup L_C)$; $L_C = C(\{r\} \cup L_A)$
hence $V_{\mathrm{Gen}(R)} = \{s,t\} \cup L_A = \{s\,,\,t\} \cup (ABC)^*\{Ap\,,\,ABq\,,\,ABCr\}$.
For any non-empty finite $\varnothing \neq E \subseteq V_{\mathrm{Gen}(R,K)}$, we define the *least approximant* $\mathrm{Gen}_E(R,K)$ of $\mathrm{Gen}(R,K)$ whose vertex set contains E, which is the hypergraph obtained from K by a minimal number of rewritings to generate all vertices in E. Precisely we begin with

$$H_0 = K_\varepsilon$$

and having defined H_n for $n \geq 0$, either we can choose $Y \in H_n$ with $Y(1) \in (N_R)_u$ for some $u \in (N_R)^*$ such that $E \cap u(N_R)^+V_R \neq \varnothing$, and we take

$$H_{n+1} = (H_n - \{Y\}) \cup \widehat{Y}$$

or if such a Y does not exist, we stop with $\mathrm{Gen}_E(R,K) = H_n$.
Taking the grammar R of Figure 3.20, we give below $\mathrm{Gen}_E(R,C12)$ the least approximant of $\mathrm{Gen}(R,C12)$ containing $E = \{CAp\}$.

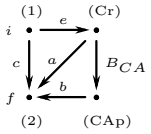


Figure 3.21. $\mathrm{Gen}_{CAp}(R,C12)$ for the grammar R of Figure 3.20.

Note that the hypergraphs H_n given to define $\mathrm{Gen}(R,K) = \bigcup_{n\geq 0}[H_n]$ are approximants:

$$H_n = \mathrm{Gen}_{E_n}(R,K) \quad \text{for} \quad E_n = \{\, v \in V_{\mathrm{Gen}(R,K)} \mid |v| \leq n+1 \,\}\,.$$

The *canonical graph* of a reduced grammar R of axiom Z is

$$\mathrm{Gen}(R) := \mathrm{Gen}(R,Z).$$

4 Generation by distance

In the previous section, we have considered transformations of grammars into equivalent normalized grammars. We now investigate transformations to get grammars generating hypergraphs by vertices of increasing distance from a given colour, either by accessibility or by non-oriented accessibility.

4.1 Regularity by restriction

The regularity of a graph is preserved by restriction to the vertices having a given colour.

Proposition 4.1. *The class of regular hypergraphs is closed under restriction to the vertices having a colour in a given set.*

Proof. Let $G \neq \varnothing$ be a regular hypergraph. By Theorem 3.12, G is generated by an outside, simple, connected and strongly reduced grammar R from its axiom Z.
Let P be a colour set. We want to construct a grammar generating

$$G_P \ := \ G_{|\{\, s \in G \,|\, \exists\, c \in P,\ c\, s \in G \,\}}\,.$$

We can restrict P to a unique colour $\#$, otherwise we take a new colour d to colour all the vertices of G having a colour in P, then we do the restriction of G to the vertices coloured by d and we remove this colour.
To each $A \in N_R$ and each $I \subseteq [\varrho(A)]$, we associate a new non-terminal A_I of arity $\varrho(A)$.
For each rule $(X, H) \in R$ and $I \subseteq [\varrho(X(1))]$, we define the hypergraph:

$$\begin{aligned} H_I \ := \ & \{\, Y \in H \mid Y(1) \in T_R \ \wedge\ \forall\, 1 < i \leq |Y|, \\ & \qquad \#\, Y(i) \in H \ \vee\ Y(i) \in [X, I] \,\} \\ \cup \ & \{\, B_J Y \mid B \in N_R \ \wedge\ BY \in H \ \wedge \\ & \quad J \ = \ \{\, j \mid 1 \leq j \leq |Y| \ \wedge\ (\#\, Y(j) \in H \ \vee\ Y(j) \in [X, I]) \,\} \end{aligned}$$

$$\text{with} \qquad [X, I] \ := \ \{\, X(i+1) \mid i \in I \,\}\,.$$

Thus the grammar

$$\{\, (A_I X\,,\ H_I) \mid A \in N_R \ \wedge\ (AX, H) \in R \ \ \wedge\ I \subseteq [\varrho(A)] \,\}$$

generates $G_\#$ from $Z_\varnothing$. Q.E.D. (Proposition 4.1)

By Propositions 3.13 and 4.1, the regular graphs are closed by restriction to a given set of degrees.
The construction of the proof of Proposition 4.1 is illustrated below.

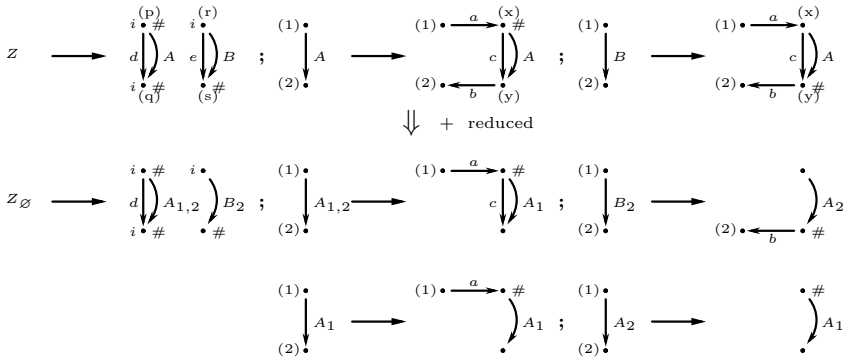


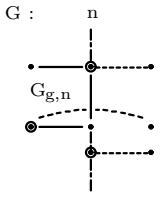
Figure 4.1. Grammar transformation for the restriction to colour $\#$.

4.2 Regularity by graduation

A *graduation* g of a hypergraph G is a mapping from V_G into $\mathbb{N}$ such that only finitely many vertices have the same value by g *i.e.* g^{-1} is locally finite: $g^{-1}(n) = \{ s \in V_G \mid g(s) = n \}$ is finite for every $n \geq 0$. We will define the regularity of a hypergraph by vertices of increasing graduation. Precisely for every $n \geq 0$, we denote

$$\begin{aligned} G_{g,n} &:= G_{|\{ s \mid g(s) \leq n \}} \\ &= \{ X \in G \mid g(X(2)) \leq n \ \wedge \ \ldots \ \wedge \ g(X(|X|)) \leq n \} \\ \text{and} \quad \partial_{g,n} G &:= \{ s \mid g(s) \leq n \ \wedge \ \exists\, X \in G,\ s \in V_X \\ &\qquad\qquad \wedge \exists\, t \in V_X\,,\ g(t) > n \} \\ &= \{ s \in V_{G - G_{g,n}} \mid g(s) \leq n \} \end{aligned}$$

the n-th *frontier* of G by g. This is illustrated by the following diagram:



where $G_{g,n}$ contains all edges depicted by a full line and $\partial_{g,n} G$ is the set of circled vertices; note that

$$V_{G_{g,n}} \cap V_{G - G_{g,n}} \ \subseteq \ \partial_{g,n} G\,.$$

We say that a hypergraph G is *regular by* g if there exists a terminal outside grammar R such that for every $n \geq 0$, R generates from its axiom

Z by $n+1$ parallel rewritings the hypergraph $G_{g,n}$ of terminal hyperarcs, plus a set of non-terminal hyperarcs of vertex set $\partial_{g,n}G$ *i.e.*

$$\forall\ n \geq 0\ \exists\ H,\ Z \underset{R}{\Longrightarrow}^{n+1} H\ \wedge\ [H] = G_{g,n}\ \wedge\ V_{H-[H]} = \partial_{g,n}G\,;$$

we also say that R generates G according to g.
Observe that if G is connected and $G_{g,m} \neq \varnothing$, we have for $n \geq 0$,

$$\partial_{g,m+n}G = \varnothing \iff G_{g,m+n} = G.$$

When V_G is a language, then word length may be used as a graduation. For instance, the canonical graph of Figure 3.20 is regular by length.

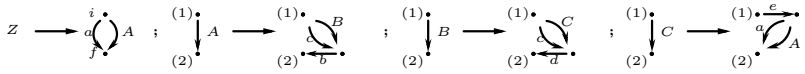


Figure 4.2. Generating the graph of Figure 3.20 by (length -2).

The regularity by length is true for any canonical hypergraph.

Proposition 4.2. *Any canonical hypergraph is regular by length.*

Proof. Let R be a grammar.
Let us construct a grammar generating $\mathrm{Gen}(R)$ by length.
By Lemma 3.11, we get a terminal outside grammar S with the same canonical hypergraph: $\mathrm{Gen}(S) = \mathrm{Gen}(R)$.
As S is outside, S generates $\mathrm{Gen}(S)$ by length minus 2.
By denoting Z the axiom of S and by adding two new constant symbols Z_0, Z_1, we complete S into $S \cup \{(Z_0, Z_1)\,, (Z_1, Z)\}$ which generates $\mathrm{Gen}(S)$ by length (from Z_0)\,. Q.E.D. (Proposition 4.2)

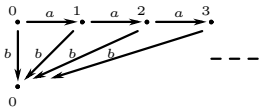
Proposition 4.2 implies that any regular hypergraph is regular by some graduation.

A dual way to express the regularity by graduation is by decomposition: we remove iteratively on the graph the vertices with graduation less than $1, 2, \ldots$. The decomposition allows to avoid the explicit use of grammars. The *decomposition* at level $n \geq 0$ of a hypergraph G by a graduation g is the following hypergraph:

$$G^g_n\ :=\ (G - G_{g,n-1})\ \cup\ \{\ max\{0, g(s) - n\}\, s \mid s \in V_{G-G_{g,n-1}}\ \}$$

obtained from G by removing $G_{g,n-1}$ with $G_{g,-1} = \varnothing$ and by colouring any remaining vertex s by the integer $max\{0, g(s) - n\}$ (assuming that G has no integer colour otherwise we must use a new integer colour: p' for each $p \geq 0$).
In particular $G^g_0\ =\ G \cup \{\ g(s)\, s \mid s \in V_G\ \}$.
We give an example in the next figure.

The graph $G = \{ n \xrightarrow{a} n+1 \mid n \geq 0 \} \cup \{ n \xrightarrow{b} \omega \mid n \geq 0 \}$ with the graduation $g(n) = n$ for $n \geq 0$ and $g(\omega) = 0$ yields the graph G_0^g:



The graphs G_n^g for $n \geq 1$ are all equal to the following graph:

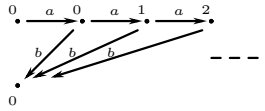


Figure 4.3. Graph decomposition.

We say that a hypergraph G is *finitely decomposable* by a graduation g if the disjoint union

$\sum_{n\geq 0} G_n^g := \{ X(1)(X(2),n)\ldots(X(|X|),n) \mid n \geq 0 \wedge X \in G_n^g \}$

only has a finite number of non isomorphic connected components.

For instance the graph G of Figure 4.3 is finitely decomposable by its graduation g with only two non isomorphic connected components: G_0^g and G_1^g. Another example is the complete binary tree

$$T := \{ u \xrightarrow{a} ua \mid u \in \{a,b\}^* \} \cup \{ u \xrightarrow{b} ub \mid u \in \{a,b\}^* \}$$

which is finitely decomposable by length: $T_0^{|\ |}$ and for every $n \geq 1$, any connected component of $T_n^{|\ |}$ is isomorphic to $T_1^{|\ |}$.

A last example is the semiline $\mathbb{N}$ which is regular but is not finitely decomposable using the graduation associating to $n \geq 0$ the $n+1$-th prime number.

By definition the vertices of G_n^g coloured by 0 are vertices of G coloured by some $i \leq n$:

$$\{ s \mid 0s \in G_n^g \} \subseteq \{ s \in V_G \mid g(s) \leq n \} \quad \text{which is finite.}$$

In particular any connected component C of $\sum_{n\geq 0} G_n^g$ has a finite set $V_{C,0} = \{ s \in V_C \mid 0\,s \in C \}$ of vertices coloured by 0. So any hypergraph G finitely decomposable by g is *bounded connected* by g in the following sense:

$$\exists\, b \geq 0 \ \ \forall\, C \text{ connected component of } \textstyle\sum_{n\geq 0} G_n^g,\ |V_{C,0}| \leq b$$

or equivalently

$$\exists\, b \geq 0 \ \ \forall\, n \geq 0 \ \ \forall\, C \text{ connected component of } G - G_{g,n-1},\ |\{ s \in V_C \mid g(s) \leq n \}| \leq b.$$

It follows that finite decomposition is a less powerful notion than regularity

(by some graduation). The regular graph G of Figure 3.17 has no finite decomposition because it is not bounded connected by any graduation g: the decomposition G^g_n at any level n has a connected component containing all the vertices of infinite (in-)degree.
The finite decomposition of a hypergraph G by a graduation g also imposes that G has finitely many connected components. It is due to the fact that G^g_0 has a finite number of non isomorphic connected components, and no connected component can be infinitely repeated because g is locally finite.

Graph $G \;=\; \{\; (m,n) \xrightarrow{a} (m,n+1) \mid m,n \geq 0 \;\}$

Graduation $g(m,n) \;=\; m+n$

G^g_0 :

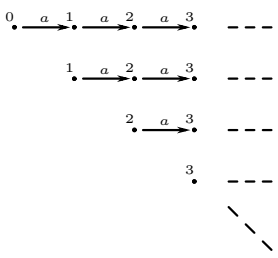


Figure 4.4. Graph regular by graduation, bounded connected, but not finitely decomposable.

Any finite decomposition can be done by a grammar generating what we remove; the converse is true when the hypergraph is bounded connected by the graduation and has only a finite number of connected components.

Proposition 4.3. *Given a graduation g of a hypergraph G,*

$$G \text{ is finitely decomposable by } g \iff \begin{cases} G \text{ is regular by } g \text{ and} \\ G \text{ is bounded connected with finitely many conn. components.} \end{cases}$$

Proof.
$\Longrightarrow$: let G be a hypergraph finitely decomposable by a graduation g.
As already mentioned, G is bounded connected by g and G has only a finite number of connected components.
It remains to show that G is regular by g.
Recall that for any $n \geq 0$,

$$G^g_n \;:=\; (G - G_{g,n-1}) \;\cup\; \{\; max\{0, g(s)-n\}\, s \mid s \in V_{G-G_{g,n-1}} \;\}$$

with $G^g_{-1} = \varnothing$. We define

$$\widehat{G}^g_n \;:=\; (G - G_{g,n}) \;\cup\; \{\; max\{0, g(s)-n\}\, s \mid s \in V_{G-G_{g,n}} \;\}$$

obtained from G^g_n by removing the hyperarcs whose vertices are all coloured

by 0 (only a finite number) and then by removing the isolated vertices coloured by 0.
Let E be a maximal set of non isomorphic conn. comp. of $\{ \widehat{G}_n^g \mid n \geq 0 \}$.
By hypothesis $\{ G_n^g \mid n \geq 0 \}$ has a finite number of non isomorphic connected components, hence E is finite.
For each $C \in E$, we order the set $V_{C,0}$ of vertices of C coloured by 0:

$$\{<C,1>,\ldots,<C,|V_{C,0}|>\} = V_{C,0}\,,$$

and we take a new symbol $[C]$ of arity $|V_{C,0}|$.
Note that for every $n \geq 0$,

$$\begin{aligned} G_{n+1}^g &= \big(\widehat{G}_n^g - \{\, cs \in \widehat{G}_n^g \mid c \in \mathbb{N} \,\}\big) \\ &\cup \{\, max\{0,c-1\}s \mid cs \in \widehat{G}_n^g \ \wedge\ c \in \mathbb{N} \,\}. \end{aligned}$$

To each $C \in E$, we associate the hypergraph

$$C' := (C - \mathbb{N}V_C) \cup \{\, max\{0,c-1\}s \mid cs \in C \ \wedge\ c \in \mathbb{N} \,\}$$

which is isomorphic to a conn. comp. of $\{ G_n^g \mid n \geq 0 \}$, and we define

$$E' := \{\, C' \mid C \in E \,\} \cup \{G_0^g\}\,.$$

For each $C \in E'$, the connected components of

$$C - \{\, X \in C \mid V_X \subseteq V_{C,0} \ \wedge\ X(1) \notin \mathbb{N} \,\}$$

and not reduced to a vertex coloured by 0, are denoted by $C_1,\ldots,C_{n_C}$.
For each $1 \leq i \leq n_C$, there is an isomorphism h_i from C_i to a unique $D_i \in E$.
To each $C \in E'$, we associate the following hypergraph:

$$\begin{aligned} \ll C \gg &:= \{\, X \in C \mid X(1) \notin \mathbb{N} \ \wedge\ V_X \subseteq V_{C,0} \,\} \\ &\cup \{\, [D_i]h_i^{-1}(<D_i,1>)\ldots h_i^{-1}(<D_i,|V_{D_i,0}|>) \mid 1 \leq i \leq n_C \,\}. \end{aligned}$$

Finally the following outside grammar:

$$R := \{\, (Z\,,\, \ll G_0^g \gg) \,\} \cup \{\, ([C]<C,1>\ldots<C,|V_{C,0}|>\,,\, \ll C' \gg) \mid C \in E \,\}$$

generates G from Z and according to g.

$\Longleftarrow$: Assume that G is regular by g, bounded connected by g and has a finite number of connected components.
We want to show that G is finitely decomposable by g.
We can assume without loss of generality that G only has one connected component: G is connected.
There exists an integer b such that for any connected component C of $\sum_{n\geq 0} G_n^g$, $|V_{C,0}| \leq b$.
Consider an outside grammar R generating G by g from its axiom Z.
By the transformation of Lemma 3.3 splitting any hyperarc into connected hyperarcs, we can assume that R is connected.
Consider an infinite parallel derivation $Z \underset{R}{\Longrightarrow} H_0 \ldots H_n \underset{R}{\Longrightarrow} H_{n+1} \underset{R}{\Longrightarrow} \ldots$
For every $n \geq 0$, we have

$$[H_n] = G_{g,n} \quad \text{and} \quad V_{H_n - [H_n]} = \partial_{g,n} G$$

hence

$$\begin{aligned}\{\ s \mid 0\,s \in G^g_n\ \} &= \{\ s \in V_{G-G_{g,n-1}} \mid g(s) \leq n\ \}\\ &\supseteq \{\ s \in V_{G-G_{g,n}} \mid g(s) \leq n\ \}\end{aligned}$$

thus

$V_{H_n-[H_n]} = \partial_{g,n}G = \{\ s \in V_{G-G_{g,n}} \mid g(s) \leq n\ \} \subseteq \{\ s \mid 0\,s \in G^g_n\ \}$.

Then for any $n \geq 0$ and any conn. comp. K of $H_n - [H_n]$, $|V_K| \leq b$.

It follows that $\{\ H_n - [H_n] \mid n \geq 0\ \}$ has a finite number of non isomorphic connected components, and we take a maximal set E of non isomorphic connected components.

Consequently E is finite and the $R^\omega(K)$ for any $K \in E$ are, up to isomorphism, the connected components of $\{\ G - G_{g,n} \mid n \geq 0\ \}$.

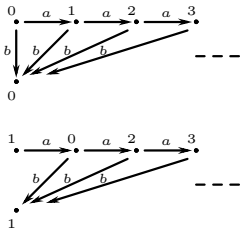
For each $K \in E$, we take $K = K_0 \underset{R}{\Longrightarrow} K_1 \ldots K_n \underset{R}{\Longrightarrow} K_{n+1} \underset{R}{\Longrightarrow} \ldots$ a derivation generating the hypergraph $K' := \bigcup_{n\geq 0}[K_n]$ which we complete by an integer colouring as follows:

$$\overline{K} := K' \cup \{\ min\{\ n \mid s \in V_{K_{n+1}}\ \}\,s \mid s \in V_{K'}\ \}.$$

So $\{\ \overline{K} \mid K \in E\ \}$ are up to isomorphism the connected components of $\{\ G^g_n \mid n > 0\ \}$. Hence G is finitely decomposable by g. Q.E.D. (Prop. 4.3)

The transformation of the necessary condition of Proposition 4.3 is illustrated below.

Gradued graph of finite decomposition



Grammar:

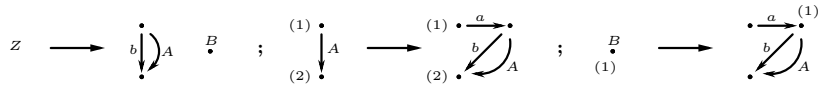


Figure 4.5. Grammar for a finitely decomposable graph.

4.3 Regularity by accessibility

A usual problem in graph theory is the accessibility problem. This problem consists in computing the set of vertices accessible from a given initial set. Here we transform any grammar into another one generating the same

graph plus a colouring of the vertices accessible from (vertices with) a given colour (cf. Proposition 4.4). This grammar transformation is expressed by least fixpoint on the grammar. Finally we give a rooted regular graph of finite degree which cannot be generated by accessibility.

The *accessible vertex* set $Acc(G,i)$ of a hypergraph G from a colour i is the smallest subset of V_G containing the set $V_{G,i}$ of vertices coloured by i and closed under the following accessibility property:

$$\begin{aligned} & fv_1 \ldots v_{\varrho(f)} \in G \ \wedge \ \varrho(f) > 1 \ \wedge \ v_1, \ldots, v_{\varrho(f)-1} \in Acc(G,i) \\ \Longrightarrow \quad & v_{\varrho(f)} \in Acc(G,i)\,. \end{aligned}$$

Equivalently $Acc(G,i)$ is the least solution of the following equation:

$$Acc(G,i) \ = \ V_{G,i} \cup Succ_G(Acc(G,i))$$

for the following *successor* relation:

$$Succ_G(E) \ := \ \{\ v \mid FE^+v \cap G \neq \varnothing\ \} \quad \text{for any} \ \ E \subseteq V_G\,.$$

So a hyperarc realises an 'and' boolean function: we access via a hyperarc $fv_1 \ldots v_{\varrho(f)}$ its last vertex $v_{\varrho(f)}$ if we have accessed all its other vertices $v_1, \ldots, v_{\varrho(f)-1}$.

A hypergraph G is *accessible* from a colour i if $Acc(G,i) = V_G$.

For instance the hypergraph $G = \{fxyz\,,\ gxy\,,\ hx\,,\ c\}$ of Figure 2.1 is accessible from h: $Acc(G,h) = \{x,y,z\}$, but the hypergraph $G = \{i\,x\,,\ j\,y\}$ is not accessible from a unique colour.

We say that a vertex r of a hypergraph G is a *root* if $Acc(G \cup \{ir\}, i) = V_G$ for i a new colour: $i \notin F_G$.

Let us mark by a given colour $\#$ the accessible vertices of any regular hypergraph: we will transform any grammar R generating a hypergraph G into another grammar generating $G \cup \{\ \#v \mid v \in Acc(G,i)\ \}$. This is illustrated in the next figure.

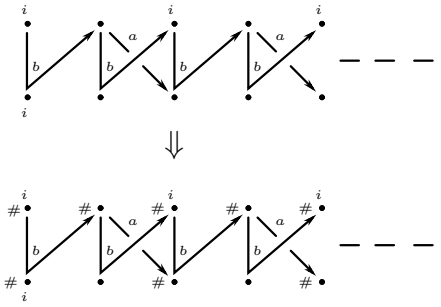


Figure 4.6. Computation of the vertices accessible from i.

The method simply translates the least fixed point defining $Acc(G,i)$ to a least fixed point on the grammar generating G.

Proposition 4.4. *The class of regular hypergraphs is effectively closed under accessible colouring.*

Proof. Let R be a grammar of axiom Z generating a hypergraph G. For colours $\iota, \#$, we want to construct a grammar generating $G \cup \{\ \#v \mid v \in Acc(G,\iota)\ \}$.
Let $1, \ldots, \varrho(R)$ be the vertices of the left hand sides of R: up to renaming, we assume that each left hand side $X \in Dom(R)$ of R is of the form $X = X(1)1 \ldots \varrho(X(1))$.
To each rule $A1 \ldots \varrho(A) \longrightarrow H_A$ in R and each $I \subseteq [\varrho(A)]$, we associate the set $Acc(A,I)$ of vertices in V_{H_A} which are accessible from I and the vertices coloured by ι in a(ny) graph of $R^\omega(H_A)$.
This family of sets $Acc(A,I)$ is the least fixed point of the following recursive system:

$$\begin{aligned} Acc(A,I) \quad &:= \quad I \ \cup\ \{\ v \mid \iota\, v \in H_A\ \} \\ &\cup \quad \{\ v \in V_{H_A} \mid T_R(Acc(A,I))^+ v \cap H_A \neq \varnothing\ \} \\ &\cup \quad \{\ Y(i) \mid \exists\ B \in N_R\,,\ BY \in H_A\ \wedge\ 1 \leq i \leq |Y| \ \wedge \\ &\qquad\qquad i \in Acc(B, \{\, j \mid Y(j) \in Acc(A,I)\}) \ \}\,. \end{aligned}$$

Precisely we take a linear order on the set

$$M \ := \ \{\ (A,I) \mid A \in N_R\ \wedge\ I \subseteq [\varrho(A)]\ \}$$

and we define

$$E \ := \ \{\ \textstyle\prod_{(A,I)\in M} P_{A,I} \mid \forall\ A \in N_R\ \forall\ I \subseteq J \subseteq [\varrho(A)],\ P_{A,I} \ \subseteq\ P_{A,J}\ \}\,.$$

So E is a complete finite set for the inclusion componentwise whose smallest element is $\vec{\varnothing} = (\varnothing, \ldots, \varnothing)$.
Then we define the mapping $f : E \longrightarrow E$ by

$$\begin{aligned} \big(f(\textstyle\prod_{(B,J)\in M} P_{B,J})\big)_{A,I} \quad &:= \quad I \ \cup\ \{\ v \mid \iota\, v \in H_A\ \} \\ &\cup \quad \{\ v \in V_{H_A} \mid T_R\, P^+_{A,I} v \cap H_A \neq \varnothing\ \} \\ &\cup \quad \{\ Y(i) \mid\ \exists\ B \in N_R\,,\ BY \in H_A\ \wedge \\ &\qquad\qquad 1 \leq i \leq |Y|\ \wedge\ i \in P_{B,\{\, j \mid Y(j) \in P_{A,I}\}\ }\ \}\,. \end{aligned}$$

Thus f is monotonous:

$$\begin{aligned} &\big(\forall\ (A,I) \in M,\ P_{A,I}\ \subseteq\ Q_{A,I}\big) \\ \Longrightarrow\quad &f\big(\textstyle\prod_{(A,I)\in M} P_{A,I}\big)\ \subseteq\ f\big(\prod_{(A,I)\in M} Q_{A,I}\big)\,. \end{aligned}$$

As E is finite, f is continuous and by the Knaster-Tarski theorem:

$$\textstyle\bigcup_{n\geq 0} f^n(\vec{\varnothing}) \ \text{ is the least fixed point of } f.$$

So we define for every $(A,I) \in M$,

$$Acc(A,I) \ := \ \Big(\textstyle\bigcup_{n\geq 0} f^n(\vec{\varnothing})\Big)_{A,I}\,.$$

To each (A,I), we associate a new non-terminal A_I of arity $\varrho(A)$, and we define the following grammar:

$$S \ := \ \{\ (A_I 1 \ldots \varrho(A)\,,\, H_{A,I}) \mid A \in N_R\ \wedge\ I \subseteq [\varrho(A)]\ \}$$

where

$$H_{A,I} \quad := \quad (H_A \cap T_R V_{H_A}^*) \ \cup \ \{\ \#v \mid v \in Acc(A,I) - [\varrho(A)]\ \} \ \cup$$
$$\{\ B_{\{\,j \mid Y(j) \in Acc(A,I)\,\}} Y \mid BY \in H_A \ \wedge \ B \in N_R\ \}\,.$$

with a restriction to the rules whose non-terminals are accessible from $Z_\varnothing$. Thus S generates from $Z_\varnothing$ the hypergraph $G \cup \{\ \#v \mid v \in Acc(G,\iota)\ \}$.

Q.E.D. (Proposition 4.4)

The construction in the proof of Proposition 4.4 is illustrated in the figure below.

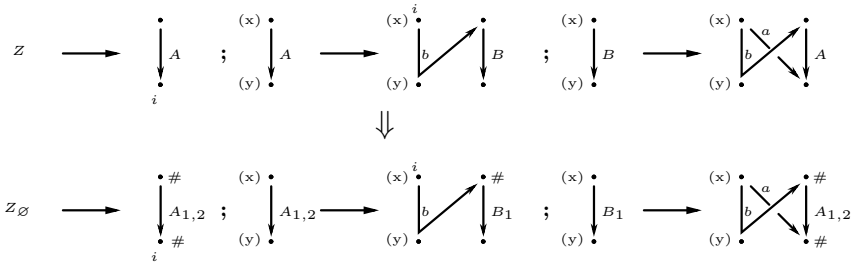


Figure 4.7. Colouring from i for the grammar of Figure 4.6.

The colouring by accessibility of a hypergraph G is a particular case of *regular colouring* by a finite hypergraph H whose vertices are colours *i.e.* $V_H \subset F_1$, and is the hypergraph defined as the least fixed point of the equation:

$$G \otimes H \ := \ G \ \cup \ \{\ c_{\varrho(f)}\, v_{\varrho(f)} \mid \exists\ f v_1 \ldots v_{\varrho(f)} \in G \ \ \exists\ f c_1 \ldots c_{\varrho(f)} \in H,$$
$$c_1 v_1\,,\ \ldots\,,\ c_{\varrho(f)-1} v_{\varrho(f)-1} \in G \otimes H\ \}.$$

In particular

$$G \cup \{\ \#v \mid v \in Acc(G,i)\ \}$$
$$= \quad G \otimes \big(\{i\,\#\} \ \cup \ \{\ f\# \ldots \# \mid f \in F_G \ \wedge \ \varrho(f) > 1\ \}\big).$$

Let us extend Proposition 4.4 to any regular colouring.

Proposition 4.5. *The class of regular hypergraphs is effectively closed under regular colouring.*

Proof. We adapt the proof of Proposition 4.4. Let H be a finite hypergraph with $V_H \subset F_1$.

Let R be a grammar of axiom Z generating a hypergraph G.

We assume that the rule associated to any $A \in N_R$ is of the form:

$$A1 \ldots \varrho(A) \longrightarrow H_A\,.$$

To each $A \in N_R$ and $I \subseteq V_H\,[\varrho(A)]$, we associate the terminal hypergraph $Acc(A,I)$ such that the family of these hypergraphs is the least fixed point of the following recursive system:

$$\begin{array}{rcl} Acc(A,I) & := & I \ \cup \ [H_A] \ \cup \ \big(Acc(A,I) \otimes H\big) \\ & \cup & \{\ c\,Y(i) \mid \exists\ B \in N_R\,,\ BY \in H_A \ \wedge\ 1 \leq i \leq |Y| \ \wedge \\ & & \qquad\qquad c\,i \in Acc(B, \{\ d\,j \mid dY(j) \in Acc(A,I)\})\ \}\,. \end{array}$$

To each (A,I), we associate a new non-terminal A_I of arity $\varrho(A)$, and we define the following grammar:

$$S \ := \ \{\ (A_I 1 \ldots \varrho(A)\,,\ H_{A,I}) \mid A \in N_R \ \wedge\ I \subseteq V_H\,[\varrho(A)]\ \}$$

where

$$\begin{array}{rcl} H_{A,I} & := & \big(Acc(A,I) - V_H\,[\varrho(A)]\big) \\ & \cup & \{\ B_{\{\ d\,j \ \mid\ dY(j)\ \in\ Acc(A,I)\ \}}Y \mid BY \in H_A \ \wedge\ B \in N_R\ \}\,. \end{array}$$

Thus S generates from $Z_\varnothing$ the hypergraph $G \otimes H$. Q.E.D. (Proposition 4.5)

We now consider the generation by accessibility. Taking any hypergraph G (whose vertices are) accessible from a given colour i, we map each vertex s to the minimum path length $g(s)$ to access s from i; precisely and inductively

$$\begin{array}{rcl} g^{-1}(0) & = & V_{G,i} \\ g^{-1}(n+1) & = & Succ_G(g^{-1}(\leq n)) \ - \ g^{-1}(\leq n) \end{array}$$

where $g^{-1}(\leq n) := g^{-1}(0) \cup \ldots \cup g^{-1}(n)$.
For instance the graph of Figure 3.20 is regular by accessibility.

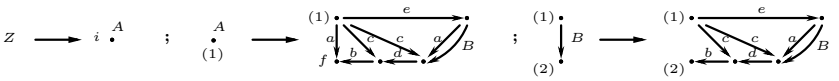


Figure 4.8. Generating the graph of Figure 3.20 by accessibility from i.

Note that any hypergraph which is regular by accessibility is of finite out-degree and has a finite number of vertices coloured by the initial colour.
In the figure below, we give a regular graph of finite degree, accessible from a colour, and which is not regular by accessibility from this colour.

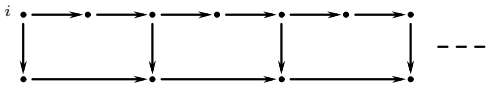


Figure 4.9. Regular graph not regular by accessibility from i.

4.4 Regularity by distance

Another usual graduation is the *distance* from a given vertex set E:

$$\begin{array}{rcl} d_G(s,E) & := & min\{\ d_G(s,t) \mid t \in E\ \} \\ \text{where} \quad d_G(s,t) & := & min(\{\ n \mid s \underset{G}{\longleftrightarrow}^n t\ \} \cup \{\omega\}). \end{array}$$

For instance the regular graph of Figure 2.7 remains regular by distance from the vertices coloured by 1 or 2 using the following outside grammar:

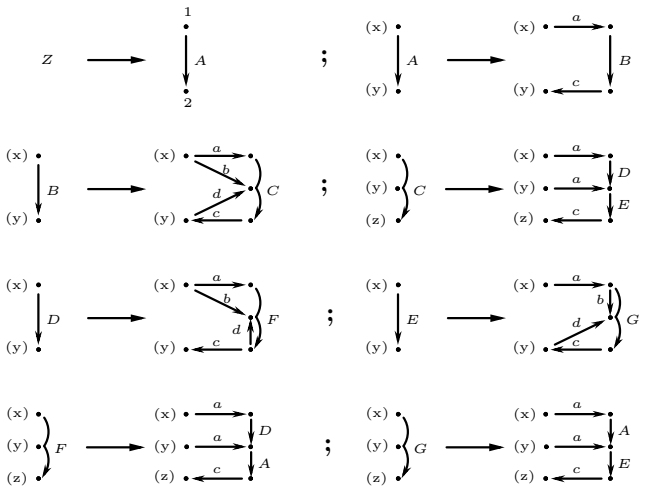


Figure 4.10. Grammar generating the graph of Figure 2.7 by distance.

We denote $d_G(s,i) := d_G(s, V_{G,i})$ the distance in a hypergraph G of a vertex s to the set of vertices coloured by i. Note that the n-th frontier of G by distance from i satisfies

$$\partial_{g,n} G \quad = \quad \{\ s \in V_{G-G_{d,n}} \mid d(s,i) = n\ \}\,.$$

We say that G is *finitely connected* by i if there is only a finite number of vertices coloured by i, and from which all vertices are connected: $V_{G,i}$ is finite and $d(s,i) < \omega$ for any $s \in V_G$.

Any grammar generating a hypergraph G of finite degree and finitely connected from a colour i, can be transformed in an effective way into a grammar generating G by distance from i. Such a graph G is also bounded connected by distance.

Theorem 4.6. *Any finitely connected regular hypergraph of finite degree is finitely decomposable by distance.*

Proof.

In part (i), we introduce the notion of frontier and of interface that allow to uniquely characterize any subset of hyperarcs in a hypergraph. Taking a regular hypergraph G finitely connected and of finite degree, we construct in part (ii) the *canonical* grammar generating G by distance. Part (iii) shows that this canonical grammar is indeed finite. Using (i)–(iii), we got that G is regular by distance. In (iv), we show that G is bounded connected by distance, and hence using Proposition 4.3, we deduce that G is

finitely decomposable by distance.

i) Let G be any hypergraph.
Consider any sub-hypergraph $H \subset G$ such that
for any connected component C of G, $H \cap C \neq C$.
Such a hypergraph H is characterized by its *frontier*:
$$Fr_G(H) := V_H \cap V_{G-H}$$
and by its *interface*:
$$\begin{aligned} In_G(H) &:= \{ X \in H \mid V_X \cap Fr_G(H) \neq \varnothing \} \\ &= \{ X \in H \mid V_X \cap V_{G-H} \neq \varnothing \}; \end{aligned}$$
in particular $Fr_G(H) \subseteq V_{In_G(H)}$.
The charaterization of H by $Fr_G(H)$ and $In_G(H)$ follows by this equality:
$$H = G\!<\!In_G(H), Fr_G(H)\!>$$
where for any $K \subseteq G$ and any $P \subseteq V_G$, the hypergraph $G\!<\!K, P\!>$ is the least fixed point of the following equation:
$$G\!<\!K, P\!> \;=\; K \cup \{ X \in G \mid V_X \cap V_{G\!<\!K,\,P\!>} \neq \varnothing \;\wedge\; V_X \cap P = \varnothing \}.$$

ii) Let R be a grammar generating a finite degree hypergraph G finitely connected by a colour ι.
We want to show that G is regular by distance d from ι:
$$d(s) := d(s, \iota) \quad \text{for any vertex } s \text{ of } G.$$
By Theorem 3.12, we can assume that R is complete outside and connected.
Up to a label renaming with adding rules, we assume that each right hand side has no two non-terminal hyperarcs with the same label, and we denote by V_R the set of non input vertices of the right hand sides of R:
$$V_R := \bigcup \{ V_H - V_X \mid (X, H) \in R \}.$$
Let $Z = H_0 \underset{R}{\Longrightarrow} H_1 \ldots H_n \underset{R}{\Longrightarrow} H_{n+1} \underset{R}{\Longrightarrow} \ldots$ be the derivation generating $\mathrm{Gen}(R)$: $\bigcup_{n \geq 0} [H_n] = \mathrm{Gen}(R)$.
As the set $V_{G,\iota}$ of vertices of G coloured by ι is finite, we denote by m the minimal derivation length to get all the vertices of G coloured by ι:
$$m := min\{ n \mid \forall\, p > n,\ (H_p - H_n) \cap \iota V_{H_p} = \varnothing \}.$$
As G is of finite degree and R is degree-outside, each rule of R has no output which is an input, hence
$$\mathrm{Gen}(R)_{d,n} \subseteq [H_{m+n}] \quad \text{for every } n \geq 0.$$
For every $n \geq 0$, we get
$$\partial_{d,n}\mathrm{Gen}(R) = \{ s \in V_{H_{m+n} - Gen(R)_{d,n}} \mid d(s) = n \}.$$
For every $n \geq 0$, we denote by $\{P_{n,1}, \ldots, P_{n,r_n}\}$ the partition of $\partial_{d,n}\mathrm{Gen}(R)$ into connected vertices of $\mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n}$ *i.e.* of $H_{m+n} - \mathrm{Gen}(R)_{d,n}$, and for every $1 \leq i \leq r_n$,
$$\begin{aligned} K_{n,i} &:= \{ X \in \mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n} \mid V_X \cap P_{n,i} \neq \varnothing \} \\ &= \{ X \in [H_{m+n+1}] - \mathrm{Gen}(R)_{d,n} \mid V_X \cap P_{n,i} \neq \varnothing \}. \end{aligned}$$

Thus for every $n \geq 0$,

$$\mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n} \;=\; \bigcup_{i=1}^{r_n} \mathrm{Gen}(R)\!<\!K_{n,i}\,,\,P_{n,i}\!>\,.$$

The left *residual* of $C \subseteq \mathrm{Gen}(R)$ by $u \in N_R^*$ is

$$u^{-1}C \;:=\; \{\; f u_1 \ldots u_{\varrho(f)} \mid f(uu_1)\ldots(uu_{\varrho(f)}) \in C \;\}$$

and p_C is the greatest common prefix in N_R^* of the vertices of C.
We take a linear ordering $<$ on $N_R \cup V_R$ that we extend on $N_R^* V_R$ by length lexicographic order.
For any $n \geq 0$ and $1 \leq i \leq r_n$, we define $p_{n,i} := p_{K_{n,i}}$ and we define the hyperarc

$$X_{n,i} \;:=\; (p_{n,i}^{-1}K_{n,i}\,,\,p_{n,i}^{-1}P_{n,i})s_1 \ldots s_q$$

with $\{s_1, \ldots, s_q\} = P_{n,i}$ and $s_1 > \ldots > s_q$;
note that the label is a pair of a finite graph with a vertex subset whose cardinal is the arity of the label.
We define the grammar $S := \bigcup_{n \geq 0} S_n$ with

$$\begin{aligned} S_0 \;&:=\; \{\; (\,Z\,,\,\mathrm{Gen}(R)_{d,0} \cup \{X_{0,1}, \ldots, X_{0,r_0}\}\,)\;\} \\ \forall\, n \geq 0,\; S_{n+1} \;&:=\; S_n \;\cup\; \{\; (X_{n,i}\,,\,K_{n,i} \cup \\ &\qquad \bigcup\{\; X_{n+1,j} \mid P_{n+1,j} \cap V_{K_{n,i}} \neq \varnothing \;\}) \mid \\ &\qquad\qquad 1 \leq i \leq r_n \;\wedge\; X_{n,i}(1) \notin N_{S_n} \;\}. \end{aligned}$$

The finiteness of S is shown in (iii).
For any $n \geq 0$ and $1 \leq i \leq r_n$, S generates from $X_{n,i}$ and by distance from ι the connected component of $\mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n}$ containing $P_{n,i}$.
Thus S generates from Z the hypergraph $\mathrm{Gen}(R)$ by distance from ι.

iii) Let us show that S is finite.
This is obtained by giving a bound b such that $d_{Gen(R)}(s,t) \leq b$ for any $n \geq 0$, any connected component C of $\mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n}$ and any $s,t \in V_C \cap \partial_{d,n}\mathrm{Gen}(R)$.
It is sufficient to extract such a bound for any $n \geq n_0$ with n_0 the smallest integer such that $\mathrm{Gen}(R)_{d,n_0} \supseteq [H_m]$.
As R is a connected grammar, we take the following integer:

$$c \;:=\; max\{\; d_{R^\omega(H)}(s,t) \mid H \in Im(R) \;\wedge\; s,t \in V_H \;\}\,.$$

Let $n \geq n_0$. Let C be a connected component of $\mathrm{Gen}(R) - \mathrm{Gen}(R)_{d,n}$ and let $s,t \in V_C$ with $d(s) = n = d(t)$.
We take a vertex z of C of minimal length. As $z \in V_C$, we have $d(z) \geq n$.
By definition of $\mathrm{Gen}(R)$, $z = wr$ for $w \in N_R^*$ and r a vertex of a right hand side of R.
Consider an undirected path of minimal length from s (resp. t) to ι; such a path goes through a vertex $x = wp$ (resp. $y = wq$) for some vertex p (resp. q) of a right hand side of R. Hence

$$d(x,y) \leq c \;\;,\;\; d(x,z) \leq c \;\;,\;\; d(y,z) \leq c$$

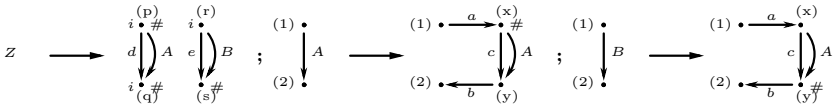
for distances on $\mathrm{Gen}(R)$. Thus

$$d(s,x)+d(x) \;=\; d(s) \;\leq\; d(z) \;\leq\; d(z,x)+d(x) \;\leq\; c+d(x)$$
so $d(s,x)\leq c$. Similarly $d(t,y)\leq c$. Finally
$$d(s,t) \;\leq\; d(s,x)+d(x,y)+d(y,t) \;\leq\; 3c.$$
Finally $b=3c$ suits (for any $n\geq n_0$).

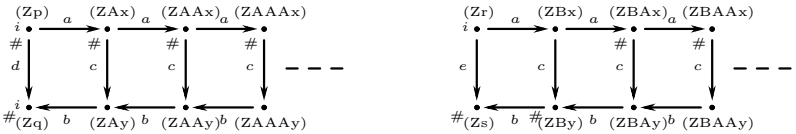
iv) By Proposition 4.3, it remains to verify that G is bounded connected by d.
Let C be a connected component of G^d_{n+1} for some $n\geq 0$.
So $C' := C - \mathbb{N}V_C$ is a connected component of $G - G_{d,n}$ with
$$V_{C,0} \;=\; V_{C'} \cap \partial_{d,n}G\,.$$
By (iii) we get $d_G(s,t)\leq b$ for any $s,t\in V_{C,0}$.
As G is of finite degree, let D be the maximum degree of its vertices.
Thus for any connected component C of $\sum_{n\geq 1} G^d_n$, we have
$$|V_{C,0}| \;\leq\; D^0+D^1+\ldots+D^b$$
meaning that G is bounded connected by d. Q.E.D. (Theorem 4.6)

The generation by distance is illustrated below.

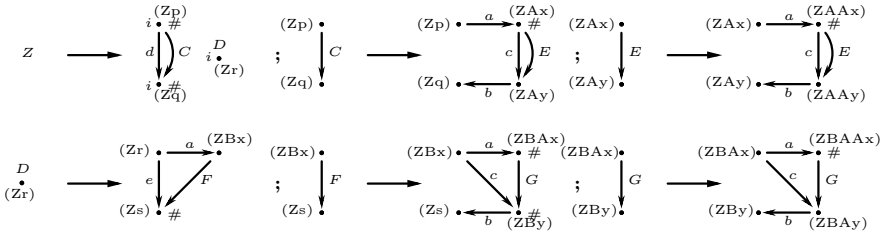
Taking grammar R of Figure 4.1



its canonical graph $Gen(R)$ is



The construction of Theorem 4.6 gives the grammar:



with $x>y$ and $p>q$ and

$$
\begin{array}{lcl}
C & = & \big(\{p \xrightarrow{a} Ax\,,\ Ay \xrightarrow{b} q\}\,,\ \{p,q\}\big) \\
D & = & \big(\{r \xrightarrow{a} Bx\,,\ r \xrightarrow{e} s\}\,,\ \{r\}\big) \\
E & = & \big(\{x \xrightarrow{a} Ax\,,\ Ay \xrightarrow{b} y\}\,,\ \{x,y\}\big) \\
F & = & \big(\{Bx \xrightarrow{a} BAx\,,\ Bx \xrightarrow{c} By\,,\ By \xrightarrow{b} s\}\,,\ \{Bx,s\}\big) \\
G & = & \big(\{Ax \xrightarrow{a} AAx\,,\ Ax \xrightarrow{c} Ay\,,\ Ay \xrightarrow{b} y\}\,,\ \{Ax,y\}\big)
\end{array}
$$

Figure 4.11. Generation by distance.

5 Graph grammars and pushdown automata

A pushdown automaton is a particular case of a labelled word rewriting system whose rules are only applied by suffix. Pushdown automata even in a weak form and the rewriting systems define the same graphs by suffix rewriting, which are exactly the regular graphs of bounded degree (cf. Theorem 5.11).

5.1 Suffix transition graphs

A labelled word rewriting system is just a finite uncoloured graph whose vertices are words. Its set of unlabelled suffix transitions is the suffix rewriting relation, whose transitive closure is a rational relation (cf. Proposition 5.2). Its set of labelled suffix transitions is called a suffix graph. Any regular restriction of this graph is regular by length (cf. Theorem 5.6). Conversely any regular graph of finite degree is a regular restriction of a suffix graph (cf. Theorem 5.8).

We fix a countable set T of symbols, called *terminals*.
A labelled word *rewriting system* S is a finite subset of $N^* \times T \times N^*$ where N is an arbitrary alphabet of *non-terminals*; we write $u \xrightarrow[S]{a} v$ for $(u,a,v) \in S$, and define

$$
\begin{array}{rcll}
Dom(S) & := & \{\ u \mid \exists\ a \in T\ \ \exists\ v \in N^*,\ u \xrightarrow[S]{a} v\ \} & \text{its } \textit{left hand sides}, \\
Im(S) & := & \{\ v \mid \exists\ a \in T\ \ \exists\ u \in N^*,\ u \xrightarrow[S]{a} v\ \} & \text{its } \textit{right hand sides}, \\
W_S & := & Dom(S) \cup Im(S) & \text{the } \textit{words} \text{ of } S, \\
N_S & := & \{\ u(i) \mid u \in W_S\ \wedge\ 1 \le i \le |u|\ \} & \text{its } \textit{non-terminals}, \\
T_S & := & \{\ a \in T \mid \exists\ u,v \in N^*,\ u \xrightarrow[S]{a} v\ \} & \text{its } \textit{terminals}.
\end{array}
$$

Rewritings in a rewriting system are generally defined as applications of rewriting rules in every context. We are only concerned with suffix rewriting.
Given a rewriting system S and a terminal $a \in T_S$, we call *labelled suffix rewriting* $\xrightarrow[S]{a}$ the binary relation on N_S^* defined by

$$wu \xrightarrow[S]{a} wv \quad \text{for any} \quad u \xrightarrow[S]{a} v \ \text{ and } \ w \in N_S^* .$$

Example 5.1. Consider the rewriting system $S = \{\varepsilon \xrightarrow{1} ab\,,\ bab \xrightarrow{2} ab\}$. We have

$$bb \xrightarrow[S]{1} bbab \xrightarrow[S]{2} bab \xrightarrow[S]{2} ab \xrightarrow[S]{1} abab \xrightarrow[S]{2} aab \ \ldots$$

For any rewriting system S, the unlabelled *suffix rewriting* is

$$\xrightarrow[S]{} \ := \ \bigcup_{a\in T_S} \xrightarrow[S]{a} \ = \ \{\ wu \longrightarrow wv \mid u \xrightarrow[S]{a} v \ \wedge \ w \in N_S^*\ \}$$

and its reflexive and transitive closure (by composition) $\xrightarrow[S]{}^*$ is the *suffix derivation*. In Example 5.1, we have $bb \xrightarrow[S]{}^* bb$ and $bb \xrightarrow[S]{}^* ab$.

We denote by $\xrightarrow[S]{}^+ \ = \ \xrightarrow[S]{} \circ \xrightarrow[S]{}^*$ the transitive closure of $\xrightarrow[S]{}$.

A well-known property is that the set of words deriving by suffix from a given word is a regular language, and a finite automaton accepting it is effectively constructible [Bü 64]. This property remains true starting from any regular set of words. More generally, the suffix derivation is itself a rational relation: it can be recognized by a transducer *i.e.* a finite automaton labelled by pairs of words.

Proposition 5.2 ([Ca 90])**.** *The suffix derivation of any word rewriting system is effectively a rational relation.*

Proof. We give here a construction improved by Carayol.

i) Let N be any alphabet. For any $P \subseteq N^*$ and for any word $u \in N^*$, we denote by $u{\downarrow}P$ the set of irreducible words obtained from u by derivation according to $P{\times}\{\varepsilon\}$:

$$u{\downarrow}P \ := \ \{\ v \mid u \xrightarrow[P\times\{\varepsilon\}]{}{}^* v \not\xrightarrow[P\times\{\varepsilon\}]{}\ \}\,.$$

We extend by union $\downarrow P$ to any language $L \subseteq N^*$:

$$L{\downarrow}P \ := \ \bigcup\{\ u{\downarrow}P \mid u \in L\ \}.$$

A standard result due to Benois [Be 69] is that for P regular, the operation $\downarrow P$ preserves regularity:

$$L, P \in \mathrm{Rat}(N^*) \quad \Longrightarrow \quad L{\downarrow}P \in \mathrm{Rat}(N^*).$$

Precisely, we have

$$L{\downarrow}P \ = \ \xrightarrow[P\times\{\varepsilon\}]{}{}^*(L) \ - \ N^*PN^*\,.$$

It remains to show that the image $\xrightarrow[P\times\{\varepsilon\}]{}{}^*(L)$ of L by the derivation $\xrightarrow[P\times\{\varepsilon\}]{}{}^*$ is regular. This property is true even if P is not regular.

Precisely and for L regular, there is a *finite automaton* $A \subseteq Q{\times}N{\times}Q$ recognizing L from an *initial state* $i \in Q$ to a subset $F \subseteq Q$ of *final states*: $L(A, i, F) = L$.

By adding iteratively ϵ-transitions between states linked by a path labelled in P, we complete A into an automaton B which is the least fixpoint of

the following equation:

$$B \;=\; A \cup \{\, p \xrightarrow{\epsilon} q \mid \exists\, u \in P,\ p \underset{B}{\overset{u}{\Longrightarrow}} q \,\}.$$

Note that we can refine B by saturating A with only elementary ε-transitions:

$$\begin{aligned} B \;=\; A \;&\cup\; \{\, p \xrightarrow{\varepsilon} q \mid p \neq q \ \wedge\ \exists\, a \in P \cap N,\ p \underset{A}{\xrightarrow{a}} q \,\} \\ &\cup\; \{\, p \xrightarrow{\varepsilon} q \mid p \neq q \ \wedge\ \exists\, aub \in P,\ a,b \in N \\ &\qquad\qquad \wedge\ p \underset{A}{\xrightarrow{a}} \underset{B}{\overset{u}{\Longrightarrow}} \underset{A}{\xrightarrow{b}} q \,\}. \end{aligned}$$

So $L(B,i,F) \;=\; \underset{P\times\{\varepsilon\}}{\longrightarrow}{}^{*}(L)$.

ii) We denote N_S by N and to each letter $x \in N$, we associate a new symbol $\overline{x} \notin N$ with $\overline{x} \neq \overline{y}$ for $x \neq y$. Let $\overline{N} := \{\, \overline{x} \mid x \in N \,\}$.
We extend by morphism $\overline{\ }$ to any word $u = x_1 \ldots x_n$ *i.e.* $\overline{u} = \overline{x_1} \ldots \overline{x_n}$.
Recall that the mirror $\widetilde{u}$ of any word $u = x_1 \ldots x_n$ is the word $\widetilde{u} = x_n \ldots x_1$.
The following set is regular:

$$\Big[\{\, \widetilde{\overline{u}}\, v \mid \exists\, a,\ u \underset{S}{\xrightarrow{a}} v \,\}^{*}\Big)\!\downarrow\!\{\, x\overline{x} \mid x \in N \,\}\Big] \;\cap\; \overline{N}^{*}N^{*}$$

meaning that we can apply by suffix a rule (u,v) by producing on the right v after having removed u on the right (using $\downarrow \{\, x\overline{x} \mid x \in N \,\}$).
This set can be written as a finite union $\bigcup_{i\in I} \overline{U}_i V_i$ where U_i, $V_i \in Rat(N^*)$ for all $i \in I$. Taking the following relation:

$$\overline{S} \;:=\; \bigcup_{i\in I} \widetilde{U_i} \times V_i$$

it is easy to verify that the suffix derivation according to S is the suffix rewriting according to $\overline{S}$:

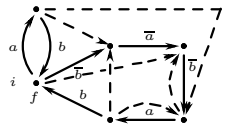
$$\underset{S}{\longrightarrow\!\!\!\!\!\rightarrow}{}^{*} \;=\; \underset{\overline{S}}{\longrightarrow\!\!\!\!\!\rightarrow}.$$

It follows that $\underset{S}{\longrightarrow\!\!\!\!\!\rightarrow}{}^{*}$ is an effective rational relation.
In particular starting from $I \in \mathrm{Rat}(N^*)$, we have

$$\underset{S}{\longrightarrow\!\!\!\!\!\rightarrow}{}^{*}(I) \;=\; \underset{\overline{S}}{\longrightarrow\!\!\!\!\!\rightarrow}(I) \;=\; Im\big(\underset{\overline{S}}{\longrightarrow\!\!\!\!\!\rightarrow} \cap\ I\times N^*\big) \;\in Rat(N^*)$$

Q.E.D. (Proposition 5.2)

Taking the system $S = \{\varepsilon \xrightarrow{1} ab\,,\, bab \xrightarrow{2} ab\}$ of Example 5.1, the construction of Proposition 5.2 gives the following finite automaton where the dashed arrows are ε-transitions:



which gives the suffix derivation of S:

$$\xrightarrow[S]{}{}^{*} \;=\; \{\varepsilon\}\times(a^+b)^* \;\cup\; b^+ab\times(a^+b)^+ \;\cup\; b^+\times(a^+b)^+ .$$

To any rewriting system S, we associate its *suffix graph*:

$$\mathrm{Suff}(S) \;:=\; \{\; wu \xrightarrow{a} wv \mid u \xrightarrow[S]{a} v \;\wedge\; w\in N_S^* \;\} \;=\; N_S^*.S$$

which is the set of its suffix transitions.
For instance the suffix graph of $\{x \xrightarrow{a} \varepsilon\,,\; x \xrightarrow{b} x^4\}$ is the regular graph of Figure 2.8.
The suffix graph of $\{x \xrightarrow{a} \varepsilon\,,\; x \xrightarrow{b} zxyx\,,\; y \xrightarrow{c} \varepsilon\,,\; z \xrightarrow{d} \varepsilon\}$ restricted to the set $(z + zxy)^*(\varepsilon + x)$ of its vertices accessible from x is the graph of Figure 2.9.
The suffix transition graphs of word rewriting systems have bounded degree.

Lemma 5.3. *The suffix graph of any rewriting system has bounded degree, and has a finite number of non isomorphic connected components.*

Proof. Let S be any labelled word rewriting system.

i) Let us verify that $\mathrm{Suff}(S) = N_S^*.S$ has bounded degree.
Let w be any vertex of this graph.
As we can at most apply all the rules of S, the out-degree of w is bounded by the number of rules: $d^+(w) \le |S|$.
Note that the inverse of $N_S^*.S$ is the suffix graph of the inverse of S:

$$\left(N_S^*.S\right)^{-1} \;=\; N_S^*.S^{-1},$$

so the in-degree of w is its out-degree for $N_S^*.S^{-1}$, hence

$$d^-(w) \;\le\; |S^{-1}| \;=\; |S|.$$

Finally the degree of w satisfies: $d(w) = d^+(w) + d^-(w) \le 2\,|S|$.

ii) We show that $N_S^*.S$ has a finite number of non isomorphic connected components.
Let H be any connected component of $N_S^*.S$. Let $w \in N_S^*$ such that

$$w.W_S \;\cap\; V_H \;\neq\; \varnothing \quad \text{and of length } |w| \text{ minimal.}$$

Such a word w is unique because it is prefix of all the vertices of H:

- by definition of w, there is $u \in W_S$ such that $wu \in V_H$;
- by induction on the length of any derivation $wu \underset{H\,\cup\,H^{-1}}{\Longrightarrow}{}^{*} v$, w is prefix of v.

By removing this common prefix to the vertices of H, we obtain the graph

$$w^{-1}H \;:=\; \{\; u \xrightarrow{a} v \mid wu \xrightarrow[H]{a} wv \;\}$$

which is isomorphic to H and has a vertex in W_S which is finite.
So the set of connected components of $\mathrm{Suff}(S)$ is finite up to isomorphism.

Q.E.D. (Lemma 5.3)

By Proposition 3.4, the second property of Lemma 5.3 is a particular case of the fact that any suffix graph is regular.

Proposition 5.4. *The suffix graph of any rewriting system can be generated by a one-rule graph grammar from its left hand side.*

Proof. Let S be any labelled word rewriting system. Let
$$E \;:=\; \{\; y \mid \exists\, x \neq \varepsilon,\; xy \in W_S \;\}$$
be the set of strict suffixes of the words of S.
We take a label Y of arity $n = |E|$ and let $\{e_1, \ldots, e_n\} = E$.
We define the grammar R restricted to the following rule:
$$Y e_1 \ldots e_n \;\longrightarrow\; S \;\cup\; \{\; Y(xe_1)\ldots(xe_n) \mid x \in N_S \;\}\,.$$
So $N_S^*.S$ is generated by R from its left hand side: $N_S^*.S \;\in\; R^\omega(Y e_1 \ldots e_n)$.
Q.E.D. (Proposition 5.4)

Taking the system $S \;=\; \{\varepsilon \xrightarrow{1} ab\,,\, bab \xrightarrow{2} ab\}$ of Example 5.1 and by applying the construction of Proposition 5.4, we get the following one-rule grammar generating the suffix graph of S.

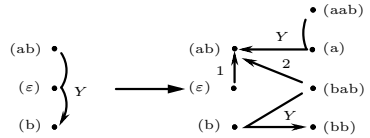


Figure 5.1. Generating the suffix graph of the system of Example 5.1.

The regularity of any suffix graph is preserved by any regular restriction.

Corollary 5.5. *Any regular restriction of a suffix graph is a regular graph.*

Proof. Let S be any labelled word rewriting system and let $P \in \mathrm{Rat}(N_S^*)$ be any regular language.
We want to show that $\mathrm{Suff}(S)_{|P}$ is a regular graph.
We can assume that each non-terminal of S is not a terminal and is an edge label: $N_S \subset F_2 - T_S$.
We complete S into the following word rewriting system:
$$\overline{S} \;:=\; S \;\cup\; \{\; \varepsilon \xrightarrow{x} x \mid x \in N_S \;\}.$$
It follows that
$$\mathrm{Suff}(\overline{S}) \;=\; \mathrm{Suff}(S) \;\cup\; \{\; u \xrightarrow{x} ux \mid u \in N_S^* \;\wedge\; x \in N_S \;\}.$$
As P is regular, there exists a finite graph H labelled in N_S which recognizes P from an initial vertex i to a vertex subset F: $L(H, i, F) = P$.
We can assume that the vertices of H are vertex colours: $V_H \subset F_1$.
By Proposition 5.4, $\mathrm{Suff}(\overline{S})$ is a regular graph.
We take a new colour $\iota \in F_1 - V_H$. By Proposition 4.5, the graph
$$G \;:=\; \mathrm{Suff}(\overline{S}) \;\cup \big(\{\iota\,\varepsilon\} \otimes (H \cup \{\iota\, i\})\big)$$
remains regular. By removing in G the arcs labelled in N_S, we get the graph
$$G' \;:=\; G \;-\; V_G \times N_S \times V_G$$

which is regular (it suffices to remove the arcs labelled in N_S in the grammar generating G).
By Proposition 4.1, the restriction of G' to the vertices coloured in F is again a regular graph G''. By removing all vertex colours from G'', we get $\mathrm{Suff}(S)_{|P}$ which is regular. Q.E.D. (Corollary 5.5)

The regularity of suffix graphs can also be obtained by vertex length.

Theorem 5.6. *Any regular restriction of a suffix graph is regular by length.*

Proof. We begin as in Corollary 5.5.
Let S be any labelled word rewriting system and let $P \in \mathrm{Rat}(N_S^*)$ be any regular language. We want to show that $\mathrm{Suff}(S)_{|P}$ is regular by vertex length.
We can assume that each non-terminal of S is not a terminal and is a label colour: $N_S \subset F_1 - T_S$
and we complete S into the following word rewriting system:

$$\overline{S} \;:=\; S \;\cup\; \{\; \varepsilon \xrightarrow{x} x \mid x \in N_S \;\}.$$

We get

$$\mathrm{Suff}(\overline{S}) \;=\; \mathrm{Suff}(S) \;\cup\; \{\; u \xrightarrow{x} ux \mid u \in N_S^* \;\wedge\; x \in N_S \;\}.$$

In particular $V_{\mathrm{Suff}(\overline{S})} = N_S^*$ and we define

$$m \;:=\; max\{\; |u| \mid u \in W_{\overline{S}} \;\}\,.$$

As P is regular, there is a finite complete graph H labelled in N_S which recognizes P from an initial vertex ι to a vertex subset F: $L(H, \iota, F) = P$.
We can assume that the vertices of H are vertex colours: $V_H \subset F_1$. We define

$$H(P) \;:=\; \{\; c\,u \mid u \in P \;\wedge\; \iota \xRightarrow[H]{u} c \;\} \quad \text{for any} \quad P \subseteq N_S^*\,.$$

i) Let us show that $\mathrm{Suff}(\overline{S}) \cup H(N_S^*)$ is regular by length.
For any $n \geq 0$, we define

$$S_n \;:=\; \{\; zx \xrightarrow{a} zy \mid x \xrightarrow[\overline{S}]{a} y \;\wedge\; min\{|zx|, |zy|\} \leq n < max\{|zx|, |zy|\} \;\}$$

in such a way that

$$\mathrm{Suff}(\overline{S}) - \mathrm{Suff}(\overline{S})_{|\;|,n} \;=\; \mathrm{Suff}(S_n).$$

For every $n \geq 0$, we get

$$\partial_{|\;|,n}\mathrm{Suff}(\overline{S}) \;=\; \{\; u \in N_S^*.\big(Dom(S_n) \cup Im(S_n)\big) \mid |u| \leq n \;\}$$

and we can compute $\{P_{n,1}, \ldots, P_{n,r_n}\}$ the partition of $\partial_{|\;|,n}\mathrm{Suff}(\overline{S})$ into connected vertices of $\mathrm{Suff}(\overline{S}) - \mathrm{Suff}(\overline{S})_{|\;|,n}$, and for every $1 \leq i \leq r_n$,

$$K_{n,i} \;:=\; \{\; u \xrightarrow[\mathrm{Suff}(S_n)]{a} v \mid \{u, v\} \cap P_{n,i} \neq \varnothing \;\wedge\; max\{|u|, |v|\} = n + 1 \;\}\,.$$

Thus with the notation (i) of the proof of Theorem 4.6, we have for every $n \geq 0$,

$$\mathrm{Suff}(\overline{S}) - \mathrm{Suff}(\overline{S})_{|\;|,n} \;=\; \bigcup_{i=1}^{r_n} \mathrm{Suff}(\overline{S}) \!<\! K_{n,i}\,,\, P_{n,i} \!>\,.$$

We take a linear ordering $<$ on N_S that we extend on N_S^* by length-lexicographic order.
For any $n \geq 0$ and $1 \leq i \leq r_n$, we take

$$p_{n,i} := min\{\ |u| - m \mid u \in P_{n,i} \ \wedge\ |u| \geq m\ \}$$

which is a common prefix of the words in $P_{n,i}$, and we define the hyperarc

$$X_{n,i} := p_{n,i}^{-1} H(P_{n,i}) s_1 \ldots s_q$$

with $\{s_1, \ldots, s_q\} = P_{n,i}$ and $s_1 < \ldots < s_q$;
note that the label is a finite set of coloured vertices.
We define the grammar $R := \bigcup_{n \geq 0} R_n$ with

$$\begin{aligned} R_0 &:= \{\ (Z\,,\, (S \cap \{\varepsilon\} \times T_S \times \{\varepsilon\}) \cup \{\iota\,\varepsilon\,,\, X_{0,1}\}\ \} \\ \forall\ n \geq 0,\ R_{n+1} &:= R_n \ \cup\ \{\ (X_{n,i}\,,\, K_{n,i} \cup H(V_{K_{n,i}} - P_{n,i}) \cup \\ &\qquad \bigcup\{\ X_{n+1,j} \mid P_{n+1,j} \cap V_{K_{n,i}} \neq \varnothing\ \}) \mid \\ &\qquad\qquad 1 \leq i \leq r_n \ \wedge\ X_{n,i}(1) \notin N_{R_n}\ \}. \end{aligned}$$

The finiteness of R is shown in (ii).
For any $n \geq 0$ and any $1 \leq i \leq r_n$, R generates from $X_{n,i}$ and by vertex length, the connected component of $\big(\mathrm{Suff}(\overline{S}) - \mathrm{Suff}(\overline{S})_{|\ |,n}\big) \ \cup\ H(\{\, u \in N_S^* \,|\, |u| > n\,\})$ containing $P_{n,i}$. Thus R generates from axiom Z the graph $\mathrm{Suff}(\overline{S}) \cup H(N_S^*)$ by vertex length.

ii) Let us show that R is finite.
It is sufficient to show that $\{\ p_{n,i}^{-1} P_{n,i} \mid n \geq 0 \ \wedge\ 1 \leq i \leq r_n\ \}$ is finite.
Let $n \geq 0$ and $1 \leq i \leq r_n$.
We show that any word in $p_{n,i}^{-1} P_{n,i}$ has length at most $2m$.
Let $u, v \in P_{n,i}$. We have $|u| \leq n$.
There exist $z \in N_S^*$ and $x \xrightarrow[S_n \,\cup\, S_n^{-1}]{a} y$ with $v = zx$ and $|zy| > n$.
Hence $|u| - |v| = |u| - |zy| \leq n - (n - |y|) = |y| \leq m$.
Assume now that v is of minimal length.
Either $|v| \leq m$, so $p_{n,i} = \varepsilon$ and thus $|p_{n,i}^{-1} u| = |u| \leq m + |v| \leq 2m$.
Or $|v| > m$, then $v = wx$ for some w and $|x| = m$.
Thus $p_{n,i} = w$ and $|p_{n,i}^{-1} u| - |x| = |u| - |v| \leq m$
hence $|p_{n,i}^{-1} u| \leq m + |x| = 2m$.

iii) It remains to end as in the proof of Corollary 5.5.
We remove in R the arcs labelled in N_S and by Proposition 4.1, we restrict to the vertices coloured by F. Then we remove the colours and apply Lemma 3.2 to get a grammar generating $\mathrm{Suff}(S)_{|L}$ by length. Q.E.D. (5.6)

Starting with the system $S = \{\varepsilon \xrightarrow{a} xx\}$ and the language $L = x(xx)^*$ recognized by the complete automaton $\{i \xrightarrow{x} f\,,\, f \xrightarrow{x} i\}$ from i to f, the construction of Theorem 5.6 yields the grammar below, which generate $\mathrm{Suff}(S)_{|L} = \{\ x^{2n+1} \xrightarrow{a} x^{2n+3} \mid n \geq 0\ \}$ by length.

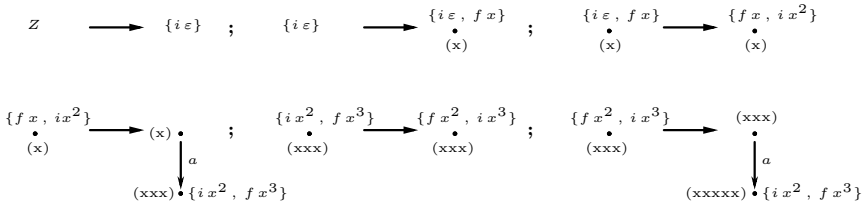


Figure 5.2. Generation by length of a regular restriction of a suffix graph.

In Subsection 3.5, we have associated to any grammar R a representant $\mathrm{Gen}(R)$ of its set of generated graphs. Any vertex of $\mathrm{Gen}(R)$ is the word of the non-terminals used to get it. This allows us to express $\mathrm{Gen}(R)$ as a suffix graph when it is of bounded degree.

Lemma 5.7. *Any grammar R generating a bounded degree uncoloured graph, can be transformed into a word rewriting system S such that any connected component (resp. any accessible subgraph) of $\mathrm{Gen}(R)$ is a connected component (resp. accessible subgraph) of $\mathrm{Suff}(S)$.*

Proof. To define $\mathrm{Gen}(R)$ simply, we assume that each right hand side has no two non-terminal hyperarcs with the same label.
We assume that the rule of any $A \in N_R$ is of the form: $A1 \ldots \varrho(A) \longrightarrow H_A$.
We write V_R the set of non input vertices of the right hand sides of R:

$$V_R \;:=\; \bigcup\{\, V_{H_A} - [\varrho(A)] \mid A \in N_R \,\}\,.$$

To each $A \in N_R$, let S_A be a graph of vertex set $V_{S_A} \subset N_R^+ V_R \cup [\varrho(A)]$ labelled in T_R such that the family of graphs S_A is the least fixed point of the following equations:

$$S_A \;=\; A \cdot \big([H_A] \cup \bigcup\{\, S_B[Y(1), \ldots, Y(\varrho(B))] \mid BY \in H_A \wedge B \in N_R \,\}\big)$$

where for any $A \in N_R$, for any graph G of vertex set $V_{S_A} \subset N_R^* V_R \cup [\varrho(A)]$ labelled in T_R and for any $a_1, \ldots, a_{\varrho(A)} \in V_R \cup [\varrho(R)]$, the *substitution* $G[a_1, \ldots, a_{\varrho(A)}]$ is the graph obtained from G by replacing in its vertices each $i \in [\varrho(A)]$ by a_i :

$$\begin{aligned} & G[a_1, \ldots, a_{\varrho(A)}] \\ := \;& \{\, u[a_1, \ldots, a_{\varrho(A)}] \xrightarrow{a} v[a_1, \ldots, a_{\varrho(A)}] \mid u \xrightarrow[G]{a} v \,\} \end{aligned}$$

with $u[a_1, \ldots, a_{\varrho(A)}] \;:=\; \begin{cases} a_i & \text{if } u = i \in [\varrho(A)] \\ u & \text{otherwise;} \end{cases}$

and where the *addition* $A{\cdot}G$ is defined by

$$A \cdot G \;:=\; \{\, A{\cdot}(u \xrightarrow{a} v) \mid u \xrightarrow[G]{a} v \,\}$$

and with $A \cdot (u \xrightarrow{a} v)$ defined by

$$\begin{cases} u \xrightarrow{a} v & \text{if } u,v \in [\varrho(A)] \ \vee \ u,v \notin [\varrho(A)] \cup V_R \\ Au \xrightarrow{a} v & \text{if } u \notin [\varrho(A)] \ \wedge \ v \in [\varrho(A)] \\ u \xrightarrow{a} Av & \text{if } u \in [\varrho(A)] \ \wedge \ v \notin [\varrho(A)] \\ Au \xrightarrow{a} Av & \text{if } u,v \notin [\varrho(A)] \ \wedge \ (u \in V_R \ \vee \ v \in V_R)\,. \end{cases}$$

The system $S = S_Z$ is suitable, for Z the axiom of R:

$$S_Z \ = \ \{ \ u \xrightarrow{a} v \ \mid \ min\{|u|,|v|\} = 2 \ \wedge \ \exists \ w, \ wu \xrightarrow[\mathrm{Gen}(R)]{a} wv \ \}.$$

Q.E.D. (Lemma 5.7)

Taking the grammar of Figure 3.20, the construction of Lemma 5.7 yields

$$\begin{aligned} S_Z &= Z \cdot (S_A[s,t]) \\ S_A &= A \cdot (\{1 \xrightarrow{a} 2\,,\ p \xrightarrow{b} 2\} \ \cup \ S_B[1,p]) \\ S_B &= B \cdot (\{1 \xrightarrow{c} 2\,,\ q \xrightarrow{d} 2\} \ \cup \ S_C[1,q]) \\ S_C &= C \cdot (\{1 \xrightarrow{c} 2\,,\ 1 \xrightarrow{e} r\} \ \cup \ S_A[r,2]) \end{aligned}$$

hence

$$\begin{aligned} S_Z \ = \ & \{Zs \xrightarrow{a} Zt\,,\ Zs \xrightarrow{c} ZAp\,,\ Zs \xrightarrow{c} ZABq\,,\ Zs \xrightarrow{e} ZABCr\} \\ \cup \ & \{ZAp \xrightarrow{b} Zt\,,\ ABq \xrightarrow{d} Ap\,,\ BCr \xrightarrow{a} Bq\,,\ BCAp \xrightarrow{b} Bq\} \\ \cup \ & \{Cr \xrightarrow{c} CAp\,,\ CBq \xrightarrow{d} Cp\,,\ Cr \xrightarrow{c} CABq\,,\ Cr \xrightarrow{e} CABCr\} \end{aligned}$$

Corollary 5.5 (or Theorem 5.6) and Lemma 5.7 imply the equality between the classes of suffix graphs and uncoloured regular graphs of bounded degree.

Theorem 5.8. *Considering the suffix graphs of labelled word rewriting systems, their connected components are the connected regular graphs of bounded degree,*
their accessible subgraphs are the rooted regular graphs of bounded degree,
their regular restrictions are the regular graphs of bounded degree.

Proof. **i)** Let S be any word rewriting system.
Let v be any vertex of $\mathrm{Suff}(S)$ *i.e.* $v \in N_S^*(Dom(S) \cup Im(S))$.
By Proposition 5.2, the set of vertices accessible from v is the regular language $\underset{S}{\longrightarrow\!\!\!|}{}^*(v)$, and the vertex set of the connected component of $\mathrm{Suff}(S)$ containing v is the regular language $\underset{S \,\cup\, S^{-1}}{\longrightarrow\!\!\!|}{}^*(v)$.
By Corollary 5.5, any regular restriction (resp. any accessible subgraph, any connected component) of $\mathrm{Suff}(S)$ is an uncoloured (resp. rooted, connected) regular graph of bounded degree.

ii) Let R be any grammar generating an uncoloured graph of bounded degree.
Let S be the word rewriting system constructed from R by Lemma 5.7.

In 5.1, we have seen that $\mathrm{Gen}(R)$ has a regular vertex set. By Lemma 5.7,

$$\mathrm{Gen}(R) \quad = \quad \mathrm{Suff}(S)_{|V_{\mathrm{Gen}(R)}}$$

hence $\mathrm{Gen}(R)$ is a regular restriction of a suffix graph.
Furthermore by Lemma 5.7, if $\mathrm{Gen}(R)$ is connected (resp. rooted) then it is a connected component (resp. accessible subgraph) of $\mathrm{Suff}(S)$. Q.E.D. (5.8)

We now restrict as much as possible the word rewriting systems to define the same suffix graphs.

5.2 Weak pushdown automata

A (real-time) *pushdown automaton* S over the alphabet T of terminals is a particular word rewriting system: S is a finite subset of $PQ\times T\times P^*Q$ where P, Q are disjoint alphabets of respectively *stack letters* and *states*; we denote by

$$\begin{array}{lcll} P_S & := & \{\ u(i) \mid 1 \leq i \leq |u| \ \wedge\ \exists\ q \in Q,\ uq \in W_S\ \} & \text{the } \textit{stack letters}, \\ Q_S & := & \{\ q \mid \exists\ u \in P^*,\ uq \in W_S\ \} & \text{the } \textit{states} \text{ of } S. \end{array}$$

A *configuration* of S is a word in $P_S^*.Q_S$: a stack word followed by a state.
The *transition graph* of S is the set of its transitions restricted to its configurations:

$$\mathrm{Tr}(S) \ := \ \{\ wu \xrightarrow{a} wv \mid u \xrightarrow[S]{a} v \ \wedge\ w \in P_S^*\ \} \quad = \quad P_S^*.S$$

it is also the suffix graph of S restricted to its configurations.
Note that a pushdown automaton is essentially a labelled word rewriting system whose left hand sides are of length 2 and such that the rules are only applied by suffix. A symmetrical way to normalize both sides of the rules of a rewriting system is given by a *weak pushdown automaton* S which is a finite set of rules of the form:

$$p \xrightarrow{a} q \quad \text{or} \quad p \xrightarrow{a} xq \quad \text{or} \quad xp \xrightarrow{a} q \quad \text{with} \quad x \in P,\ p, q \in Q,\ a \in T$$

where P and Q are disjoint alphabets of stack letters and states; we also write P_S and Q_S for respectively the stack letters and the states (appearing in the rules) of S. The *transition graph* of S is also the set of its (suffix) transitions restricted to its configurations: $\mathrm{Tr}(S) := P_S^*.S$.

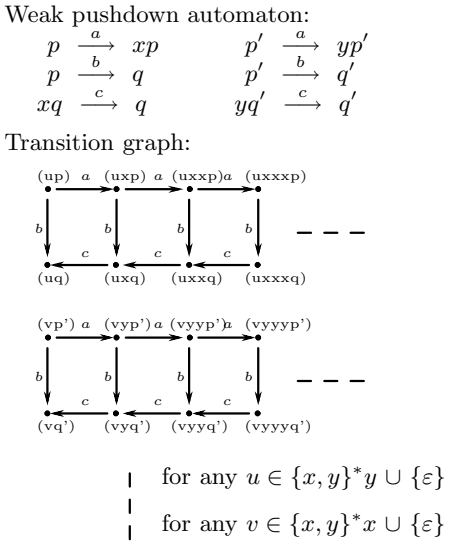


Figure 5.3. The transition graph of a weak pushdown automaton.

We define the same suffix graphs by normalizing labelled word rewriting systems as pushdown automata or weak pushdown automata.

Theorem 5.9. *The suffix graphs of labelled word rewriting systems, the transition graphs of pushdown automata, and the transition graphs of weak pushdown automata,*
have up to isomorphism the same connected components, the same accessible subgraphs and the same regular restrictions.

Proof. **i)** Let S be any weak pushdown automaton.
Let us construct a pushdown automaton $\overline{S}$ simulating S: the connected components (resp. accessible subgraphs, regular restrictions) of $\mathrm{Tr}(S)$ are connected components (resp. accessible subgraphs, regular restrictions) of $\mathrm{Tr}(\overline{S})$.
We take a new symbol $\perp$ and we define the pushdown automaton:

$$\begin{aligned}
\overline{S} \quad := \quad & \{\, yp \xrightarrow{a} yxq \mid p \xrightarrow[S]{a} xq \ \wedge\ y \in N_S \cup \{\perp\} \,\} \\
\cup \quad & \{\, yp \xrightarrow{a} yq \mid p \xrightarrow[S]{a} q \ \wedge\ y \in N_S \cup \{\perp\} \,\} \\
\cup \quad & \{\, xp \xrightarrow{a} q \mid xp \xrightarrow[S]{a} q \,\}\,.
\end{aligned}$$

Thus $P_{\overline{S}} = P_S \cup \{\perp\}$ and $Q_{\overline{S}} = Q_S$. Furthermore
$u \xrightarrow[\mathrm{Tr}(S)]{a} v \iff \perp u \xrightarrow[\mathrm{Tr}(\overline{S})]{a} \perp v$ for any $u, v \in P_S^*.Q_S$.
It follows that for any $L \in \mathrm{Rat}((P_S \cup Q_S)^*) \cap P_S^*.Q_S$ written by abuse of notation as $\mathrm{Rat}(P_S^*.Q_S)$,

$$\mathrm{Tr}(S)_{|L} \;=\; \mathrm{Tr}(\overline{S})_{|\perp L}$$

and for any vertex v of $\mathrm{Tr}(S)$ *i.e.* $v \in P_S^*.W_S$, the connected component (resp. accessible subgraph) of $\mathrm{Tr}(S)$ containing v (resp. from v) is the connected component (resp. accessible subgraph) of $\mathrm{Tr}(\overline{S})$ containing (resp. from) $\perp v$.

ii) Let S be any pushdown automaton.
Thus S is simulated by itself as a rewriting system over $P_S \cup Q_S$ because

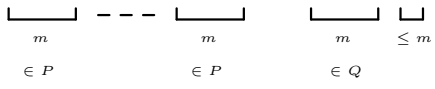
$$\mathrm{Tr}(S)_{|L} \;=\; \mathrm{Suff}(S)_{|L} \quad \text{for any } L \in \mathrm{Rat}(P_S^*.Q_S)$$

and for any $v \in P_S^*.W_S$, the connected component (resp. accessible subgraph) of $\mathrm{Tr}(S)$ containing v (resp. from v) is the connected component (resp. accessible subgraph) of $\mathrm{Suff}(S)$ containing (resp. from) v.

iii) Let S be any labelled word rewriting system.
We want to simulate S by a weak pushdown automaton $\overline{S}$.
Let m be the greatest length of the words of S:

$$m \;:=\; max\{\, |u| \mid u \in W_S \,\}.$$

As in (i), we take a new symbol $\perp$ to mark on the left the words over N_S.
Any word in $\perp N_S^*$ is decomposed from left to right into m blocks (the last block being of length $\leq m$):



by using the two bijections:

$$i \;\text{ from }\; N_S^m \cup \perp N_S^{m-1} \;\text{ to a new alphabet }\; P$$

$$\text{and} \quad j \;\text{ from }\; \{\, \perp w \mid w \in N_S^* \wedge |w| < 2m \,\} \cup \{\, w \in N_S^* \mid m < |w| \leq 2m \,\} \;\text{ to a new alphabet }\; Q,$$

and according to the injection k defined from $N_S^* \cup \perp N_S^*$ into $P^*.Q$ by

$$\begin{cases} k(\varepsilon) & := & \varepsilon \\ k(u) & := & j(u) \qquad \text{if } u \in Dom(j) \\ k(u) & := & i(w)\,k(v) \quad \text{if } u = wv \notin Dom(j) \wedge |w| = m\,. \end{cases}$$

For every $n \geq 0$, we denote by $f(n) := \lceil \frac{n}{m} \rceil$ the (minimal) number of blocs of length m necessary to contain n letters.
By applying (by suffix) any rule of S, we can add or delete at most m letters, hence

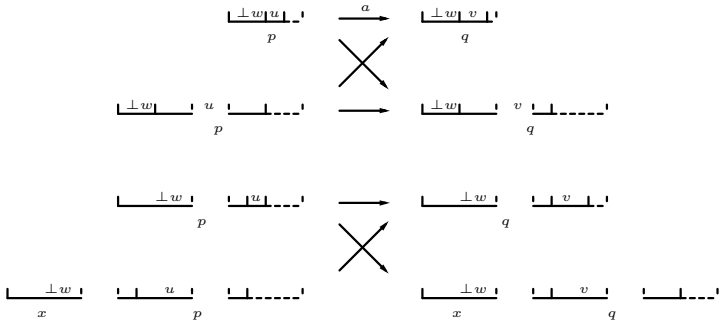
$$|\, f(|u|) - f(|v|)\,| \;\leq\; 1 \quad \text{for any } u \xrightarrow[S]{a} v\,.$$

We define the weak pushdown automaton $\overline{S} := \overline{S}' \cup \overline{S}''$ with

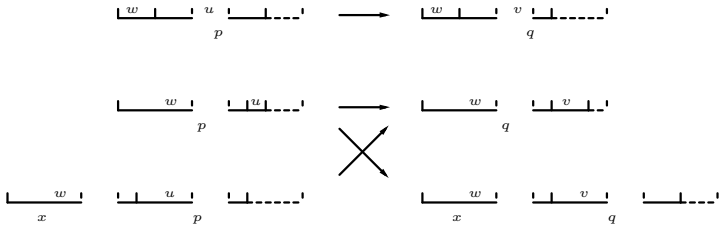
$$\overline{S}' := \{\, k(\perp wu) \xrightarrow{a} k(\perp wv) \mid u \xrightarrow[S]{a} v \wedge w \in N_S^* \wedge f(\perp wu) + f(\perp wv) \leq 5 \,\}$$

$$\overline{S}'' := \{\, k(wu) \xrightarrow{a} k(wv) \mid u \xrightarrow[S]{a} v \wedge w \in N_S^* \wedge 4 \leq f(wu) + f(wv) \leq 5 \,\}$$

We illustrate below the different types of rules for $\overline{S}'$:



We illustrate below the different types of rules for $\overline{S}''$:



Note that we have

$$u \xrightarrow[\perp N_S^*.S]{a} v \quad \Longrightarrow \quad k(u) \xrightarrow[P^*.\overline{S}]{a} k(v)$$

$$u \in \perp N_S^* \ \wedge \ k(u) \xrightarrow[P^*.\overline{S}]{a} w \quad \Longrightarrow \quad \exists\, v,\ u \xrightarrow[\perp N_S^*.S]{a} v \ \wedge \ k(v) = w$$

$$v \in \perp N_S^* \ \wedge \ w \xrightarrow[P^*.\overline{S}]{a} k(v) \quad \Longrightarrow \quad \exists\, u,\ u \xrightarrow[\perp N_S^*.S]{a} v \ \wedge \ k(u) = w\,.$$

It follows that the image by k of the connected component of $\perp N_S^*.S$ containing $\perp u$ is equal to the connected component of $P^*.\overline{S}$ containing $k(\perp u)$.
Furthermore the accessible subgraph from $\perp u$ of $\perp N_S^*.S$ is equal to the accessible subgraph from $k(\perp u)$ of $P^*.\overline{S}$.
We also deduce that the suffix graph $\mathrm{Suff}(S) = N_S^*.S$ is isomorphic to

$$k(\perp N_S^*.S) \ = \ \overline{S}' \ \cup \ i(\perp N_S^{m-1}).(i(N_S^m))^*.\overline{S}'' \ = \ (P^*.\overline{S})_{|k(\perp N_S^*)}\,,$$

hence $N_S^*.S$ is not isomorphic to $P^*.\overline{S}$ (we need a restriction).
More generally we have

$$k\big((\perp N_S^*.S)_{|\perp M}\big) \ = \ (P^*.\overline{S})_{|k(\perp M)} \quad \text{for any } \ M \subseteq N_S^*$$

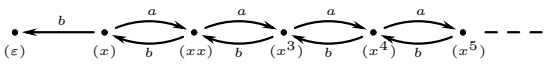
and if $M \in \mathrm{Rat}(N_S^*)$ then $k(\perp M) \in \mathrm{Rat}(P^*.Q)$.
Consequently any regular restriction of $N_S^*.S$ is isomorphic to a regular restriction of the transition graph of the weak pushdown automaton $\overline{S}$.

Q.E.D. (Theorem 5.9)

Let us illustrate the construction of the proof (iii) of Theorem 5.9 applied to the labelled word rewriting system:

$$S \;=\; \{x \xrightarrow{a} xx\,,\; x \xrightarrow{b} \varepsilon\}.$$

Its suffix graph $\mathrm{Suff}(S)$ is the following rooted graph:



Note that $L(\mathrm{Suff}(S), x, \varepsilon)$ is the Lukasiewicz language.
By applying the construction (iii) of Theorem 5.9, the greatest length of S is $m = 2$.
Its set of states is $Q \;=\; \{1\,,\,2\,,\,3\,,\,4\,,\,p\,,\,q\}$ with the following bijection j:

$$\begin{array}{rcl@{\quad;\quad}rcl@{\quad;\quad}rcl} \bot & \longmapsto & 1 & \bot x & \longmapsto & 2 & \bot xx & \longmapsto & 3 \\ \bot xxx & \longmapsto & 4 & xxx & \longmapsto & p & xxxx & \longmapsto & q \end{array}$$

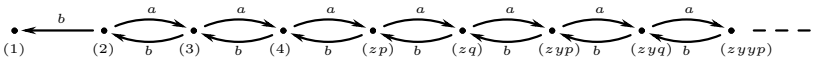
and its set of pushdown letters is $P \;=\; \{y, z\}$ with the bijection i:

$$\begin{array}{rcl@{\quad;\quad}rcl} xx & \longmapsto & y & \bot x & \longmapsto & z \end{array}$$

By coding the arcs of $\mathrm{Suff}(S)$ restricted to $\{\varepsilon\,,\ldots,\,x^5\}$, we get the following weak pushdown automaton $\overline{S}$:

$$\left|\begin{array}{rcl@{\quad;\quad}rcl@{\quad;\quad}rcl} 2 & \xrightarrow{b} & 1 & 2 & \xrightarrow{a} & 3 & 3 & \xrightarrow{b} & 2 \\ 3 & \xrightarrow{a} & 4 & 4 & \xrightarrow{b} & 3 & 4 & \xrightarrow{a} & zp \\ zp & \xrightarrow{b} & 4 & p & \xrightarrow{a} & q & q & \xrightarrow{b} & p \\ q & \xrightarrow{a} & yp & yp & \xrightarrow{b} & q \end{array}\right.$$

Its transition graph $\mathrm{Tr}(\overline{S})$ accessible from 2 (or connected to 2) is the following:



The use of weak pushdown automata, instead of word rewriting systems or of pushdown automata, allows simpler constructions. For instance, let us restrict Theorem 5.6 to weak pushdown automata.

Proposition 5.10. *Let S be a weak pushdown automaton.*
Let H be a finite deterministic graph labelled in P_S and coloured in Q_S recognizing from a vertex i the configuration language:

$$L \;=\; \{\; u\,q \mid u \in P_S^* \;\wedge\; q \in Q_S \;\wedge\; i \underset{H}{\overset{u}{\Longrightarrow}} s \;\wedge\; qs \in H \;\}.$$

Thus $\mathrm{Suff}(S)_{|L}$ is generated by length by a grammar with $|V_H| + 1$ rules.

Proof. Let $Q_S \;=\; \{q_1, \ldots, q_n\}$ be the set of states of S.
We associate to any vertex s of H a new label $[s]$ of arity n and we define the grammar R with the axiom rule

$$Z \;\longrightarrow\; \{\,[i]q_1 \ldots q_n\,\} \;\cup\; \{\; p \underset{S}{\xrightarrow{a}} q \mid pi, qi \in H \;\}\,,$$

and for any vertex s of H, we take the following rule:

$$
\begin{array}{rcl}
[s]q_1 \ldots q_n & \longrightarrow & \{\ xp \xrightarrow[S]{a} xq \mid p \xrightarrow[S]{a} q \ \wedge\ \exists\, t,\ s \xrightarrow[H]{x} t \ \wedge\ pt\,, qt \in H\ \} \\
& \cup & \{\ p \xrightarrow[S]{a} xq \mid \exists\, t,\ s \xrightarrow[H]{x} t \ \wedge\ ps\,, qt \in H\ \} \\
& \cup & \{\ xp \xrightarrow[S]{a} q \mid \exists\, t,\ s \xrightarrow[H]{x} t \ \wedge\ pt\,, qs \in H\ \} \\
& \cup & \{\ [t](xq_1)\ldots(xq_n) \mid s \xrightarrow[H]{x} t\ \}
\end{array}
$$

Thus R generates by length $(P_S^*.S)_L$ from its axiom Z. Q.E.D. (Prop. 5.10)

Taking the weak pushdown automaton of Figure 5.3 restricted to the system

$$S \ = \ \{p \xrightarrow{a} xp\,,\ p \xrightarrow{b} q\,,\ xq \xrightarrow{c} q\}$$

and the regular language $L = (xx)^*p \cup x^*q$ of configuration recognized from vertex i by the following finite deterministic automaton:

the construction of Proposition 5.10 gives the following grammar which generates $\mathrm{Suff}(S)_L$ by length.

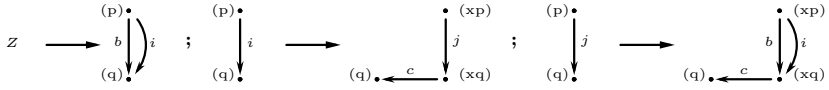


Figure 5.4. Regular restriction of a weak pushdown graph.

5.3 Main result

Finally we put together Theorem 5.8 and Theorem 5.9, and we recall Theorem 4.6 and Theorem 5.6.

Theorem 5.11. *The suffix graphs of labelled word rewriting systems,*
the transition graphs of pushdown automata,
the transition graphs of weak pushdown automata,
have up to isomorphism
the same connected components: the connected regular graphs of bounded degree,
the same accessible subgraphs: the rooted regular graphs of bounded degree,
the same regular restrictions: the regular graphs of bounded degree.
These graphs are regular by length, and also by distance when they are connected.

All these equivalences are effective. Note that by Theorem 4.6 (or Proposition 4.3), the regularity by distance for the connected graphs coincides with the finite decomposition by distance. It follows a 'famous' original result.

Theorem 5.12 ([MS 85])**.** *The connected components of pushdown automata are the connected graphs of bounded degree having a finite decomposition by distance.*

This result has been expressed with the usual pushdown automata which are intermediate devices between the general labelled word rewriting systems (applied by suffix) and the weak pushdown automata. Furthermore the finite decomposition by distance for the connected graphs of bounded degree is a normal form of the regularity.

6 Languages

Any graph G traces the language $L(G,i,f)$ of the labels of its paths from a colour i to a colour f. By Theorem 5.11, the regular graphs trace exactly the context-free languages, and by restriction to path grammars, we give directly a context-free grammar generating the path labels of any regular graph (cf. Propositions 6.2 and 6.3). Finally we verify that the deterministic regular graphs trace exactly the deterministic context-free languages (cf. Proposition 6.5).

6.1 Path grammars

The regular languages are the languages recognized by the finite automata:

$\mathrm{Rat}(T^*) := \{ L(G,i,f) \mid G \text{ finite} \wedge F_G \cap F_2 \subseteq T \wedge i,f \in F_1 \}$

and the context-free languages, which are the languages recognized by the pushdown automata, are the languages recognized by the regular graphs:

$\mathrm{Alg}(T^*) := \{ L(G,i,f) \mid G \text{ regular} \wedge F_G \cap F_2 \subseteq T \wedge i,f \in F_1 \}.$

This equality follows by Theorem 5.11 because by adding ε-transitions, we can transform any regular graph G into a regular graph $\overline{G}$ of bounded degree recognizing the same language: $L(G,i,f) = L(\overline{G},i,f)$.

Let us give a simple construction to get directly a context-free grammar generating the recognized language of a regular graph. In fact and contrary to the previous sections, we just need transformations preserving the recognized language but not the structure. First by adding ε-transitions, we can start from a unique vertex to end to a unique vertex.

Precisely, let R be a grammar and H be a finite hypergraph such that $R^\omega(H)$ are only coloured graphs. For any colours i,f, we denote

$$L(R,H,i,f) := L(\mathrm{Gen}(R,H),i,f)$$

the label set of the paths from i to f of any generated graph by R from H, or in particular for the canonical graph $\mathrm{Gen}(R,H)$ defined in 3.5.

For Z the axiom of R, we also write

$$L(R,i,f) := L(R,Z,i,f) = L(\mathrm{Gen}(R),i,f).$$

We say that R is an *initial grammar* for the colours i,f when only the right hand side H of Z is coloured by i,f, and i,f colour a unique vertex:

$$|H \cap i\,V_H| = 1 = |H \cap f\,V_H|.$$

Lemma 6.1. *For any grammar R and colours i,f, we can get an initial grammar S labelled in $F_R \cup \{\varepsilon\}$ and recognizing the same language: $L(R,i,f) = L(S,i,f)$.*

Proof. Let R be any grammar generating from its axiom Z a coloured graph G.

To any non-terminal $A \in N_R - \{Z\}$, we associate a new symbol A' of arity $\varrho(A)+2$.

We take two new vertices p,q which are not vertices of R.

We define the following grammar:

$$\begin{array}{rcl} S & := & \{\ (Z\,,\, K' \cup \{ip\,,\, fq\}) \mid (Z,K) \in R\ \} \\ & \cup & \{\ (A'Xpq\,,\, K') \mid (AX,K) \in R\ \wedge\ A \neq Z\ \} \end{array}$$

where for any hypergraph $K \in Im(S)$, the graph K' is the following:

$$\begin{array}{rcl} K' & := & \{\ s \xrightarrow[K]{a} t \mid a \in T_R\ \} \ \cup\ \{\ A'Xpq \mid AX \in K\ \wedge\ A \in N_R\ \} \\ & \cup & \{\ p \xrightarrow{\varepsilon} s \mid i\,s \in K\ \} \ \cup\ \{\ s \xrightarrow{\varepsilon} q \mid f\,s \in K\ \}\,. \end{array}$$

Assuming that $p, q \notin V_G$, S generates from its axiom Z the following graph:

$$H := (G - F_1\, V_G) \cup \{ip\,,\, fq\} \cup \{\ p \xrightarrow{\varepsilon} s \mid i\,s \in G\ \} \cup \{\ s \xrightarrow{\varepsilon} q \mid f\,s \in G\ \}$$

satisfying $L(G,i,f) = L(H,i,f)$ *i.e.* $L(R,i,f) = L(S,i,f)$.
Note that for G having an infinite number of initial (resp. final) vertices, the vertex p (resp. q) in H is of infinite out-degree (resp. in-degree). By adding new ε-arcs, we can avoid these infinite degrees. Q.E.D. (Lemma 6.1)

Let us illustrate the construction of Lemma 6.1.

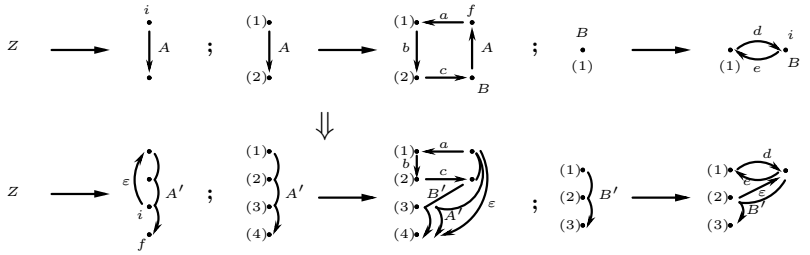


Figure 6.1. Same language from an initial vertex to a final vertex.

To preserve the recognized language of a regular graph (and not the structure), we can restrict to grammars having only arcs (of arity 2).
A *path grammar* R is a deterministic graph grammar without axiom and whose each rule is of the form $A12 \longrightarrow H$ where H is a finite set of arcs having no arc of source 2 and no arc of goal 1.
We give below a path grammar which is *acyclic*: each right hand side is an acyclic graph.

Figure 6.2. An acyclic path grammar.

For any path grammar R, any $A \in N_R$ and any derivation
$$A12 = H_0 \underset{R}{\Longrightarrow} H_1 \ldots H_n \underset{R}{\Longrightarrow} \ldots$$
we define the following languages:
$$\begin{array}{rcll} L_n(R,A) & := & L(H_n,1,2) & \subseteq \quad (N_R \cup T_R)^* \quad \forall\ n \geq 0 \\ L(R,A) & := & \bigcup_{n\geq 0}\big(L_n(R,A) \cap T_R^*\big) & \subseteq \quad T_R^*\ . \end{array}$$

Proposition 6.2. [CD 07] *For any grammar R and colours i, f, we can get a path grammar S recognizing from a non-terminal A the language $L(S,A) = L(R,i,f)$.*

Proof. **i)** We assume that each rule of R is of the form: $A1\ldots\varrho(A) \longrightarrow H_A$ for any $A \in N_R$.
Let Z be the axiom of R. By Lemma 6.1, we suppose that R is initial: i (resp. f) colours only H_Z (not the other right hand sides of R) and on a unique vertex $\overline{p}$ (resp. $\overline{q} \neq \overline{p}$).
We assume that 0 is not a vertex of R and we take a new set of labels of arity 2:
$$\{\ A_{i,j} \mid A \in N_R\ \wedge\ 1 \leq i,j \leq \varrho(A)\ \}\ \cup\ \{Z'\}\,.$$
We define the *splitting* $\prec G \succ$ of any $(T_R \cup N_R)$-hypergraph G as being the graph:
$$\begin{array}{rcl} \prec G \succ & := & \{\ s \xrightarrow{a} t \mid ast \in G\ \wedge\ a \in T_R\ \} \\ & \cup & \{\ s \xrightarrow{A_{i,j}} t \mid A \in N_R\ \wedge\ 1 \leq i,j \leq \varrho(A)\ \wedge \\ & & \qquad \exists\ s_1,\ldots,s_{\varrho(A)},\ As_1\ldots s_{\varrho(A)} \in G\ \wedge\ s = s_i\ \wedge\ t = s_j\ \} \end{array}$$
and for $p, q \in V_G$ and $P \subseteq V_G$ with $0 \notin V_G$, we define for $p \neq q$
$$\begin{array}{rcl} G_{p,q,P} & := & \{\ s \xrightarrow[\prec G \succ]{a} t \mid t \neq p\ \wedge\ s \neq q\ \wedge\ s,t \notin P\ \}_{|\{\ s \mid p \longrightarrow^* s \longrightarrow^* q\}} \\ G_{p,p,P} & := & \big(\{\ s \xrightarrow[\prec G \succ]{a} t \mid t \neq p\ \wedge\ s,t \notin P\ \} \\ & \cup & \{\ s \xrightarrow{a} 0 \mid s \xrightarrow[\prec G \succ]{a} p\ \}\big)_{|\{\ s \mid p \longrightarrow^* s \longrightarrow^* 0\}}\,. \end{array}$$
This allows to define the splitting of R as being the following path grammar:
$$\begin{array}{rcl} & \prec R \succ & \\ := & \{\ A_{i,j}12 \longrightarrow h_{i,j}\big((H_A)_{i,j,[\varrho(A)]-\{i,j\}}\big) \mid A \in N_R\ \wedge\ 1 \leq i,j \leq \varrho(A)\ \} & \\ \cup & \{\ Z'12 \longrightarrow \big(h(\prec H_Z \succ)\big)_{|\{\ s \mid 1 \longrightarrow^* s \longrightarrow^* 2\}}\ \} & \end{array}$$
where $h_{i,j}$ is the vertex renaming of $(H_A)_{i,j,[\varrho(A)]-\{i,j\}}$ defined by
$$h_{i,j}(i) = 1\,,\ h_{i,j}(j) = 2\,,\ h_{i,j}(x) = x \ \text{ otherwise, for } i \neq j$$
$$h_{i,i}(i) = 1\,,\ h_{i,i}(0) = 2\,,\ h_{i,i}(x) = x \ \text{ otherwise,}$$
and h is the vertex renaming of H_Z defined by

$h(\overline{p}) = 1\,,\ h(\overline{q}) = 2\,,\ h(x) = x$ otherwise.

We then put $\prec R \succ$ into a reduced form.

ii) Let us show that $L(R,i,f) = L(\prec R \succ, Z')$.

For any $A \in N_R$, we take a derivation

$$A1 \ldots \varrho(A) = H_0 \underset{R}{\Longrightarrow} H_1 \underset{R}{\Longrightarrow} \ldots H_n \underset{R}{\Longrightarrow} \ldots$$

we write $H_\omega = \bigcup_{n \geq 0} [H_n]$ and for every $1 \leq i,j \leq \varrho(A)$ and $0 \leq n \leq \omega$, we define the following languages:

$$\begin{array}{rcl} L_n(R,A,i,j) & := & L\big((H_n)_{i,j,[\varrho(A)]-\{i,j\}}, i, j\big) \quad \text{for } i \neq j \\ L_n(R,A,i,i) & := & L\big((H_n)_{i,i,[\varrho(A)]-\{i\}}, i, 0\big) \quad . \end{array}$$

Note that for any $A \in N_R$, any $i,j \in [\varrho(A)]$ and $n \geq 0$, we have

$$\begin{array}{rcl} L_n(R,A,i,j) & \subseteq & \big(T_R \cup \{\ A_{p,q} \mid A \in N_R \ \wedge\ p,q \in [\varrho(A)]\ \}\big)^* \\ \text{and} \quad L_\omega(R,A,i,j) & \subseteq & T_R^* \,. \end{array}$$

Let us verify that for any $A \in N_R$ and $1 \leq i,j \leq \varrho(A)$, we have

$$L_\omega(R,A,i,j) = L(\prec R \succ, A_{i,j})\,.$$

As $L_\omega(R,A,i,j) = \bigcup_{n \geq 0} \big(L_n(R,A,i,j) \cap T_R^*\big)$, it is sufficient to prove by induction on $n \geq 0$ that

$$L_n(R,A,i,j) = L_n(\prec R \succ, A_{i,j})\,.$$

$n = 0$: we have $L_0(R,A,i,j) = \{A_{i,j}\} = L_0(\prec R \succ, A_{i,j})$.

$n = 1$: we have $L_1(R,A,i,j) = L((H_A)_{i,j,[\varrho(A)]-\{i,j\}}, i, j) = L_1(\prec R \succ, A_{i,j})$.

$n \implies n+1$:

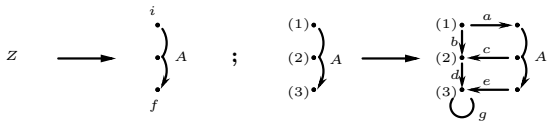
$$\begin{array}{rcl} L_{n+1}(R,A,i,j) & = & L_1(R,A,i,j)\,[L_n(R,B,p,q)/B_{p,q}] \\ & = & L_1(\prec R \succ, A_{i,j})\,[L_n(\prec R \succ, B_{p,q})/B_{p,q}] \ \text{ by ind. hyp.} \\ & = & L_{n+1}(\prec R \succ, A_{i,j})\,. \end{array}$$

Finally we have

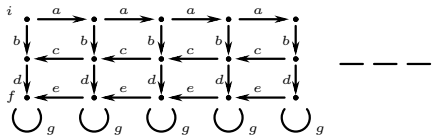
$$\begin{array}{rcl} L(R,i,f) & = & L(\mathrm{Gen}(R), i, f) \\ & = & L(\prec H_Z \succ, \overline{p}, \overline{q})\,[L_\omega(R,A,i,j)/A_{i,j}] \\ & = & L_1(\prec R \succ, Z')\,[L(\prec R \succ, A_{i,j})/A_{i,j}] \\ & = & L(\prec R \succ, Z')\,. \end{array}$$

Q.E.D. (Proposition 6.2)

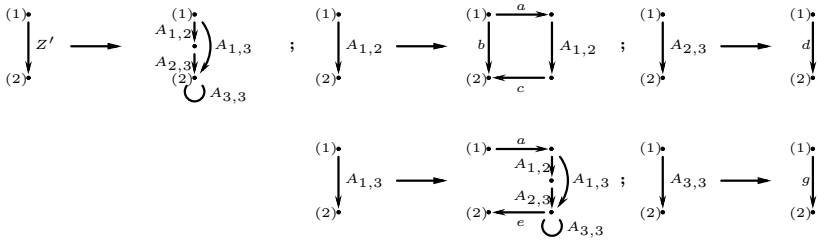
Let us illustrate Proposition 6.2 starting from the following grammar R:



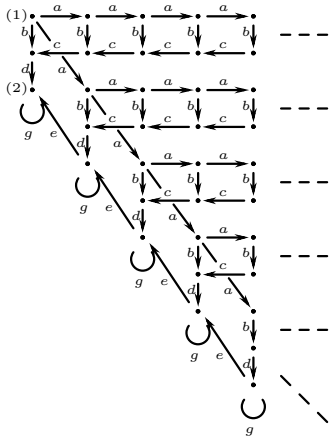
Its generated graph $R^{\omega}(Z)$ is given below.



Proposition 6.2 splits grammar R into the following grammar $\prec R \succ$ in reduced form:



We get the following generated graph $\prec R \succ^{\omega}(Z'12)$:



Obviously the graphs $R^{\omega}(Z)$ and $\prec R \succ^{\omega}(Z'12)$ are not isomorphic but by Proposition 6.2 they recognize the same language:

$$L(R^{\omega}(Z)\,,\,i\,,\,f) \;=\; L(\prec R \succ, Z') \;=\; \{\; a^{m+n}bc^{m}d(eg^{*})^{n} \mid m,n \geq 0 \;\}.$$

We now show that path grammars are language-equivalent to context-free

grammars (on words).
Recall that a *context-free grammar* P is a finite binary relation on words in which each left hand side is a letter called a non-terminal, and the remaining letters of P are terminals. By denoting N_P and T_P the respectives sets of non-terminals and terminals of P, the rewriting $\underset{P}{\longrightarrow}$ according to P is the binary relation on $(N_P \cup T_P)^*$ defined by

$$UAV \underset{P}{\longrightarrow} UWV \quad \text{if} \quad (A,W) \in P \quad \text{and} \quad U,V \in (N_P \cup T_P)^*.$$

The derivation $\underset{P}{\longrightarrow}^*$ is the reflexive and transitive closure of $\underset{P}{\longrightarrow}$ with respect to composition. The language $L(P,U)$ generated by P from any $U \in (N_P \cup T_P)^*$ is the set of terminal words deriving from U:

$$L(P,U) \ := \ \{\ u \in T_P^* \mid U \underset{P}{\longrightarrow}^* u\ \}\,.$$

Path grammars and context-free grammars are language-equivalent with linear time translations.

Proposition 6.3 ([CD 07])**.**
a) *We can transform in linear time any path grammar R into a context-free grammar $\widehat{R}$ such that $L(R,A) = L(\widehat{R},A)$ for any $A \in N_R$.*

b) *We can transform in linear time any context-free grammar P into an acyclic path grammar $\vec{P}$ such that $L(P,A) = L(\vec{P},A)$ for any $A \in N_P$.*

Proof. **i)** The first transformation is analogous to the translation of any finite automaton into an equivalent right linear grammar.
To each non-terminal $A \in N_R$, we take a vertex renaming h_A of $R(A12)$ such that $h_A(1) = A$ and $h_A(2) = \varepsilon$, and the image $Im(h_A) - \{\varepsilon\}$ is a set of symbols with $Im(h_A) \cap Im(h_B) = \{\varepsilon\}$ for any $B \in N_R - \{A\}$.
We define the following context-free grammar:

$$\widehat{R} \ := \ \{\ (h_A(s), a h_A(t)) \mid \exists\ A \in N_R\,,\ \ s \underset{R(A12)}{\overset{a}{\longrightarrow}} t\ \}.$$

Note that each right side of $\widehat{R}$ is a word of length at most 2, and the number of non-terminals of $\widehat{R}$ depends on the description length of R:

$$|N_{\widehat{R}}| \ = \ \left(\textstyle\sum_{A \in N_R} |V_{R(A12)}|\right) \ - \ |N_R|\,.$$

For instance, the path grammar of Figure 6.2 is transformed into the following context-free grammar:

$$\begin{array}{lllllllllll}
A & = & aC & ; & C & = & BD & ; & D & = & c \\
B & = & aF + bE & ; & E & = & aG + d & ; & F & = & AG \\
G & = & AH & ; & H & = & c
\end{array}$$

ii) For the second transformation, we have $N_{\vec{P}} = N_P$ and for each $A \in N_P$, its right hand side in $\vec{P}$ is the set of distinct paths from 1 to 2 labelled by the right hand sides of A in P.
We translate the context-free grammar P into the following acyclic path

grammar:

$$\vec{P} \;:=\; \{\; (A12\,,\, H_A) \mid A \in Dom(P) \;\}$$

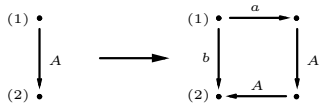
such that for each non-terminal $A \in Dom(P)$, the graph H_A is the set of right hand sides of A in P starting from 1 and ending to 2 :

$$\begin{aligned}
H_A \;:=\;& \{\; 1 \xrightarrow{B} (B,V) \mid (A,BV) \in P \;\wedge\; |B| = 1 \;\wedge\; V \neq \varepsilon \;\} \\
\cup\;& \{\; (U,BV) \xrightarrow{B} (UB,V) \mid (A,UBV) \in P \;\wedge\; |B| = 1 \;\wedge\; U,V \neq \varepsilon \;\} \\
\cup\;& \{\; (U,B) \xrightarrow{B} 2 \mid (A,UB) \in P \;\wedge\; |B| = 1 \;\wedge\; U \neq \varepsilon \;\} \\
\cup\;& \{\; 1 \xrightarrow{B} 2 \mid (A,B) \in P \;\wedge\; |B| = 1 \;\} \\
\cup\;& \{\; 1 \xrightarrow{\varepsilon} 2 \mid (A,\varepsilon) \in P \;\}\,.
\end{aligned}$$

Note that $N_P = N_{\vec{P}}$ and $T_P = T_{\vec{P}} - \{\varepsilon\}$.
For instance the context-free grammar $\{(A, aAA)\,,\,(A,b)\}$ generating from A the Lukasiewicz language, is translated into the acyclic path grammar reduced to the unique rule:

$$A12 \;\longrightarrow\; \{1 \xrightarrow{b} 2\,,\, 1 \xrightarrow{a} (a,AA)\;,\; (a,AA) \xrightarrow{A} (aA,A)\;,\; (aA,A) \xrightarrow{A} 2\}$$

and represented below:



Q.E.D. (Proposition 6.3)

Note that by using the two transformations of Proposition 6.3, we can transform in linear time any path grammar into a language equivalent acyclic path grammar.
By Proposition 6.2 and Proposition 6.3 (a), the recognized languages of regular graphs are generated by context-free grammars. The converse is true by Proposition 6.3 (b).

Corollary 6.4. *The regular graphs recognize exactly the context-free languages.*

6.2 Deterministic languages

We now focus on the deterministic regular graphs.
We say that a coloured graph G is *deterministic from* a colour i if i colours a unique vertex of G, and two arcs with the same source have distinct labels:

$$|G \cap i\,V_G| = 1 \quad \text{and} \quad (p \xrightarrow[G]{a} q \;\wedge\; p \xrightarrow[G]{a} r \;\;\Longrightarrow\;\; q = r).$$

The languages recognized by the deterministic regular graphs:

$$\mathrm{DAlg}(T^*) \;:=\; \{\; L(G,i,f) \mid G \text{ regular and deterministic from } i, \;\wedge\; F_G \cap F_2 \subseteq T \;\wedge\; i,f \in F_1 \;\}$$

are the languages recognized by the deterministic pushdown automata.

Proposition 6.5. *The deterministic regular graphs recognize exactly the deterministic context-free languages.*

Proof. Recall that a *deterministic pushdown automaton* S over an alphabet T of terminals is a finite subset of $PQ\times(T\cup\{\varepsilon\})\times P^*Q$ where P,Q are disjoint alphabets of respectively stack letters and states, and such that S is deterministic for any $a\in T\cup\{\varepsilon\}$:

$$(xp \xrightarrow{a} uq \;\wedge\; xp \xrightarrow{a} vr) \;\Longrightarrow\; uq = vr$$

and each left-hand side of an ε-rule is not the left-hand side of a terminal rule:

$$(xp \xrightarrow{\varepsilon} uq \;\wedge\; xp \xrightarrow{a} vr) \;\Longrightarrow\; a = \varepsilon\,.$$

The language $L(\mathrm{Tr}(S),xp,F)$ recognized by S starting from an initial configuration xp and ending to a regular set $F\subseteq P^*Q$ of final configurations, is a *deterministic context-free language*.

i) Let us verify that $L(\mathrm{Tr}(S),xp,F)$ is traced by a deterministic regular graph. We take two colours i and f.
By Proposition 5.4 or more precisely by Corollary 5.5, the following coloured graph:

$$G \;:=\; \mathrm{Tr}(S) \;\cup\; \{i\,(xp)\} \;\cup\; \{\; f\,u \mid u\in F \;\}$$

is a regular graph.
Let R be a grammar generating G. We define the following grammar:

$$R' \;:=\; \{\; (X',H') \mid (X,H)\in R \;\}$$

where for any hyperarc $fs_1\ldots s_n$ of R, we associate the hyperarc

$$(fs_1\ldots s_n)' \;:=\; f[s_1]\ldots[s_n]$$

that we extend by union to any right hand side H of R:

$$H' := \{\; Y' \mid Y\in H \;\}$$

and such that for any vertex $s\in V_H$,

$$[s] \;:=\; \{\; t \mid t \xrightarrow[[H]]{\varepsilon}{}^{*}\; s \;\vee\; s \xrightarrow[[H]]{\varepsilon}{}^{*}\; t \;\}.$$

Thus R' is without ε-arc and

$$L(R',i,f) \;=\; L(R,i,f) \;=\; L(G,i,f) \;=\; L(\mathrm{Tr}(S),xp,F).$$

ii) Let i,f be colours and R be a grammar such that $\mathrm{Gen}(R)$ is deterministic from i.
We want to show that $L(R,i,f)$ is a deterministic context-free language.
By Proposition 4.4 (and 4.1), we assume that $\mathrm{Gen}(R)$ is accessible from i.
By Lemma 3.11, we can assume that R is terminal outside.
For any rule $(X,H)\in R$, we define

$$Out(X(1)) \;:=\; \{\; i \mid 1<i\le\varrho(X(1)) \;\wedge\; \exists\, a,\; X(i) \xrightarrow[\mathrm{Gen}(R,\,X)]{a} \;\}$$

the ranks of the input vertices which are source of an arc in the generated

graph from X. Precisely $\big(Out(A)\big)_{A\in N_R}$ is the least fixed point of the system: for each $(X,H)\in R$,

$$\begin{aligned} Out(X(1)) &= \{\, i \mid \exists\, a,\ X(i)) \xrightarrow[[H]]{a} \,\} \\ &\cup \{\, i \mid \exists\, Y\in H\cap N_R V_H^*\ \exists\, j\in Out(Y(1)),\ X(i)=Y(j)\,\}. \end{aligned}$$

We rewrite non-terminal hyperarcs in the right hand sides of R until all the terminal arcs of input source are produced. We begin with the grammar:

$$R_0 := R$$

and having constructed a grammar R_n for $n\geq 0$, we choose a rule $(X,H)\in R_n$ and a non-terminal hyperarc $Y\in H\cap N_R V_H^*$ such that

$$V_X\cap\{\, Y(i)\mid i\in Out(Y(1))\,\}\neq\varnothing$$

and we rewrite Y in H to get a hypergraph K *i.e.* $H \underset{R_n,\,Y}{\longrightarrow} K$ in order to replace H by K in R_n:

$$R_{n+1} := (R_n - \{(X,H)\})\cup\{(X,K)\}.$$

If such a choice is not possible, we stop with $\overline{R}:=R_n$.
As $\mathrm{Gen}(R)$ is deterministic, it is of bounded out-degree, hence $\overline{R}$ exists.
By construction, $\overline{R}$ is equivalent to R:
$R^\omega(X) = \overline{R}^\omega(X)$ for any $X\in Dom(R)=Dom(\overline{R})$.
Furthermore $\overline{R}$ satisfies the following property:

$$\forall\,(X,H)\in\overline{R}\ \ \forall\, Y\in H\cap N_{\overline{R}}V_H^*$$
$$V_X\cap\{\, Y(i)\mid i\in Out(Y(1))\,\}=\varnothing$$

meaning that any input which is a vertex of a non-terminal hyperarc Y cannot be a source of an arc in the generated graph from Y.
For each rule $(X,H)\in\overline{R}$, we denote

$$InOut(X(1)) := \bigcup\{\, V_X\cap V_Y \mid Y\in H\ \wedge\ Y(1)\in N_{\overline{R}}\,\}$$

the set of input-output vertices; and for each $s\in InOut(X(1))$, we take a new vertex $s'\notin V_H$ and to any non-terminal hyperarc $Y\in H$ with $Y(1)\in N_{\overline{R}}$, we associate the hyperarc $Y' = Y(1)Y(2)'\ldots Y(|Y|)'$ with $s':=s$ for any $s\in V_H - InOut(X(1))$.
We define the grammar R' by associating to each rule $(X,H)\in\overline{R}$, the following rule:

$$X \longrightarrow [H]\cup\{\, Y'\mid Y\in H\wedge Y(1)\in N_{\overline{R}}\,\}\cup\{\, s'\xrightarrow{\varepsilon} s\mid s\in InOut(X(1))\,\}$$

Thus $L(R,i,f) = L(R',i,f)$ and the graph $\mathrm{Gen}(R')$ is of finite degree, deterministic over $T_R\cup\{\varepsilon\}$ and such that any source of an ε-arc is not source of an arc labelled in T_R.
By Theorem 5.11, $\mathrm{Gen}(R')$ is the transition graph of a pushdown automaton S accessible from an initial configuration c_0 with a regular set F of final configurations:

$$\mathrm{Gen}(R') = \mathrm{Tr}(S)_{\{\ c\ \mid\ c_0\ \longrightarrow^*\ c\ \}}\cup\{i\,c_0\}\cup\{\, f\,c\mid c\in F\,\}.$$

Finally S is a deterministic pushdown automaton recognizing the language:

$$L(Tr(S),i,F) = L(R',i,f) = L(R,i,f).$$ Q.E.D. (Proposition 6.5)

Due to a lack of space (and time), we have only presented a first (and partial) survey on deterministic graph grammars. After defining suitable normal forms, we explored the notion of regularity of a graph with respect to a finite-index graduation of its vertices. Together with a generic representation of grammar-generated graphs, this yields a canonical representation of any given regular graph. These definitions and techniques constitute a basic toolkit for conveniently manipulating deterministic graph grammars. As an illustration, we were able to prove in a self-contained way several known structural results concerning regular graphs, the most important being their links with the transition graphs of pushdown automata.
This is only a first step in studying deterministic graph grammars, and many interesting developments remain to be explored. We hope that this paper might encourage further work on the subject. In particular, we believe that grammars will prove an invaluable tool in extending finite graph theory to the class of regular graphs, as well as finite automata theory to some sub-families of context-free languages. Some efforts in these directions have already begun to appear ([Ca 06, CD 07]). Other leads for further research concern the use of grammars as a tool for more general computations (a particular case is Proposition 4.4), and the design of geometrical proofs for results related to context-free languages (e.g. the standard pumping lemma).
Let us conclude with a natural question: how can one extend deterministic graph grammars in order to generate the structure of infinite automata [Th 01], in particular those associated to pushdown automata using stack of stacks [Th 03],[Car 06] ?

Let me thank Arnaud Carayol and Antoine Meyer for their help in drafting this paper.

Many thanks to Wolfgang Thomas for his support, and happy birthday.

References

[Be 69] M. Benois *Parties rationnelles du groupe libre*, C.R. Académie Sci. Paris, Sér. A 269, 1188–1190 (1969).

[Bü 64] R. Büchi *Regular canonical systems*, Archiv für Mathematische Logik und Grundlagenforschung 6, pp. 91–111 (1964);
or in *The collected works of J. Richard Büchi*, edited by S. Mac Lane and D. Siefkes, Springer-Verlag, New York, pp. 317–337 (1990).

[Car 06] A. Carayol *Automates infinis, logiques et langages*, PhD Thesis, IFSIC, University of Rennes 1 (2006).

[Ca 90] D. Caucal *On the regular structure of prefix rewriting*, 15^{th} CAAP, LNCS 431, A. Arnold (Ed.), 87–102 (1990);
or in Theoretical Computer Science 106, 61–86 (1992).

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

[CD 07] D. Caucal and T.H. Dinh *Path algorithms on regular graphs*, 16^{th} FCT, LNCS 4639, E. Csuhaj-Varjú, Z. Ésik (Eds.), 199–212 (2007).

[Co 89] B. Courcelle *Infinite graphs of bounded width*, Mathematical Systems Theory 21-4, 187–221 (1989).

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

[Ro 97] G. Rozenberg *Handbook of graph grammars and computing by graph transformation*, World Scientific Publishing Company (1997).

[Th 01] W. Thomas *A short introduction to infinite automata*, 5^{th} DLT, LNCS 2295, W. Kuich, G. Rozenberg, A. Salomaa (Eds.), 130–144 (2001).

[Th 03] W. Thomas *Constructing infinite graphs with a decidable monadic theory*, 28^{th} MFCS, LNCS 2747, B. Rovan, P. Vojtáš (Eds.), 113–124 (2003).