The equivalence problem for t-turn DPDA is co-NP.

Géraud Sénizergues

technical report nr 1297-03
LaBRI and Université de Bordeaux I **

Abstract. We introduce new tools allowing to deal with the equality-problem for prefix-free languages. We illustrate our ideas by showing that, for every fixed integer $t \geq 1$, the equivalence problem for t-turn deterministic pushdown automata is co-NP. This complexity result refines those of [Val74,Bee76].

Keywords:deterministic pushdown automata;equivalence problem; complexity; matrix semi-groups.

1 Introduction

The so-called "equivalence problem for deterministic pushdown automata" (denoted by $\mathrm{Eq}(D_0, D_0)$ for short) is the following decision problem:

INSTANCE: two dpda A, B
QUESTION: L(A) = L(B)?

i.e. do the given automata recognize the *same* language ?

This problem was shown to be decidable in ([Sén97],[Sén01, sections 1-9]). This decidability result has been generalised in ([Sén01, section 11],[Sén98],[Sén99]) and some simplifications of the method presented in [Sén97] have been found in [Sti99,Sti01]. Nevertheless, the intrinsic complexity of this problem is far for beeing understood. A first progress in this direction has been achieved in [Sti02] by showing that $\mathrm{Eq}(D_0, D_0)$ is primitive recursive. We present here some new tools allowing to tackle this question.

Let us recall that, following (and generalising) the point of view of [HHY79], we represent the computations of a dpda, via the notion of strict-deterministic grammar $G =< X, V, P >$, as a right-action of X^* over a subset of matrices of

** mailing adress:LaBRI and UFR Math-info, Université Bordeaux1
351 Cours de la libération -33405- Talence Cedex.
email:ges@labri.u-bordeaux.fr; fax: 05-56-84-66-69;
URL:http://dept-info.labri.u-bordeaux.fr/∼ges/

"polynomials" over the set of variables V. Every equation, generated from an initial equation $v_1 \equiv v_2$, can be put in the form:

$$\alpha \cdot A_1 A_2 \cdots A_\lambda \cdot S \equiv \beta \cdot A_1 A_2 \cdots A_\lambda \cdot S, \tag{1}$$

where the A_i are square matrices , α, β (resp. S) are row-vectors (resp. a column-vector).

1.1 Tools

Our main new tool is lemma 32 which states a property of this algebra of matrices: if λ is the dimension of the matrices A_i and all the equations obtained by removing some of the matrices (the same on both sides) are valid, then the equation (1) must be valid too. A second ingredient allowing to cut down the complexity of comparison algorithms is the following observation: suppose that equation (1) occured after λ successive "stacking" derivations. Then, the smaller equations corresponding to the $2^\lambda - 1$ strict subwords of the word $A_1 A_2 \ldots A_\lambda$ must occur, if not on the same branch, in the same comparison-tree. This observation would be enough if the tree was built with derivations only (the so-called T_A transformations of Hopcroft-Korenjac). In general the situation is more complicated, but we introduce a notion of deduction relation (section 4) which can be seen as an "extended semi-ring congruence closure". If an equation corresponding to a subword $A_{i_1} A_{i_2} \cdots A_{i_p}$ is missing, this equation is nevertheless in the congruence closure of the tree, which is enough to imply (1).

1.2 Result

We choosed to illustrate these new tools on a class of automata where it is possible to construct a finite comparison-tree by a straightforward strategy. We hope this will ease the understanding of the general ideas.
We thus obtain a polynomial upper-bound on the divergence of two non-equivalent t-turn dpda (theorem 55). It follows that the equivalence problem for t-turn dpda is in co-NP (corollary 56), while it was only known to be in DTIME($2^{2^{c_1 \cdot n}}$) by [Bee76] (decidability was established in [Val74]).

1.3 Contents

We recall in section 2 some basic definitions concerning automata, grammars, monoids and semi-rings.
We recall in section 3 the notion of *deterministic* matrices and their basic algebraic properties. We introduce in section 4 our deduction relation.
In section 5, we show the main result: the divergence of two finite-turn dpda A, B is upper-bounded by a polynomial function of the size of (A, B).
In section 6 we present some extensions of the main result; we give comparisons with related works and finally, we present perspectives for future developments.

Table of Contents

2 Preliminaries

2.1 Grammars

Let us recall [Har78, definition 11.4.1].

Definition 21 *Let $G =< X, V, P >$ be a context-free grammar. G is said* strict-deterministic *iff there exists an equivalence relation $\smile$ over $X \cup V$ fulfilling the following conditions:*
1- X is a class $\pmod{\smile}$
2-for every $v, v' \in V, \alpha, \beta, \beta' \in (X \cup V)^*$, if $v \longrightarrow_P \alpha \cdot \beta$ and $v' \longrightarrow_P \alpha \cdot \beta'$ and $v \smile v'$, then either:

2.1- both $\beta, \beta' \neq \epsilon$ *and* $\beta[1] \smile \beta'[1] \pmod{\smile}$
2.2- or $\beta = \beta' = \epsilon$ *and* $v = v'$.

(In the above definition, $\gamma[1]$ denotes the first letter of the word γ). Any equivalence $\smile$ satisfying the above condition is said to be a *strict equivalence* for the grammar G. The grammar G is said *normalised* iff, in addition, every rule $(v, \gamma) \in P$ is such that $\gamma \in X \cup X \cdot V \cup X \cdot V \cdot V$. In what follows, we consider only normalised grammars. It is well-known that every strict-deterministic grammar can be reduced in such a normalised form, in polynomial time.

2.2 Right-actions

A language over an alphabet W is a subset of W^*. The study of languages over W naturally leads to considering the semi-ring $(\mathcal{P}(W^*), \cup, \cdot, \emptyset, \epsilon)$. In the context of grammars, one often uses the notation $+$ for the union operation and names "polynomials" the sum of all the right-hand sides of a given variable. Therefore, we choose to use the framework of formal power series: $(\mathbb{B}\langle\langle\ W\ \rangle\rangle, +, \cdot, 0, 1)$ denotes the semi-ring of *boolean series* over W, which is isomorphic to the semi-ring $(\mathcal{P}(W^*), \cup, \cdot, \emptyset, \epsilon)$; similarly $\mathbb{B}\langle\ W\ \rangle$ denotes the sub-semi-ring of polynomials over the undeterminates W.
This framework also emphasizes the fact that, most of our arguments can be adapted to coefficients in some semi-rings K other than the semi-ring $\mathbb{B}$ (see [Sén01, section 11] for example). The reader is referred to [Sén02, section 2.3] for more details on the right-actions of a monoid over a semi-ring.

Residual action We recall the following classical σ-right-action $\bullet$ of the monoid W^* over the semi-ring $\mathbb{B}\langle\langle\ W\ \rangle\rangle$: for all $S, S' \in \mathbb{B}\langle\langle\ W\ \rangle\rangle, u \in W^*$

$$S \bullet u = S' \Leftrightarrow \forall w \in W^*, (S'_w = S_{u \cdot w}),$$

(i.e. $S \bullet u$ is the *left-quotient* of S by u , or the *residual* of S by u).

Grammatical action Let $(V, \smile)$ be the structured alphabet associated with some normalised strict-deterministic grammar $G =< X, V, P >$. We define the right-action $\odot$ as the unique σ-right-action of the monoid X^* over the semi-ring $\mathbb{B}\langle\langle\ V\ \rangle\rangle$ such that: for every $v \in V, \beta \in V^*, x \in X$

$$(v \cdot \beta) \odot x = (\sum_{(v,h) \in P} h \bullet x) \cdot \beta, \tag{2}$$

$$\epsilon \odot x = \emptyset. \tag{3}$$

2.3 Equivalence

Let us consider the unique substitution $\varphi : \mathbb{B}\langle\langle\ V\ \rangle\rangle\ \to \mathbb{B}\langle\langle\ X\ \rangle\rangle$ fulfilling: for every $v \in V$,

$$\varphi(v) = \{u \in X^* \mid v \xrightarrow{*}_P u\},$$

(in other words, φ maps every subset $L \subseteq V^*$ on the language generated by the grammar G from the set of axioms L).
We denote by $\equiv$ the kernel of φ i.e.: for every $S, T \in \mathbb{B}\langle\langle\ W\ \rangle\rangle$,

$$S \equiv T \Leftrightarrow \varphi(S) = \varphi(T).$$

For every integer n, we introduce the relations $=_n$ over $\mathbb{B}\langle\langle\ X\ \rangle\rangle$ and $\equiv_n$ over $\mathbb{B}\langle\langle\ V\ \rangle\rangle$ defined by:

$$U =_n U' \Leftrightarrow U \cap X^{\leq n} = U' \cap X^{\leq n};\ \ S \equiv_n S' \Leftrightarrow \varphi(S) =_n \varphi(S').$$

The equivalence relation $\equiv$ (resp. $=_n, \equiv_n$) is extended, componentwise, to matrices (see §2.4).

2.4 Matrices

Let us call a structured alphabet any pair $(W, \smile)$ such that $\smile$ is an equivalence relation over W.
The equivalence relation $\smile$ is extended to W^* by: for every $w_1, w_2 \in W^*$, $w_1 \smile w_2$ iff either $w_1 = w_2$ or there exists $w \in W^*, v_1, v_2 \in W, w_1', w_2' \in W^*$ such that

$$w_1 = wv_1w_1', w_2 = wv_2w_2', v_1 \neq v_2, v_1 \smile v_2.$$

Let us denote by $\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$ the set of (n, m)-matrices with entries in the semi-ring $\mathbb{B}\langle\langle\ W\ \rangle\rangle$.

Definition 22 *Let* $S \in \mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$. S *is said* deterministic *iff, for every* $i \in [1, n], j, k \in [1, m]$, $w, w' \in W^*$,
1- $w \in S_{i,j}, w \in S_{i,k} \Rightarrow j = k$
2- $w \in S_{i,j}, w' \in S_{i,k} \Rightarrow w \smile w'$

This notion is an adaptation of the notion of *set of associates* defined in [HHY79, definition 3.2 p. 188].
The set of deterministic matrices in $\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$ is denoted by $\mathsf{D}\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$.
The right-actions $\bullet$ and $\odot$ are extended componentwise to matrices.

Lemma 23 *Let* $S \in \mathsf{D}\mathbb{B}_{1,m}\langle\langle\ W\ \rangle\rangle, T \in \mathbb{B}_{m,s}\langle\langle\ W\ \rangle\rangle, u \in W^*$. *Exactly one of the following cases is true:*
1- $\exists j, S_j \bullet u \notin \{\emptyset, \epsilon\}$
in this case $(S \cdot T) \bullet u = (S \bullet u) \cdot T$.
2- $\exists j_0, \exists u', u'', u = u' \cdot u'', S_{j_0} \bullet u' = \epsilon$;
in this case $(S \cdot T) \bullet u = T_{j_0} \bullet u''$.
3- $\forall j, \forall u' \preceq u, S_j \bullet u = \emptyset, S_j \bullet u' \neq \epsilon$;
in this case $(S \cdot T) \bullet u = \emptyset = (S \bullet u) \cdot T$.

In the case where $(W, \smile)$ is the structured alphabet $(V, \smile)$ defined by some strict-deterministic grammar, the above lemma is also fulfilled by the right-action $\odot$ (instead of $\bullet$) and for $u \in X^*$.

Lemma 24 *For every* $S \in \mathsf{D}\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle, T \in \mathsf{D}\mathbb{B}_{m,s}\langle\langle\ W\ \rangle\rangle, u \in W^*$,
1- $S \cdot T \in \mathsf{D}\mathbb{B}_{n,s}\langle\langle\ W\ \rangle\rangle$.
2- $S \bullet u \in \mathsf{D}\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$.

In the case where $(W, \smile)$ is the structured alphabet $(V, \smile)$ defined by some strict-deterministic grammar, it is also true that $S \odot u \in \mathsf{D}\mathbb{B}_{n,m}\langle\langle\ W\ \rangle\rangle$, for $u \in X^*$.
Let us introduce an operation on row-vectors. Given $S \in \mathsf{D}\mathbb{B}_{1,m}\langle\langle\ W\ \rangle\rangle$ and $1 \leq j_0 \leq m$ we define the vector $S' = \nabla^*_{j_0}(S)$ as follows:
if $S = (a_1, \ldots, a_j, \ldots, a_m)$ then $S' = (a'_1, \ldots, a'_j, \ldots, a'_m)$ where

$$a'_j = a^*_{j_0} \cdot a_j \text{ if } j \neq j_0\ ,\ \ a'_j = \emptyset \text{ if } j = j_0.$$

Lemma 25 *Let* $S \in \mathsf{D}\mathbb{B}_{1,m}\langle\langle\ W\ \rangle\rangle$ *and* $1 \leq j_0 \leq m$.
Then $\nabla^*_{j_0}(S) \in \mathsf{D}\mathbb{B}_{1,m}\langle\langle\ W\ \rangle\rangle$.

2.5 Matrices expressing derivations

Let us define here handful notations in order to describe derivations of a grammar within a matricial formalism. For every $1 \leq n, 1 \leq i \leq n$, we define the row-vectors $\epsilon^n_i, \emptyset^n$ as: $\epsilon^n_i = (\epsilon^n_{i,j})_{1 \leq j \leq n}$ where

$$\epsilon^n_{i,j} = \emptyset(\text{ if } i \neq j);\ \ \epsilon^n_{i,i} = \epsilon;\ \ \emptyset^n = (\emptyset, \ldots, \emptyset).$$

Given a strict-deterministic grammar G (see §2.1) we fix some system of representatives mod $\smile$: $\mathcal{E} = \{E_1, \ldots, E_q\}$.
We let $N = \mathrm{Card}(V), N_i = \mathrm{Card}([E_i]_\smile)$, $[E_i]_\smile = \{E_{i,1}, E_{i,2}, \ldots, E_{i,N_i}\}$. We define the row-vectors:

$$[E_i] = (0, \ldots, 0, E_{i,1}, E_{i,2}, \ldots, E_{i,N_i}, 0, \ldots, 0)$$

where $E_{i,1}$ is placed in column $N_1 + N_2 + \ldots + N_{i-1} + 1$. For every class $[E_i]_\smile$,and every letter $x \in X$, one of the three cases is realised:

$$[E_i] \odot x = [E_j] \cdot M_{i,x},\ \text{for some } M_{i,x} \in \mathsf{D}\mathbb{B}_{N,N}\langle\langle\ V\ \rangle\rangle \tag{4}$$

where all the lines of $M_{i,x}$, with index $k \notin [N_1 + N_2 + \ldots + N_{i-1} + 1, N_1 + N_2 + \ldots + N_{i-1} + N_i]$ are null, and at least one line with index $k \in [N_1 + N_2 + \ldots + N_{i-1} + 1, N_1 + N_2 + \ldots + N_{i-1} + N_i]$ has one entry of length ≥ 1, or

$$[E_i] \odot x = [E_j] \cdot M_{i,x},\ \text{for some } M_{i,x} \in \mathsf{D}\mathbb{B}_{N,N}\langle\langle\ V\ \rangle\rangle \tag{5}$$

where all the lines of $M_{i,x}$, are either null or equal to some ϵ^N_k, or

$$[E_i] \odot x = \epsilon^N_j,\ \text{where } j \in [N_1 + \ldots + N_{i-1} + 1, N_1 + \ldots + N_{i-1} + N_i]. \tag{6}$$

2.6 Derivations

For every $S, S' \in \mathsf{D}\mathbb{B}_{1,\lambda}\langle\langle\ V\ \rangle\rangle$ and every $x \in X$, such that $S \neq \epsilon_i^\lambda$ (for every $i \in [1, \lambda]$) and $S \odot x = S'$, we must have

$$S = [E_i] \cdot T, \ \ S' = ([E_i] \odot x) \cdot T,$$

for some $i \in [1, q]$ and some $T \in \mathsf{D}\mathbb{B}_{1,\lambda}\langle\langle\ V\ \rangle\rangle$.
We write $S \uparrow (x) S'$ if the couple $([E_i], x)$ fulfills condition(4) or (5).
We write $S \downarrow (x) S'$ if the couple $([E_i], x)$ fulfills condition(5) or (6).
Given a word $u = x_1 x_2 \cdots x_\ell$, the notation $S \Uparrow (u) S'$ means that:

$$S \uparrow (x_1) S \odot x_1, S \odot x_1 \uparrow (x_2) S \odot x_1 x_2, \ldots, S \odot x_1 x_2 \cdots x_{\ell-1} \uparrow (x_\ell) S'.$$

The notation $S \Downarrow (u) S'$ means that:

$$S \downarrow (x_1) S \odot x_1, S \odot x_1 \downarrow (x_2) S \odot x_1 x_2, \ldots, S \odot x_1 x_2 \cdots x_{\ell-1} \downarrow (x_\ell) S'.$$

Let us notice that, when simultaneously $S_1 \uparrow (x) S_1'$ and $S_2 \uparrow (x) S_2'$, then

$$S_1' = \alpha_1 \cdot M \cdot S, S_2' = \alpha_2 \cdot M \cdot S \tag{7}$$

where

$$\alpha_1 = ([E_{j_1}], 0^N), \alpha_2 = (0^N, [E_{j_2}], M = \begin{pmatrix} M_{i_1,x} & 0 \\ 0 & M_{i_2,x} \end{pmatrix}, S = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}.$$

A sequence of deterministic row-vectors $S_0, S_1, \ldots, S_\ell$ is a *derivation* iff there exist $x_1, \ldots, x_\ell \in X$ such that $S_0 \odot x_1 = S_1, \ldots, S_{n-1} \odot x_\ell = S_\ell$. The *length* of this derivation is ℓ. If $u = x_1 \cdot x_2 \cdot \ldots \cdot x_\ell$ we call $S_0, S_1, \ldots, S_\ell$ the derivation *associated* with (S, u). We denote this derivation by $S_0 \xrightarrow{u} S_\ell$.
A derivation $S_0, S_1, \ldots, S_\ell$ is said to be *increasing* (resp. *decreasing*) iff it is the derivation associated to a pair (S, u) such that $S = S_0$ and $S_0 \Uparrow (u) S_n$ (resp. $S_0 \Downarrow (u) S_n$). A derivation $S_0, S_1, \ldots, S_n$ is said to be a *sub-derivation* of a derivation $S_0', S_1', \ldots, S_m'$ iff there exists some $i \in [0, m]$ such that, $\forall j \in [1, n], S_j = S_{i+j}'$.

3 Systems of equations

We deal here with systems of equations over $\mathsf{D}\mathbb{B}\langle\langle\ X\ \rangle\rangle$, where $(X, \smile)$ is the structured alphabet induced by the terminal alphabet X of a strict-deterministic grammar. (Let us recall that X has only one class w.r.t. the equivalence $\smile$.) The *divergence* between two languages $S, S' \subseteq X^*$ is defined by $\mathrm{Div}(S, S') = \inf\{|u| \mid u \in S \Delta S'\}$ (where Δ denotes the symmetric difference operation). The *valuation* of a language S can be defined as $\mathrm{Val}(S) = \mathrm{Div}(S, \emptyset)$.

Lemma 31 *Let $\alpha, \beta, S, T \in \mathsf{D}\mathbb{B}\langle\langle\ X\ \rangle\rangle, u \in X^*$. The following relations hold:*
1-$\mathrm{Div}(\alpha S, \beta S) = \mathrm{Div}(\alpha, \beta) + \mathrm{Val}(S)$
2-$\mathrm{Div}(\alpha S, \alpha T) = \mathrm{Val}(\alpha) + \mathrm{Div}(S, T)$

3-$\mathrm{Val}(S \cdot T) = \mathrm{Val}(S) + \mathrm{Val}(T)$
4-$\mathrm{Div}(S,T) \leq \mathrm{Div}(S \bullet u, T \bullet u) + |u|$
5-$\mathrm{Div}(S, \alpha \cdot S + \beta) \leq \mathrm{Div}(S, \alpha^* \beta)$
(if $(\alpha, \beta) \in \mathrm{D}\mathbb{B}_{1,2}\langle\langle\ X\ \rangle\rangle, \alpha \neq \epsilon$*).*

Proof: **Point 3**: This is well-known even for $S,T \in \mathbb{B}\langle\langle\ X\ \rangle\rangle$.
Point 1:
1.1 Let us rule out some extreme cases.
If $\mathrm{Div}(\alpha,\beta) = \infty$, this means that $\alpha = \beta$, hence it is true that $\alpha S = \beta S$, so that: $\mathrm{Div}(\alpha S, \beta S) = \infty = \mathrm{Div}(\alpha,\beta) + \mathrm{Val}(S)$.
If $\mathrm{Val}(S) = \infty$, this means that $S = \emptyset$, hence it is true that $\alpha S = \beta S$ and we again have: $\mathrm{Div}(\alpha S, \beta S) = \infty = \mathrm{Div}(\alpha,\beta) + \mathrm{Val}(S)$.
1.2 Let us suppose now that

$$\mathrm{Div}(\alpha,\beta) < \infty \text{ and } \mathrm{Val}(S) < \infty.$$

Let

$$u = \min(\alpha \Delta \beta) \text{ and } v = \min(S),$$

(here by min is meant the least word for the short-lex ordering defined by some fixed total ordering over the alphabet X).
We can suppose, for example, that

$$\alpha \bullet u = \epsilon \text{ while } \beta \bullet u \neq \epsilon.$$

By minimality of u, it must be true that: $\forall u' \preceq u, \beta \bullet u \neq \epsilon$.
Hence, by lemma 23, we have:

$$\alpha S \bullet uv = \epsilon \text{ and } \beta S \bullet uv = ((\beta \bullet u)S) \bullet v.$$

As $\beta \bullet u \neq \epsilon$ one of (8,9) must hold:

$$\exists v_1 \in X^+, \exists v_2 \in X^*, v = v_1 v_2 \text{ and } (\beta \bullet u) \bullet v_1 = \epsilon, \tag{8}$$

or

$$\forall v_1 \preceq v, (\beta \bullet u) \bullet v_1 \neq \epsilon. \tag{9}$$

When (8) holds, by minimality of v, $S \bullet v_2 \neq \epsilon$, hence $\beta S \bullet uv \neq \epsilon$.
When (9) holds, $\beta S \bullet uv = (\beta \bullet uv)S$ with $\beta \bullet uv \neq \epsilon$, hence $\beta S \bullet uv \neq \epsilon$. In both cases $uv \in \alpha S - \beta S$, so that

$$\mathrm{Div}(\alpha S, \beta S) \leq \mathrm{Div}(\alpha,\beta) + \mathrm{Val}(S).$$

1.3 Let us prove the reverse inequality.
Let $u \in \alpha S \Delta \beta S$. Notice that, for every $u' \preceq u$, it is not possible that $\alpha \bullet u' = \beta \bullet u' = \epsilon$. By lemma 23 there exists some $u' \preceq u \mid \alpha \bullet u' = \epsilon$ or $\beta \bullet u' = \epsilon$. Hence the following definition is well-founded (i.e. the set considered in its righthand-side is non-empty):

$$u_1 = \min\{u' \preceq u \mid \alpha \bullet u' = \epsilon \Leftrightarrow \beta \bullet u' \neq \epsilon\}. \tag{10}$$

For example, $\alpha \bullet u_1 = \epsilon$ while $\beta \bullet u_1 \neq \epsilon$.
Let $u_2 \in X^*$ such that $u = u_1 \cdot u_2$. As $u \in \alpha S \Delta \beta S$, we must have:

$$S \bullet u_2 = \epsilon \Leftrightarrow ((\beta \bullet u_1) \cdot S) \bullet u_2 \neq \epsilon. \tag{11}$$

By (10)

$$|u_1| \geq \mathrm{Div}(\alpha, \beta)$$

and by (11) $|u_2| \geq \min\{\mathrm{Val}(S), \mathrm{Val}((\beta \bullet u_1) \cdot S)\}$, which, by point 3 of the lemma shows that

$$|u_2| \geq \mathrm{Val}(S).$$

At end, $|u| = |u_1| + |u_2| \geq \mathrm{Div}(\alpha, \beta) + \mathrm{Val}(S)$.
Point 2:
2.1 The extreme cases where $\mathrm{Val}(\alpha) = \infty$ or $\mathrm{Div}(S, T) = \infty$ can be ruled out as in part 1.1 of this proof.
2.2 We suppose now that

$$\mathrm{Val}(\alpha) < \infty \text{ and } \mathrm{Div}(S, T) < \infty.$$

Let

$$u = \min(\alpha) \text{ and } v = \min(S \Delta T).$$

Using lemma 23 we get:

$$(\alpha S) \bullet uv = S \bullet v \text{ and } (\alpha T) \bullet uv = T \bullet v.$$

Hence $uv \in (\alpha S)\Delta(\alpha T)$, so that

$$\mathrm{Div}(\alpha S, \alpha T) \leq \mathrm{Val}(\alpha) + \mathrm{Div}(S, T).$$

2.3 Let $u \in \alpha S \Delta \alpha T$. We can assume, for example that:

$$\alpha S \bullet u = \epsilon \text{ while } \alpha T \bullet u \neq \epsilon. \tag{12}$$

We claim that there exists some decomposition $u = u' \cdot u''$ such that

$$\alpha \bullet u' = \epsilon, \alpha S \bullet u = S \bullet u'' \text{ and } \alpha T \bullet u = T \bullet u''. \tag{13}$$

If $S = \epsilon$, claim (13) is fulfilled by $u' = u, u'' = \epsilon$.
If $S \neq \epsilon$, let us consider the 3 cases of lemma 23 applied on $\alpha S \bullet u$:
- case 1 cannot occur since $(\alpha \bullet u) \cdot S \neq \epsilon$;
- case 3 cannot occur since $(\alpha S \bullet u) \neq \emptyset$.
- case 2 must then occur, which proves claim (13).
Assumption (12) together with claim (13) show that $u'' \in S \Delta T$. At end, $|u| = |u'| + |u''| \geq \mathrm{Val}(\alpha) + \mathrm{Div}(S, T)$.
Point 4:
The inclusions $u(S \bullet u) \subseteq S, u(T \bullet u) \subseteq T$ imply that

$$\mathrm{Div}(S, T) \leq \mathrm{Div}(u(S \bullet u), u(T \bullet u)).$$

Point 2 , where $\alpha = u$, gives:

$$\mathrm{Div}(u(S \bullet u), u(T \bullet u)) \le \mathrm{Div}(S \bullet u, T \bullet u) + |u|.$$

These two inequalities show the one of point 4.
Point 5: this is proved in [Sén02, equation (43) p. 580]. □

Lemma 32 (subwords lemma) *Let* $\lambda \in \mathbb{N} - \{0\}$. *Then, for every* $n \in \mathbb{N}, \alpha, \beta \in \mathsf{D}\mathbb{B}_{1,\lambda}\langle\langle\ X\ \rangle\rangle, A_1, A_2 \ldots, A_\lambda \in \mathsf{D}\mathbb{B}_{\lambda,\lambda}\langle\langle\ X\ \rangle\rangle, S \in \mathsf{D}\mathbb{B}_{\lambda,1}\langle\langle\ X\ \rangle\rangle$,
if all the equations

$$\alpha \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S =_n \beta \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S$$

$$\textit{for } 0 \le p \le \lambda - 1, 1 \le i_1 < i_2 < \ldots < i_p \le \lambda$$

are valid, then the following equation is valid too:

$$\alpha \cdot A_1 A_2 \cdots A_\lambda \cdot S =_n \beta \cdot A_1 A_2 \cdots A_\lambda \cdot S.$$

Proof: We prove the lemma by induction on λ.
Basis: λ=1.
By lemma 31,

$$\begin{aligned}\mathrm{Div}(\alpha AS, \beta AS) &= \mathrm{Div}(\alpha, \beta) + \mathrm{Val}(AS)\\ &= \mathrm{Div}(\alpha, \beta) + \mathrm{Val}(A) + \mathrm{Val}(S)\\ &= \mathrm{Div}(\alpha S, \beta S) + \mathrm{Val}(A).\end{aligned}$$

Hence, for every $n \ge 0$,

$$\alpha S =_n \beta S \Rightarrow \alpha AS =_n \beta AS.$$

Induction step:$\lambda \to \lambda + 1$:
Let us suppose that

$$\alpha \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S =_n \beta \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S \tag{14}$$

$$\text{for } 0 \le p \le \lambda, 1 \le i_1 < i_2 < \ldots < i_p \le \lambda + 1.$$

Let $u \in X^*, |u| \le n$.
Case 1: $\forall v \preceq u, \forall j \in [1, \lambda + 1], \alpha \bullet v \ne \epsilon_j^{\lambda+1}, \beta \bullet v \ne \epsilon_j^{\lambda+1}$.
Then, by lemma 23, points 1,3,

$$(\alpha \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u = (\alpha \bullet u) \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S;\ (\beta \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u = (\beta \bullet u) \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S$$

hence

$$(\alpha \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u =_0 (\beta \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u.$$

Case 2: $\exists v \preceq u, \exists j \in [1, \lambda+1], \alpha \bullet v = \epsilon_j^{\lambda+1} = \beta \bullet v$.
Then, by lemma 23, point 2,

$$(\alpha \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u = (\epsilon_j^{\lambda+1} \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet w = (\beta \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u.$$

(where $u = v \cdot w$).
Case 3: $\exists v \preceq u$ such that

$$\exists j \in [1, \lambda+1], \alpha \bullet v = \epsilon_j^{\lambda+1} \Leftrightarrow \beta \bullet v \neq \epsilon_j^{\lambda+1}.$$

Let $v \preceq u$ be the smallest prefix of u fulfilling the above property. We may suppose, for example that

$$\alpha \bullet v = \epsilon_{j_0}^{\lambda+1} \text{ while } \beta \bullet v \neq \epsilon_{j_0}^{\lambda+1}.$$

The minimality of v implies that

$$\forall v' \preceq v, \forall j \in [1, \lambda+1], \beta \bullet v' \neq \epsilon_j^{\lambda+1}.$$

By lemma 23 it follows that

$$(\alpha \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S) \bullet v = \epsilon_{j_0}^{\lambda+1} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S \tag{15}$$

$$(\beta \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S) \bullet v = (\beta \bullet v) \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S \tag{16}$$

$$\text{for } 0 \leq p \leq \lambda, 1 \leq i_1 < i_2 < \ldots < i_p \leq \lambda+1.$$

Up to some permutation of the indices (rows and columns) in all matrices, we can suppose that $j_0 = \lambda + 1$.
Let us note $n' = n - |v|$. Letting v act (by $\bullet$) on both sides of equations (14), we obtain the set of equations:

$$\epsilon_{\lambda+1}^{\lambda+1} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S =_{n'} (\beta \bullet v) \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S. \tag{17}$$

The fact that $\beta \bullet v \neq \epsilon_{\lambda+1}^{\lambda+1}$ allows us to apply Arden's lemma:
let $\gamma \in \mathsf{D}\mathbb{B}_{1,\lambda}\langle\langle\, X \,\rangle\rangle$ defined by:

$$\gamma = ((\beta_{\lambda+1} \bullet v)^*(\beta_1 \bullet v), (\beta_{\lambda+1} \bullet v)^*(\beta_2 \bullet v), \ldots, (\beta_{\lambda+1} \bullet v)^*(\beta_\lambda \bullet v))$$

Each equation (17) gives rise to the equation

$$\epsilon_{\lambda+1}^{\lambda+1} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S =_{n'} (\gamma, 0) \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S. \tag{18}$$

Notice that $(\gamma, 0) = \nabla_{\lambda+1}^*(\beta \bullet v)$ hence, by lemma 24,point 2 and lemma 25, it belongs to $\mathsf{D}\mathbb{B}_{1,\lambda+1}\langle\langle\, X \,\rangle\rangle$.
We introduce the matrices $P \in \mathsf{D}\mathbb{B}_{\lambda+1,\lambda}\langle\langle\, X \,\rangle\rangle, \bar{P} \in \mathsf{D}\mathbb{B}_{\lambda,\lambda+1}\langle\langle\, X \,\rangle\rangle$:

$$P = \begin{pmatrix} \mathrm{I}_\lambda \\ \gamma \end{pmatrix}; \;\; \bar{P} = \begin{pmatrix} \mathrm{I}_\lambda \; 0_\lambda^1 \end{pmatrix}.$$

Each equation (18) can be rewritten as:

$$A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S =_{n'} P \cdot \bar{P} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S. \tag{19}$$

Let us consider one equation (17) where $i_1 = 1, p \leq \lambda$. In such a single equation, taking into account the different equations (19) associated with all the suffixes of the sequence $i_2, i_3, \ldots, i_p$, we obtain the new equation:

$$(\epsilon_{\lambda+1}^{\lambda+1} A_1)(P\bar{P}A_{i_2})\cdot(P\bar{P}A_{i_3})\cdots(P\bar{P}A_{i_p})\cdot(P\bar{P}S) =_{n'} (\beta \bullet v A_1)(P\bar{P}A_{i_2})\cdot(P\bar{P}A_{i_3})\cdots(P\bar{P}A_{i_p})\cdot(P\bar{P}S).$$

which can be bracketed, as well,

$$(\epsilon_{\lambda+1}^{\lambda+1} A_1 P)\cdot(\bar{P}A_{i_2}P)\cdot(\bar{P}A_{i_3}P)\cdots(\bar{P}A_{i_p}P)\cdot(\bar{P}S) =_{n'} (\beta \bullet v A_1 P)\cdot(\bar{P}A_{i_2}P)\cdot(\bar{P}A_{i_3}P)\cdots(\bar{P}A_{i_p}P)\cdot(\bar{P}S). \tag{20}$$

Let us take:

$$\alpha' = \epsilon_{\lambda+1}^{\lambda+1} A_1 P, \beta' = (\beta \bullet v) A_1 P, A'_j = \bar{P} A_j P (\text{ for } 2 \leq j \leq \lambda + 1), S' = \bar{P} S.$$

The items $n', \alpha', \beta', A'_j (2 \leq j \leq \lambda + 1), S'$ are fulfilling the hypothesis of the lemma for the integer λ. By induction hypothesis, it must be true that

$$\alpha' \cdot A'_2 A'_3 \cdots A'_{\lambda+1} \cdot S' =_{n'} \beta' \cdot A'_2 A'_3 \cdots A'_{\lambda+1} \cdot S'. \tag{21}$$

Using now the equations (19) "backwards", equation (21) can be translated as:

$$\epsilon_{\lambda+1}^{\lambda+1} \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S =_{n'} (\beta \bullet v) \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S. \tag{22}$$

Let $w \in X^*$ be the suffix such that $u = vw$. As $|w| = n'$, by equation (22)we get

$$(\epsilon_{\lambda+1}^{\lambda+1} \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet w =_0 (\beta \bullet v \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet w.$$

i.e.

$$(\alpha \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u =_0 (\beta \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S) \bullet u.$$

As this is true for every $|u| \leq n$, we can conclude that

$$\alpha \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S =_n \beta \cdot A_1 A_2 \cdots A_{\lambda+1} \cdot S.$$

□

Remark 33 *By means of the homorphism φ, the same lemma holds for $\mathsf{D}\mathbb{B}\langle\langle\, V\, \rangle\rangle$ (instead of $\mathsf{D}\mathbb{B}\langle\langle\, X\, \rangle\rangle$) and for the equivalence relations $\equiv_n$ (instead of $=_n$).*

4 Deduction rules

4.1 The deduction relation

We denote by $\mathcal{A}$ the set $\mathsf{D}\mathbb{B}\langle\langle\, V\, \rangle\rangle \times \mathsf{D}\mathbb{B}\langle\langle\, V\, \rangle\rangle$.

We define then a binary relation $|\!\!\vdash\!\!- \subseteq \mathcal{P}(\mathcal{A}) \times \mathcal{A}$, the *elementary deduction relation*, as the set of all the pairs having one of the following forms:

(R0)

$$\emptyset \mid\!\vdash\!- (S,S)$$

for $S \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle$,

(R1)

$$\{(S,T)\} \mid\!\vdash\!- (T,S)$$

for $S,T \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle$,

(R2)

$$\{(S,T),(T,U)\} \mid\!\vdash\!- (S,U)$$

for $S,T,U \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle$,

(R3)

$$\{(S,S'),(T,T')\} \mid\!\vdash\!- (S+T,S'+T')$$

for $(S,T),(S',T') \in \mathsf{D}\mathbb{B}_{1,2}\langle\langle\ V\ \rangle\rangle$,

(R4)

$$\{(S,S')\} \mid\!\vdash\!- (S\cdot T,S'\cdot T)$$

for $S,S',T \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle$,

(R5)

$$\{(T,T')\} \mid\!\vdash\!- (S\cdot T,S\cdot T')$$

for $S,T,T' \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle$,

(R6)

$$\{(S\cdot T'+S',T')\} \mid\!\vdash\!- (S^{*}\cdot S',T')$$

for $(S,S') \in \mathsf{D}\mathbb{B}_{1,2}\langle\langle\ V\ \rangle\rangle, T' \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle, S \neq \epsilon$,

(R7)

$$\{(\varepsilon,S)\} \mid\!\vdash\!- (T,U)$$

for $S,T,U \in \mathsf{D}\mathbb{B}\langle\langle\ V\ \rangle\rangle, S \neq \varepsilon$.

(R8)

$$\{(\alpha A_{i_1}A_{i_2}\cdots A_{i_p}S,\beta A_{i_1}A_{i_2}\cdots A_{i_p}) \mid 1 \leq i_1 < i_2 < \ldots < i_p \leq n, 0 \leq p \leq n-1\}$$

$\mid\!\vdash\!- (\alpha A_1A_2\cdots A_nS,\beta A_1A_2\cdots A_nS)$
for $\alpha,\beta \in \mathsf{D}\mathbb{B}_{1,n}\langle\langle\ V\ \rangle\rangle, A_1,A_2,\ldots,A_n \in \mathsf{D}\mathbb{B}_{n,n}\langle\langle\ V\ \rangle\rangle, S \in \mathsf{D}\mathbb{B}_{n,1}\langle\langle\ V\ \rangle\rangle$.

Remark 41
One can check that, by the results of §2.4, the above rules really belong to $\mathcal{P}(\mathcal{A})\times\mathcal{A}$.

Finally, the binary relation $\vdash\!-\ \subseteq \mathcal{P}(\mathcal{A})\times\mathcal{P}(\mathcal{A})$ is defined by: $\forall P,Q \in \mathcal{P}(\mathcal{A})$

$$P \vdash\!- Q \Leftrightarrow (\forall q \in Q-P, \exists P' \subseteq P, \text{ such that } P' \mid\!\vdash\!- q).$$

By $\overset{*}{\vdash\!-}$ we denote the transitive closure of $\vdash\!-$, we call it the *deduction relation*.

4.2 Properties

Lemma 42 *: For every $P, Q \in \mathcal{P}(\mathcal{A})$,*

$$P \overset{*}{\vdash\!-} Q \Rightarrow \mathrm{Div}(P) \leq \mathrm{Div}(Q)$$

Proof: It suffices to check that every rule $\mathcal{H} \,|\!\vdash\!- q$ fulfills:

$$\mathrm{Div}(\mathcal{H}) \leq \mathrm{Div}(q).$$

The only non-trivial checks are for rules of type R6,R8.The case of R6 is ruled out by lemma 31, point 5. The case of R8 is ruled out by lemma 32 and remark 33. □

Lemma 43 *: For every $P, Q \in \mathcal{P}(\mathcal{A}), x \in X$,*

$$(P \overset{*}{\vdash\!-} Q) \Rightarrow (P \cup P \odot x \overset{*}{\vdash\!-} Q \odot x).$$

Proof: It suffices to check that, every rule $\mathcal{H} \,|\!\vdash\!- q$ fulfills:

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!-} \{q \odot x\}.$$

Let us distinguish every type of elementary rule.

(R0): $q = (S, S)$. Hence $q \odot x = (S \odot x, S \odot x)$, hence $\emptyset \overset{*}{\vdash\!-} q \odot x$.
(R1,R2,R3): easy verification.
(R4):$\mathcal{H} = \{(S, S')\}, q = (S \cdot T, S' \cdot T)$.
If $S = S' = \epsilon$:
then $q \odot x = (T \odot x, T \odot x)$, hence $\emptyset \overset{*}{\vdash\!-} q \odot x$.
If $(S = \epsilon) \Leftrightarrow (S' \neq \epsilon)$:
then, by R7, $\{(S, S')\} \,|\!\vdash\!- q \odot x$
If $S \neq \epsilon, S' \neq \epsilon$:
$q \odot x = (S \odot x \cdot T, S' \odot x \cdot T)$. Hence $\mathcal{H} \odot x \,|\!\vdash\!-_{R4} q \odot x$.
(R5):$\mathcal{H} = \{(T, T')\}, q = (S \cdot T, S \cdot T')$.
If $S \neq \epsilon$, $q \odot x = ((S \odot x) \cdot T, (S \odot x) \cdot T')$.
Hence $\mathcal{H} \vdash\!- \{q \odot x\}$.
If $S = \epsilon$, $\mathcal{H} \odot x \overset{*}{\vdash\!-} q \odot x$.
(R6):$\mathcal{H} = \{(S \cdot T' + S', T')\}, q = (S^* \cdot S', T')$ (with $S \neq \epsilon$).
Let us notice that, $S^* \cdot S' = S \cdot S^* \cdot S' + S'$. Using the fact that $S \neq \epsilon$, we obtain:

$$(S^* \cdot S') \odot x = (S \odot x) \cdot S^* \cdot S' + S' \odot x.$$

As well, S being not a unit implies that:

$$\mathcal{H} \odot x = \{((S \odot x) \cdot T' + S' \odot x, T' \odot x)\},$$

$$q \odot x = (S \odot x) \cdot S^* S' + S' \odot x, T' \odot x).$$

Consider the following deductions:

$$\{q\} = \{(S^* \cdot S', T')\} \vdash\!\!-_{R5} \{((S \odot x) \cdot S^* \cdot S', (S \odot x) \cdot T')\}; \tag{23}$$

given the fact that the line-vectors $((S \odot x) \cdot S^* \cdot S', S' \odot x), ((S \odot x)T', S' \odot x)$ are deterministic (because they have the form $(S \odot x, S' \odot x) \cdot \gamma$ for suitably chosen deterministic matrices γ)

$$\{((S \odot x) \cdot S^* \cdot S', (S \odot x) \cdot T')\} \vdash\!\!-_{R3} \{((S \odot x) \cdot S^* \cdot S' + S' \odot x, (S \odot x) \cdot T' + S' \odot x)\}. \tag{24}$$

These deductions (23,24), together with rules R0,R3 show that:

$$\{q\} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{((S \odot x) \cdot S^* \cdot S' + S' \odot x, T' \odot x)\} = \{q \odot x\}.$$

As $\mathcal{H} \overset{*}{\vdash\!\!-} \{q\}$, it follows that

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{q \odot x\}.$$

(R7):$\mathcal{H} = \{(\epsilon, S)\}, q = (T, U)$ (with $S \neq \epsilon$).
Then $\{(\epsilon, S)\} \mid\!\vdash\!\!- (T \odot x, U \odot x)$.
(R8): From now on, we call $\mathcal{C}$ the set of instances of metarules (R1-R7). At this stage of the proof we have already established that, for every $\mathcal{L} \subseteq \mathcal{A}, q \in \mathcal{A}$

$$\mathcal{L} \mid\!\vdash\!\!-_{\mathcal{C}} q \Rightarrow (\mathcal{L} \cup \mathcal{L} \odot x \overset{*}{\vdash\!\!-}_{\mathcal{C}} \{q \odot x\}). \tag{25}$$

We prove by induction on $n \geq 1$ the following:
for every $\alpha, \beta \in \mathsf{D}\mathbb{B}_{1,n}\langle\langle\ V\ \rangle\rangle, A_1, A_2, \ldots, A_n \in \mathsf{D}\mathbb{B}_{n,n}\langle\langle\ V\ \rangle\rangle, S \in \mathsf{D}\mathbb{B}_{n,1}\langle\langle\ V\ \rangle\rangle$,
if

$$\mathcal{H} = \{(\alpha A_{i_1} A_{i_2} \cdots A_{i_p} S, \beta A_{i_1} A_{i_2} \cdots A_{i_p}) \mid 1 \leq i_1 < i_2 < \ldots < i_p \leq n, 0 \leq p \leq n-1\},$$

and

$$q = (\alpha A_1 A_2 \cdots A_n S, \beta A_1 A_2 \cdots A_n S),$$

then

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{q \odot x\}.$$

Basis: $n = 1$.
Case 1: $\alpha \neq \epsilon, \beta \neq \epsilon$.
$q \odot x = (\alpha \odot x A_1 S, \beta \odot x A_1 S)$, hence $\mathcal{H} \odot x \vdash\!\!- \{q \odot x\}$.
Case 2: $\alpha = \epsilon = \beta$.
Then $\emptyset \mid\!\vdash\!\!- q \odot x$ hence $\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{q \odot x\}$.
Case 3: $\alpha = \epsilon \Leftrightarrow \beta \neq \epsilon$.
We may suppose, for example, that

$$\alpha = \epsilon \text{ while } \beta \neq \epsilon.$$

In other words: $\mathcal{H} = \{(S, \beta \cdot S)\}$ and $q = (A_1 \cdot S, \beta \cdot A_1 \cdot S)$.
If $A_1 = \epsilon$, then $q \odot x \in \mathcal{H} \odot x$.
If $A_1 \neq \epsilon$ then $q \odot x = (A_1 \odot x \cdot S, \beta \odot x \cdot A_1 \cdot S)$.
From the following deductions:

$$(S, \beta S) \mid\!\vdash\!-_{R6} (S, 0), \;\; (S, 0) \mid\!\vdash\!-_{R5} (\alpha S, 0), \;\; (S, 0) \mid\!\vdash\!-_{R5} (\beta S, 0),$$

and $\{(\alpha S, 0), (\beta S, 0)\} \overset{*}{\vdash\!-}_{\{R1,R2\}} \{(\alpha S, \beta S)\}$, it follows that $\mathcal{H} \overset{*}{\vdash\!-} \{q \odot x\}$.
Induction step: $n = \lambda + 1, \lambda \geq 1$.
Case 1: $\forall j \in [1, n], \alpha \neq \epsilon_j^n, \beta \neq \epsilon_j^n$.
We can conclude as for $n = 1$.
Case 2: $\exists j \in [1, n], \alpha = \epsilon_j^n = \beta$.
We can conclude as for $n = 1$.
Case 3: $\exists j \in [1, n], \alpha = \epsilon_j^n \Leftrightarrow \beta \neq \epsilon_j^n$.
We may suppose, for example, that

$$\alpha = \epsilon_{\lambda+1}^{\lambda+1} \text{ while } \beta \neq \epsilon_{\lambda+1}^{\lambda+1}.$$

We follow the same lines of argumentation as in the proof of lemma 32. We suppose here that $\mathcal{H}$ consists of all the equations

$$(\epsilon_{\lambda+1}^{\lambda+1} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S, \beta \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S) \tag{26}$$

for $0 \leq p \leq \lambda, 1 \leq i_1 < i_2 < \ldots < i_p \leq \lambda + 1$. Let

$$\gamma = (\beta_{\lambda+1}^* \beta_1, \beta_{\lambda+1}^* \beta_2, \ldots, \beta_{\lambda+1}^* \beta_\lambda).$$

$$P = \begin{pmatrix} \mathrm{I}_\lambda \\ \gamma \end{pmatrix}; \;\; \bar{P} = \begin{pmatrix} \mathrm{I}_\lambda & 0_\lambda^1 \end{pmatrix}.$$

Taking into account the fact that $\beta_{\lambda+1} \neq \epsilon$, we can apply rule (R6) on every equation (26); we obtain:

$$\mathcal{H} \overset{*}{\vdash\!-}_{\mathcal{C}} \{(\epsilon_{\lambda+1}^{\lambda+1} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S, (\gamma, 0) \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S)\}, \tag{27}$$

for $0 \leq p \leq \lambda, 1 \leq i_1 < i_2 < \ldots < i_p \leq \lambda + 1$.
For every matrices $M, N \in \mathsf{D}\mathbb{B}_{k,\ell}\langle\langle\ V\ \rangle\rangle$, let us denote by $[M, N]$ the set of equations:

$$[M, N] = \{(M_{i,j}, N_{i,j}) \mid 1 \leq i \leq k, 1 \leq j \leq \ell\}.$$

Each deduction (27) allows to state:

$$\mathcal{H} \overset{*}{\vdash\!-}_{\mathcal{C}} [A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S, P \cdot \bar{P} \cdot A_{i_1} A_{i_2} \cdots A_{i_p} \cdot S]. \tag{28}$$

Let us define

$$\alpha' = \epsilon_{\lambda+1}^{\lambda+1} A_1 P, \beta' = \beta A_1 P, A'_j = \bar{P} A_j P(\text{ for } 2 \leq j \leq \lambda + 1), S' = \bar{P} S.$$

All the equations (28) can be used to build the following $\mathcal{C}$-deductions:

$$\mathcal{H} \overset{*}{\vdash\!\!-}_{\mathcal{C}} \{(\alpha' A'_{i_2} \cdot A'_{i_3} \cdots A'_{i_p} \cdot S', \alpha A_1 A_{i_2} \cdot A_{i_3} \cdots A_{i_p} \cdot S)\} \tag{29}$$

$$\mathcal{H} \overset{*}{\vdash\!\!-}_{\mathcal{C}} \{(\beta' A'_{i_2} \cdot A'_{i_3} \cdots A'_{i_p} \cdot S', \beta A_1 A_{i_2} \cdot A_{i_3} \cdots A_{i_p} \cdot S)\} \tag{30}$$

for $0 \leq p \leq \lambda, 2 \leq i_2 < \ldots < i_p \leq \lambda + 1$. We call these two types of equations (29,30) the *translation*-equations. From each equation (26) where $i_1 = 1$, taking into account the translation equations, we obtain the new deduction:

$$\mathcal{H} \overset{*}{\vdash\!\!-}_{\mathcal{C}} \{(\alpha' A'_{i_2} A'_{i_3} \cdots A'_{i_p} \cdot S', \beta' A'_{i_2} A'_{i_3} \cdots A'_{i_p} \cdot S')\} \tag{31}$$

for $1 \leq p \leq \lambda, 2 \leq i_2 < \ldots < i_p \leq \lambda + 1$. Let us note

$$\mathcal{H}' = \{(\alpha' A'_{i_2} \cdots A'_{i_p} S', \beta' A'_{i_2} \cdots A'_{i_p} S') \mid 1 \leq p \leq \lambda, 2 \leq i_2 < \ldots < i_p \leq n\}.$$

By (31), $\mathcal{H} \overset{*}{\vdash\!\!-}_{\mathcal{C}} \mathcal{H}'$, hence, by (25):

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-}_{\mathcal{C}} \mathcal{H}' \cup \mathcal{H}' \odot x \tag{32}$$

The translation equations, together with property (25) show that:

$$\begin{aligned} \mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-}_{\mathcal{C}} & \{((\alpha' A'_{i_2} A'_{i_3} \cdots A'_{i_p} S') \odot x, (\alpha A_1 A_{i_2} A_{i_3} \cdots A_{i_p} S) \odot x)\} \\ \cup \quad & \{((\beta' A'_{i_2} A'_{i_3} \cdots A'_{i_p} S') \odot x, (\beta A_1 A_{i_2} A_{i_3} \cdots A_{i_p} S) \odot x)\}. \end{aligned} \tag{33}$$

By induction hypothesis, as $\mathcal{H}' \mid\!\vdash\!\!-_{R8} (\alpha' A'_2 \cdots A'_{\lambda+1} S', \beta' A'_2 \cdots A'_{\lambda+1})$

$$\mathcal{H}' \cup \mathcal{H}' \odot x \overset{*}{\vdash\!\!-} \{((\alpha' A'_2 \cdots A'_{\lambda+1} S') \odot x, (\beta' A'_2 \cdots A'_{\lambda+1}) \odot x)\} \tag{34}$$

The deductions (32,33,34) alltogether show that:

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{((\alpha A_1 A_2 \cdots A_{\lambda+1} S) \odot x, (\beta A_1 A_2 \cdots A_{\lambda+1} S) \odot x)\}$$

i.e.

$$\mathcal{H} \cup \mathcal{H} \odot x \overset{*}{\vdash\!\!-} \{q \odot x\}.$$

□

4.3 Self-provable sets

Let $P \subseteq \mathcal{A}$. The subset P is said:
ε-consistent iff $\forall (S,T) \in P, (S = \varepsilon) \Leftrightarrow (T = \varepsilon)$
right-stable iff $\forall x \in X, P \overset{*}{\vdash\!\!-} P \odot x$,
self-provable iff P is ε-consistent and right-stable.

Lemma 44 *If P is self-provable then* $\mathrm{Div}(P) = \infty$.

In other words, if P is self-provable, every pair (S,T) belonging to P consists of equivalent series.

5 Application to t-turn dpda

We show here that the divergence of two t-turn dpda is upper-bounded by some polynomial function of the size of the automata.

5.1 Turns and weights

We relate in this paragraph the notion of t-turn deterministic pushdown automaton with a notion of k-weighted strict-deterministic grammar.
As we essentially deal with grammars and do not use pushdown automata, (except for comparison with the previous works by [Val74,Bee76]), we limit ourselves to some minimal statements and sketch of proofs concerning automata.

Definition 51 *Let* $G =< X, V, P >$ *be a normalised strict-deterministic context-free grammar and let* k *be an integer.* G *is said to be* k*-weighted iff there exists a map* $\tau : V \to [0, k]$ *such that every rule of* P *has one of the following forms:*
1- $v \to x \cdot v_1 v_2$*, with* $v, v_1, v_2 \in V, x \in X, \tau(v) \geq \tau(v_1) + \tau(v_2)$ *and* $\tau(v_1) \geq 1$
2- $v \to x \cdot v_1$*, with* $v, v_1 \in V, x \in X, \tau(v) \geq \tau(v_1)$
3- $v \to x$*, with* $v \in V, x \in X$*.*

We recall that a deterministic pushdown automaton $\mathcal{M}$ is said to be *t-turn* iff, every successful computation of $\mathcal{M}$: $q_0 z_0 \xrightarrow{u}_{\mathcal{M}} q$ can be factorised into a number $s \leq t + 1$ of sub-computations:

$$q_0 z_0 \xrightarrow{u_1}_{\mathcal{M}} q_1 \omega_1 \xrightarrow{u_2}_{\mathcal{M}} q_2 \omega_2 \ldots \xrightarrow{u_i}_{\mathcal{M}} q_i \omega_i \ldots \xrightarrow{u_s}_{\mathcal{M}} q_s \omega_s$$

such that, for i even (resp. odd), the computation $q_{i-1}\omega_{i-1} \xrightarrow{u_i}_{\mathcal{M}} q_i \omega_i$ uses decreasing (resp. increasing) transitions only. The number $t+1$ is thus an upper-bound on the number of successive "monotone strokes" while t is an upper-bound on the number of "turns" of every computation of $\mathcal{M}$. A d.p.d.a. $\mathcal{M}$ is said *finite-turn* iff it is t-turn for some integer t.

Theorem 52 *1- Let* $G =< X, V, P >$ *be a* k*-weighted strict-deterministic context-free grammar and* $v \in V$*. Then, one can construct a dpda* $\mathcal{M}$*, which is* $2k - 1$*-turn, and recognizes* $\mathrm{L}(G, v)$*.*
2- Let $\mathcal{M}$ *be some* t*-turn (*t *odd integer) dpda, with empty-stack accepting modes. Then, one can construct a* $(t + 1)/2$*-weighted strict-deterministic context-free grammar* $G =< X, V, P >$ *and* $v \in V$*, such that* $\mathrm{L}(G, v)$ *is the language recognized by* $\mathcal{M}$*.*
Moreover, both transformations can be achieved in deterministic polynomial time.

Sketch of proof: 1- Suppose G, v are given. For every word $w \in V^*$, let $t(w) = 2\tau(w) - 1$. For every polynomial $S \in \mathsf{D}\mathbb{B}_{1,1}\langle\, V \,\rangle$ we define $t(S) = \max\{t(w) \mid w \in S\}$. One can show that every derivation $S \xrightarrow{u} \epsilon$ can be factorised into at most $t(S) + 1$ derivations:

$$S \Uparrow (u_1)\ S_1 \Downarrow (u_2)\ S_2 \Uparrow (u_3)\ S_3 \cdots \Downarrow (u_{t(S)+1})\ \epsilon$$

with $u = u_1 \cdot u_2 \cdots u_{t(S)+1}$. From this one can extract the dpda $\mathcal{M}$.
2- Suppose $\mathcal{M}$ is given. Let G be the strict-deterministic grammar contructed from $\mathcal{M}$ as in [Har78, lemma 11.5.2]. Let us define the weight of variable $[p, z, q]$ (p, q are states of the automaton, z is a letter of the stack-alphabet) as: $\tau([p, z, q]) =$ "the maximum number of up-strokes in a computation starting from state p and top-symbol z". The grammar obtained fails to meet definition 51, just because of rules of the form $v \to x \cdot v_1 v_2$ where $\tau(v_1) = 0$. It remains to eliminate these productions by introducing variables in bijection with the words of length 2: $\{v_1 \cdot v \mid v_1, v \in V, \tau(v) > \tau(v_1) = 0\}$ and the corresponding productions. The transformed grammar G' is k-weighted with $k = (t+1)/2$. □
We fix a k-weighted strict-deterministic c.f. grammar $G =< X, V, P >$ within this section. We also fix two variables $v_1, v_2 \in V$ and deal with the equality problem for $\mathrm{L}(G, v_1), \mathrm{L}(G, v_2)$.

5.2 Parallel derivations

We show that every long increasing derivation, must contain a sequence of equations which is a "germ" for an application of rule R8 (which is based on the subwords lemma).

Lemma 53 *Let us suppose that* $S_1, S_2, S'_1, S'_2 \in \mathsf{D}\mathbb{B}_{1,1}\langle\langle\ V\ \rangle\rangle, u \in X^*$ *and* $S_i \Uparrow (u)\ S'_i$ *(for* $i \in \{1, 2\}$*). If* $|u| \geq 2N^3$*, there exist* $\alpha, \beta \in \mathsf{D}\mathbb{B}_{1,2N}\langle\langle\ V\ \rangle\rangle$, $|\alpha| \geq 1, |\beta| \geq 1, u', u'' \in X^*, u_{2N}, u_{2N-1}, \ldots, u_1 \in X^+$, $M_{2N}, M_{2N-1}, \ldots, M_1 \in \mathsf{D}\mathbb{B}_{2N,2N}\langle\langle\ V\ \rangle\rangle, S \in \mathsf{D}\mathbb{B}_{2N,1}\langle\langle\ V\ \rangle\rangle$ *such that:*
1- $u' \cdot u_{2N} \cdot u_{2N-1} \cdots u_1 \cdot u'' = u$
2- $S_1 \Uparrow (u')\ \alpha \cdot S \Uparrow (u_{2N})\ \alpha \cdot M_{2N} \cdot S \ldots \Uparrow (u_i)\ \alpha \cdot M_i M_{i+1} \cdots M_{2N} \cdot S \ldots \Uparrow (u_1)\ \alpha \cdot M_1 M_2 \cdots M_{2N} \cdot S \Uparrow (u'') S'_1$
3- $S_2 \Uparrow (u')\ \beta \cdot S \Uparrow (u_{2N})\ \beta \cdot M_{2N} \cdot S \ldots \Uparrow (u_i)\ \beta \cdot M_i M_{i+1} \cdots M_{2N} \cdot S \ldots \Uparrow (u_1)\ \beta \cdot M_1 M_2 \cdots M_{2N} \cdot S \Uparrow (u'')\ S'_2$

Sketch of proof: By the form of the transitions (4,5), and the trick mentionned in equation(7), for every prefix w_j of length j, of the word u,

$$S_1 \odot w_j = [E_{k_j}] \cdot M_j M_{j-1} \cdots M_1 \cdot S_0,$$

$$S_2 \odot w_j = [E_{\ell_j}] \cdot M_j M_{j-1} \cdots M_1 \cdot S_0,$$

for some integers $k_j, \ell_j \in [1, q], M_j \in \mathsf{D}\mathbb{B}_{2N,2N}\langle\langle\ V\ \rangle\rangle$, and $S_0 = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}$. As the couples (k_j, ℓ_j) can take at most $q^2 \leq N^2$ values, one of them must appear $2N$ times. By extraction and renaming, a sequence satisfying points (1),(2),(3) of the lemma can be obtained. □

5.3 A right-stable set

For every integer $n \geq 0$ let us define

$$P_n = \{(v_1 \odot u, v_2 \odot u) \mid u \in X^{\leq n}\}.$$

Let $N_1 = 1 + 2 \cdot (1 + 2 \cdot N^3)^{4 \cdot k}$.

Lemma 54 *The set P_{N_1} is right-stable.*

Proof: In step 1, we show that, for every $u \in X^{N_1}$, there exists a prefix $u' \preceq u$ such that $P_{|u'|-1} \overset{*}{\vdash\!\!-} \{(v_1 \odot u', v_2 \odot u')\}$. In step 2, we conclude that P_{N_1} is right-stable.
Step 1 Let $u_0 \in X^{N_1}$.
Case 1 Suppose that the word u_0 admits a decomposition $u_0 = u_0' u u_0''$ such that, for every $i \in 1, 2$,

$$v_i \xrightarrow{u_0'} S_i \Uparrow (u)\, S_i' \xrightarrow{u_0''} S_i$$

with $|u| \geq 2N^3$.
In this case, by lemma 53, points (1),(2),(3) of this lemma are true. Notice that, for every subsequence of indices:

$$1 \leq i_1 < i_2 < \ldots < i_p \leq 2N, 0 \leq p \leq 2N - 1$$

we have

$$(v_1 \odot (u_0' \cdot u_{i_p} \cdots u_{i_2} \cdot u_{i_1}), v_2 \odot (u_0' \cdot u_{i_p} \cdots u_{i_2} \cdot u_{i_1}) = (\alpha M_{i_1} M_{i_2} \cdots M_{i_p} S, \beta M_{i_1} M_{i_2} \cdots M_{i_p} S) \tag{35}$$

Every lefthand-side of an identity (35) belongs to $P_{|u_0' u|-1}$. The set of all the righthand-sides of identities (35) allows to deduce the equation $(v_1 \odot u_0' u, v_2 \odot u_0' u)$, by rule R8. Therefore, the prefix $u' = u_0' u \preceq u_0$ has the property that

$$P_{|u'|-1} \overset{*}{\vdash\!\!-} \{(v_1 \odot u', v_2 \odot u')\}. \tag{36}$$

Case 2
Let us suppose that the word u_0 admits no decomposition of the form assumed in case 1.
By the arguments of part 1 of the proof of theorem 52, the whole parallel derivation $(v_1, v_2) \xrightarrow{u_0} (v_1 \cdot u_0, v_2 \cdot u_0)$ can be factorised into at most $4k$ derivations:

$$v_1 = S_{1,0} \xrightarrow{u_1} S_{1,1} \xrightarrow{u_2} S_{1,2} \xrightarrow{u_3} S_{1,3} \cdots \xrightarrow{u_{4k}} S_{1,4k}$$

$$v_2 = S_{2,0} \xrightarrow{u_1} S_{2,1} \xrightarrow{u_2} S_{2,2} \xrightarrow{u_3} S_{2,3} \cdots \xrightarrow{u_{4k}} S_{2,4k}$$

with $u = u_1 \cdot u_2 \cdots u_{4k}$ and every derivation $S_{i,j} \xrightarrow{u_j} S_{i,j+1}$ is monotone i.e. either increasing or decreasing. Let us denote

$$H(j) = \max\{|S_{1,j}|, |S_{2,j}|\}; \;\; T(j) = |u_j|.$$

(Intuitively, $H(j)$ is the height of configuration $(S_{1,j}, S_{2,j})$ while $T(j)$ is the time consumed by the j-th monotone factor of the derivation).
Let us distinguish the four types of monotonicity.
$\underline{S_{i,j-1} \Uparrow (u_j)\ S_{i,j+1}}$ (for both $i \in \{1,2\}$):
By the assumption of case 2, $T(j) \leq 2 \cdot N^3$, hence $H(j) \leq H(j-1) + 2 \cdot N^3$.
$\underline{S_{i,j-1} \Downarrow (u_j)\ S_{i,j+1}}$ (for both $i \in \{1,2\}$):
Then $H(j) \leq H(j-1)$. But the assumption of case 2 implies that the same height on both tracks cannot be maintained more than $2 \cdot N^3$ steps of derivation, hence $T(j) \leq H(j-1) \cdot (4 \cdot N^3)$.
$\underline{S_{i,j-1} \Uparrow (u_j)\ S_{i,j+1};\ \ S_{3-i,j-1} \Downarrow (u_j)\ S_{3-i,j+1}}$(for some $i \in \{1,2\}$):
The assumption of case 2 implies that, on track $3-i$, the same height cannot be maintained more than $2 \cdot N^3$ steps of derivation. Hence $T(j) \leq H(j-1) \cdot (2 \cdot N^3)$. It follows that $H(j) \leq H(j-1) \cdot (1 + 2 \cdot N^3)$.
This analysis shows that:

$$H(0) = 1;\ \ T(0) = 0;\ \ H(j) \leq H(j-1) \cdot (1 + 2 \cdot N^3);\ \ T(j) \leq H(j-1) \cdot (4 \cdot N^3).$$

It follows that $H(4k) \leq (1 + 2 \cdot N^3)^{4k}$ and

$$T(4k) \leq 4N^3 \sum_{j=0}^{4k-1} H(j) \leq 4N^3 (\sum_{j=0}^{4k-1} (1+2N^3)^j) \leq 4N^3 \cdot \frac{1+2N^3)^{4k}}{1+2N^3-1} \leq 2 \cdot (1+2N^3)^{4k}.$$

Hence we would have $|u_0| \leq T(4k) < N_1$, which is impossible. It follows that case 1 must occur, which achieves step 1 .

Step 2

Let $p \in P_{N_1}$ and $x \in X$. The equation p must have the form

$$p = (v_1 \odot u, v_2 \odot u) \text{ for some } u \in X^*, |u| \leq N_1.$$

If $|u| < N_1$, then $p \odot x = (v_1 \odot ux, v_2 \odot ux) \in P_{N_1}$. In this case $P_{N_1} \ \overset{*}{\vdash\!\!-}\ \{p \odot x\}$. Suppose that $|u| = N_1$. In step 1 we established that there exists some decomposition $u = u' \cdot u''$ such that $P_{|u'|-1} \ \overset{*}{\vdash\!\!-}\ \{(v_1 \odot u', v_2 \odot u')\}$. Applying now $|u''| + 1$ times lemma 43, we obtain that: $P_{|u|} \ \overset{*}{\vdash\!\!-}\ \{(v_1 \odot ux, v_2 \odot ux)\}$, i.e.

$$P_{N_1} \ \overset{*}{\vdash\!\!-}\ \{p \odot x\}.$$

□

Theorem 55 *There exists a constant $K \in \mathbb{N}$ such that, for every positive integer $k \geq 1$ and every strict-deterministic c.f grammar $G =< X, V, P >$ which is k-weighted and every $v_1, v_2 \in V$, it holds that, either $v_1 \equiv v_2$ or:*

$$\mathrm{Div}(\mathrm{L}(G, v_1), \mathrm{L}(G, v_2)) \leq K \cdot (2 \cdot \parallel G \parallel)^{12 \cdot k}.$$

Proof: Let us consider the subset P_{N_1}:
- either it is ε-consistent, and, by lemma 44, $v_1 \equiv v_2$;
- or it is not ε-consistent, which means that there exists some witness $u \in X^{\leq N_1}$ in the symmetric difference $\mathrm{L}(G, v_1) \Delta \mathrm{L}(G, v_2)$. □

Corollary 56 *For every positive integer $k \geq 1$, the equivalence problem for k-weighted strict-deterministic c.f. grammars (resp. for $2k-1$-turn deterministic pushdown automata) is in co-NP.*

Follows immediately from theorem 55 and theorem 52.

6 Extension, comparison, perspectives

6.1 Extension

Let us sketch some further applications of the same method leading to either improvements of the complexity or extension of the class of automata.

Finite-turn automata In fact the above method can be pushed further in order to obtain a polynomial upper-bound on the divergence of two weighted grammars with respect to the sum of the maximum weight of the variables and of the size of the grammars (i.e. number of variables, when they are assumed in normal form).

Theorem 61 *There exist integers $K_1, K_2 \in \mathbb{N}$ such that, for every c.f grammar $G =< X, V, P >$ which is k-weighted (for some $k \geq 0$) and strict-deterministic, and every $v_1, v_2 \in V$, it holds that, either $v_1 \equiv v_2$ or:*

$$\mathrm{Div}(\mathrm{L}(G, v_1), \mathrm{L}(G, v_2)) \leq K_1 \cdot (k + \parallel G \parallel)^{K_2}.$$

This result can be obtained by a combination of the arguments used for the weaker theorem 55 with some more elaborate "strategy" for constructing some self-proving set P. The general idea is to use a transformation similar to the transformations T_B^+, T_B^- used in [Sén97,Sén01]. Notice that these transformations behave correctly with respect to the deduction relation introduced in section 4.

Extended equivalence problem From the upper-bounds given in theorem 55, combined with the general extension scheme described in [Sén89], one can deduce upper-bounds for the divergence of a weighted grammar and a general strict-deterministic grammar.

Theorem 62 *There exists constants $K_3, K_4 \in \mathbb{N}$ such that, for every positive integer $k \geq 1$, for every strict-deterministic c.f grammar $G_1 =< X, V_1, P_1 >$ which is k-weighted, for every strict-deterministic c.f grammar $G_2 =< X, V_2, P_2 >$ and for every axioms $v_1 \in V_1, v_2 \in V_2$, it holds that, either $v_1 \equiv v_2$ or:*

$$\mathrm{Div}(\mathrm{L}(G_1, v_1), \mathrm{L}(G_2, v_2)) \leq K_3 \cdot 2^{K_4 \cdot (\|G_1\| + \|G_2\| + k)}.$$

As well, from theorem 61 and [Sén89] we obtain

Theorem 63 *There exists constants $K_5, K_6 \in \mathbb{N}$ such that, for every positive integer $k \geq 1$, for every strict-deterministic c.f grammar $G_1 =< X, V_1, P_1 >$ which is k-weighted, for every strict-deterministic c.f grammar $G_2 =< X, V_2, P_2 >$ and for every axioms $v_1 \in V_1, v_2 \in V_2$, it holds that, either $v_1 \equiv v_2$ or:*

$$\mathrm{Div}(\mathrm{L}(G_1, v_1), \mathrm{L}(G_2, v_2)) \leq K_5 \cdot (k + \parallel G_1 \parallel + 2^{\|G_2\|})^{K_6}.$$

6.2 Comparison

Let us compare our method and results with previous works on the same subject.

C1 Let us consider the sequence of the so-called "Zimin words", on a denumerable alphabet $\{A_1, \ldots, A_i, \ldots\}$, defined by:

$$Z_0 = \epsilon; \quad Z_{i+1} = Z_i \cdot A_{i+1} \cdot Z_i.$$

Let us set $B_i = A_i Z_{i-1}$ (for $i \geq 1$).
We then have:

$$Z_\lambda = B_1 B_2 \cdots B_\lambda$$

and the set of strict subwords

$$B_{i_1} B_{i_2} \cdots B_{i_p}$$

for $0 \leq p \leq \lambda - 1, 1 \leq i_1 < i_2 < \ldots < i_p \leq \lambda$ is exactly the set of strict suffixes

$$B_{\lambda-p+1} B_{\lambda-p+2} \cdots B_\lambda.$$

Therefore, the subwords lemma (lemma 32) implies the following. Consider a sequence of equations of the form

$$\alpha \hat{B}_{\lambda-p+1} \hat{B}_{\lambda-p+2} \cdots \hat{B}_\lambda S \equiv_n \beta \hat{B}_{\lambda-p+1} \hat{B}_{\lambda-p+2} \cdots \hat{B}_\lambda S, \tag{37}$$

where the words B_i are defined as above, and the matrices $\hat{B}_i$ are obtained by replacing the letters A_i by square matrices $\hat{A}_i \in \mathsf{D}\mathbb{B}_{\lambda,\lambda}\langle\langle V \rangle\rangle$. This set of equations implies the equation

$$\alpha \hat{B}_1 \hat{B}_2 \cdots \hat{B}_\lambda S \equiv_n \beta \hat{B}_1 \hat{B}_2 \cdots \hat{B}_\lambda S. \tag{38}$$

The statement that (37) $\Rightarrow$ (38) is the "extension theorem " of [Sti02, p.828]. In this way one can see the "subwords lemma" as an improvement of the "extension-theorem"[1]. Let us notice that the most immediate upper-bound for the complexity of the algorithm presented in [Sti02] is closely linked with the largest length $L(n,m)$ of a word avoiding Z_n on an alphabet of cardinality m (this integer exists because, by [Lot02, proposition 3.1.4], each Z_n is an unavoidable pattern; the function $L(n,m)$ seems to be a non-elementary function of the integer n, for any given $m \geq 2$).

C2 In [Bee76] it was already proved that, given G as in theorem 55, one can construct a simulator $\mathcal{N}$ of both grammars (in fact of the corresponding $2k-1$-turn automata) of size :

$$\text{size}(\mathcal{N}) \leq \parallel G \parallel^{c_4 \cdot U} \;\; \text{with } U = \mu^{c_1 \cdot n \cdot \mu^2}, \mu = \parallel G \parallel^2$$

(see in [Bee76]: size($\mathcal{N}$) p.318, line 6, definition of μ p.309, definition of U p.317, line 28). In this expression , n denotes the number of turns (p.308, line 10), hence here $n = 2k-1$. This bound , for a fixed $k \geq 1$, is a double exponential function of $\parallel G \parallel$.Therefore one can consider that this previous method gave:

$$\text{Div}(\text{L}(G, v_1), \text{L}(G, v_2)) \leq 2^{2^{2^{c_2 \cdot \parallel G \parallel}}},$$

and the equivalence-problem for k-turn dpda was established to be in

$$\text{DTIME}(2^{2^{c_1 \cdot \parallel G \parallel}})$$

Our theorem 55 and corollary 56 are thus improving these previous bounds from [Bee76].

C3 In [OI80] it was already proved that, given G_1, G_2 as in theorem 62, their divergence is upper-bounded by a tour of 4 exponentials and the extended equivalence-problem for k-turn dpda was established to be in

$$\text{DTIME}(2^{2^{2^{c_3 \cdot (\parallel G_1 \parallel + \parallel G_2 \parallel)}}}).$$

Our theorem 62 is thus improving these previous bounds from [OI80].

6.3 Perspectives

Let us sketch some perspectives for further investigations on the same topics.

1. Once we know the equivalence-problem for k-turn dpda is co-NP, we cannot escape the questions: is this problem co-NP-complete? is it NP? is it P?

[1] From a psychological point of view, the connection between these two statements must be oriented the other way round: it is the "extension theorem" that led the author to the idea of the "subwords lemma".

Let us mention that the equivalence problem for k-tape deterministic finite automata is known to be in co-NP (in fact co-NLIN) for every $k \geq 1$ ([HK91]). The particular case where $k = 2$ has been shown to be in P ([FG82]), and it can be seen also as a strict subcase of the equivalence problem for 1-turn dpda (see the discussion in [Bee76, pages 318-319]).

2. We are strongly convinced that the method presented here can be used to drop the complexity of equivalence algorithms for other sub-classes of d.p.d.a. It is also tempting to try to adapt the method to general d.p.d.a. (the best upper-bound known at the moment is that this problem is primitive recursive ([Sti02])).
3. The subwords lemma is likely to be extendable to series in $\mathsf{K}\langle\langle\ X\ \rangle\rangle$, for any field (or division-ring) of coefficients K. Therefore the same kind of method might be applicable to equivalence problems for deterministic pushdown transducers.
4. The subwords lemma deals, in fact, with *arbitrary prefix languages*; it might be useful for treating other equality problems for prefix languages, (e.g. for deterministic indexed languages: the languages recognized by deterministic pushdown of pushdown automata, see [Mas76]).

References

[Bee76] C. Beeri. An improvement on Valiant's decision procedure for equivalence of deterministic finite-turn pushdown automata. *TCS 3*, pages 305–320, 1976.

[FG82] E.P. Friedman and S.A. Greibach. A polynomial algorithm for deciding the equivalence problem for 2-tape deterministic finite state acceptors. *SIAM J. COMPUT., vol.11, No 1*, pages 166–183, 1982.

[Har78] M.A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley, Reading, Mass., 1978.

[HHY79] M.A. Harrison, I.M. Havel, and A. Yehudai. On equivalence of grammars through transformation trees. *TCS 9*, pages 173–205, 1979.

[HK91] T. Harju and J. Karhumäki. The equivalence problem of multitape finite automata. *Theoretical Computer Science 78*, pages 347–355, 1991.

[Lot02] Lothaire. *Algebraic combinatorics on words.* Cambridge University Press, 2002.

[Mas76] A.N. Maslov. Multilevel stack automata. *Problemi Peredachi Informatsii*, pages 55–62, 1976.

[OI80] M. Oyamaguchi and Y. Inagaki. The equivalence problem for two dpda's,one of which is a finite-turn or one-counter machine. *Information and Control 45*, pages 90–115, 1980.

[Sén89] G. Sénizergues. Church-Rosser controlled rewriting systems and equivalence problems for deterministic context-free languages. *Information and Computation, vol. 81, no. 3*, pages 265–279, 1989.

[Sén97] G. Sénizergues. The Equivalence Problem for Deterministic Pushdown Automata is Decidable. In *Proceedings ICALP 97*, pages 671–681. Springer, LNCS 1256, 1997.

[Sén98] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In Rajeev Motwani, editor, *Proceedings FOCS'98*, pages 120–129. IEEE Computer Society Press, 1998.

[Sén99] G. Sénizergues. T(A) =T(B)? In *Proceedings ICALP 99*, volume 1644 of *LNCS*, pages 665–675. Springer-Verlag, 1999. Full proofs in technical report 1209-99 of LaBRI,$T(A) = T(B)$?, pages 1-61.

[Sén01] G. Sénizergues. L(A) = L(B) ? decidability results from complete formal systems. *Theoretical Computer Science*, 251:1–166, 2001.

[Sén02] G. Sénizergues. L(A) = L(B) ? a simplified decidability proof. *Theoretical Computer Science*, 281:555–608, 2002.

[Sti99] C. Stirling. Decidability of dpda's equivalence. Technical report, Edinburgh ECS-LFCS-99-411, 1999. Pages 1-25.

[Sti01] C. Stirling. Decidability of dpda's equivalence. *Theoretical Computer Science*, 255:1–31, 2001.

[Sti02] C. Stirling. Deciding DPDAEquivalence is Primitive Recursive. In *Proceedings ICALP 02*, pages 821–832. Springer, LNCS 2380, 2002.

[Val74] L.G. Valiant. The equivalence problem for deterministic finite-turn pushdown automata. *Information and Control 25*, pages 123–133, 1974.