The Dot-Depth Hierarchy
v.
Iterated Block Products of DA

Bernd Borchert

WSI-2004-09

Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Arbeitsbereich Theoretische Informatik/Formale Sprachen
Sand 13
D-72076 Tübingen

borchert@informatik.uni-tuebingen.de

© WSI 2004
ISSN 0946-3852
The Dot-Depth Hierarchy v. Iterated Block Products of DA

Bernd Borchert
Universität Tübingen, Germany
email: borchert@informatik.uni-tuebingen.de

Abstract
Like the sequence of the classes of the dot-depth hierarchy the sequence of classes given by the $n$-fold iterated block product of DA has the class of starfree regular languages as its limit. It is shown that this DA-block-product hierarchy grows more slowly than the dot-depth hierarchy: in fact already $\Sigma_2^L$ of the dot-depth hierarchy contains properness witnesses for all levels of the DA-block-product hierarchy.

1 Introduction
The dot-depth hierarchy is a way to classify the complexity of starfree regular languages: the lower a starfree language sits in the dot-depth hierarchy the less complex it is supposed to be. But there exist alternative ways to classify the starfree languages which are only partially comparable with the dot-depth hierarchy, for example the until/since depth from temporal logic [TW04].

Another classification of the starfree languages is considered here: the hierarchy given by the $n$-fold iterated block product of DA. DA is the set of monoids corresponding as syntactic monoids to the languages in $\Sigma_2^L$ of the the dot-depth hierarchy, a very robust class with many characterizations [TT02].

The block product $\Box$ is also coming from the algebraic side and is the two-sided version of the wreath product on finite monoids, resp. on classes of monoids, see [RT89, ST02, TW04]. In this paper, DA and block products of DA will be identified with their corresponding language classes.

It is easy to see that the iterated block product $DA^{\Box^n}$ of DA, defined strongly bracketed as

$$DA^{\Box^n} := DA \Box (\ldots (DA \Box DA)),$$

is a subset of $\Delta_{n+1}^L$ of the dot-depth hierarchy, so the two hierarchies are in one direction comparable. It is also known that $\Delta_{n+1}^L$ contains languages from $DA^{\Box n}$ which are not in the full level $DD_n^L$ of the dot-depth hierarchy – this fact can be interpreted in the way that some parts of the DA-block-product hierarchy are growing as fast as the dot-depth hierarchy. The main result of this note is that other parts of the DA-block-product hierarchy are growing slowly compared with the dot-depth hierarchy: it is shown that already $\Sigma_2^L$ contains for every $n \geq 1$ witnesses of the properness of the inclusion $DA^{\Box n} \subset DA^{[n+1] \Box}$. A graphical summary of the results is sketched in Figure 2.
Figure 1: The dot-depth hierarchy

2 Preliminairies

The dot-depth of a starfree regular language counts the minimal nesting depth of concatenations (= “dot products”) one needs to represent the language by a starfree regular expression. There are two versions of the dot-depth hierarchy: the classical one by Cohen & Brzozowski [CH71] and the variant by Straubing and Thérien [St81, The81]. They only differ slightly, see [St94], i.e. the level $n + 1$ of one contains the level $n$ of the other. We consider in this paper only the second version, and we will use a logical characterization of its levels [Tho82, PP86]. The dot-depth hierarchy consists for every $n \geq 0$ of the classes $\Sigma^n, \Pi^n, \text{DD}_n^n$, and $\Delta_n$, each of which is formally a mapping from the sets of finite alphabets to a set of regular languages over this alphabet. The class $\Sigma_n$ is, according to a characterization of Thomas [Tho82] and Perrin & Pin [PP86], the set of languages definable with a $\Sigma_n$ alternation prefix in first-order logic on words with the signature $[\prec]$ plus a unary predicate for each letter of the respective alphabet, see [St94, PW97]. $\Pi_n$ is the set of complements of languages in $\Sigma_n$, $\text{DD}_n^n$ (usually called $L_n$) is the Boolean closure of $\Sigma_n$, and $\Delta_n$ is defined as $\Sigma_n \cap \Pi_n$. It holds the proper inclusions as depicted in Figure 1, see for example [St94, PW97].

The syntactical monoid $M_L$ of a language $L$ over alphabet $A$ consists of the equivalence classes $[u]$ for $u \in A^*$ defined by the the equivalence relation

$$[u] = [v] \iff \forall w, z \in A^* : wuz \in L \iff wuz \in L.$$ (1)
The monoid operation can be defined by $[u][v] := [w]$, especially it holds for all words $u, v, w, z$ from $\Sigma^*$:

$$
\text{if } [u] = [v] \text{ then } [wuz] = [wvz].
$$

(2)

A language is regular iff its syntactical monoid is finite, and it is starfree iff moreover there exists a number $\omega$ such for all $x \in A^*$ it holds

$$
[x^\omega] = [x^n x^n] \text{ for every } n \geq 0.
$$

(3)

The class of monoids $DA$, which naming letters stand for the algebraic notions “D-classes” and “aperiodic”, is the algebraic pendant of the language class $\Delta^2$ from the dot-depth hierarchy, in the sense that a language $A$ is in $\Delta^2$ if and only if its syntactical monoid $M_A$ is in $DA$, see for example [PW97, TT02]. By this correspondence, and because this paper tries to stay on the language side only, $DA$ will stand for $\Delta^2$ from now on. The following characterization of $DA$, which is very close to the algebraic definition of $DA$, see [TT02], will be used extensively.

**Lemma 1 (DA)** A language $L$ over alphabet $\Sigma$ belongs to $DA$ iff for all words $x, y, z \in \Sigma^*$ it holds in $M_L$:

$$
[((xyz)^{\omega} y (xyz)^{\omega} )] = [(xyz)^{\omega}].
$$

(4)

For the definition of the block product we also stay on the language side (besides a little dip into the syntactic monoid), see [TW04].

**Definition 1 (block product)** The block product $K \Box J$ of a language $J$ over alphabet $\Sigma$ and a language $K \in DA$ over alphabet $M_J \times \Sigma \times M_J$ (where $M_J$ is the syntactic monoid of $J$) is the language over alphabet $\Sigma$ consisting of all words $x = x_1 \cdots x_n$ in $\Sigma^*$ such that the following word $\tau(x)$ is in $K$:

$$
\tau(x) := ([\varepsilon], x_1, [x_2 \cdots x_n]) ([x_1], x_2, [x_3 \cdots x_n]) \cdots ([x_1 \cdots x_{n-1}], x_n, [\varepsilon]).
$$

(5)

The block product $K \Box J$ of two classes of languages $K$ and $J$ is the set of block products $K \Box J$ such that $K \in K$ and $J \in J$.

The block product is in general not associative, see for example [ST02]. Therefore, we have two extrem cases (and many in between) concerning the bracketing: The strongly iterated block product of $n$ languages $K_n, \ldots, K_1$ (we prefer them to be numbered from the right) is defined as

$$
K_n \Box (K_{n-1} \Box (\ldots (K_2 \Box K_1)\ldots))
$$

while the $n$-fold weakly iterated block product is defined as

$$
((\ldots (K_n \Box K_{n-1})\ldots) \Box K_2) \Box K_1.
$$

Let $DA^{\Box n}$ be the set of all $n$-fold strongly iterated block products of DA languages. It holds that every weakly iterated block product of DA languages is in $DA^{\Box n}$, see for example [ST02], likewise every other bracketing of an $n$-fold block product of DA languages results in a language contained
in $\text{DA}^{n[]}$, This justifies that we speak of $\text{DA}^{n[]}$ as the $n$-fold iterated block product of $\text{DA}$, without mentioning the strong bracketing.

The class $\text{DA}$ and every block product expression built on it, like $\text{DA}^{n[]}$, is a variety of languages, i.e. it is closed under Boolean operations, under left and right quotients and under inverse homomorphic images, see [Pin86, ST02].

We state the following facts about the relation of $\text{DA}^{n[]}$ and the dot-depth hierarchy. They can be derived from results in the literature, the proofs below are only sketched.

**Theorem 1** Let $n \geq 1$.
(a) $\text{DA}^{n[]} \subseteq \Delta_{n+1}^L$,
(b) $\text{DA}^{n[]}$ contains languages in $\Delta_{n+1}^L - \text{DD}_n^L$,
(c) $\bigcup_{n \geq 1} \text{DA}^{n[]} = \text{equal the class of starfree languages.}$

**Proof.** (a) For $n = 1$ this holds by definition. For the induction consider a language $L$ in $\text{DA}^{(n+1)[]}$, i.e. $L = L_1 \sqcap L_0$ with $L_1 \in \text{DA}$ and $L_0 \in \text{DA}^{n[]}$. In order to get a $\Sigma_{n+2}$ expression for $L$ take the $\Sigma_2$ expression for $L_0$ and plug the $\Pi_{n+1}$ expression for $L_1$, which exists by induction hypothesis, into it. The two $\forall$ levels collapse and in total it is a $\Sigma_{n+2}$ expression. In order to get a $\Pi_{n+2}$ expression for $L$ plug the $\Sigma_{n+1}$ formula for $L_1$ into the $\Pi_2$ expression for $L_0$. This shows $L \in \Sigma_{n+2}^L \cap \Pi_{n+2}^L = \Delta_{n+2}^L$.

(b) Consider for $n \geq 2$ two following languages $D_n$ on alphabet $\{0, 1, \ldots, 2n - 2\}$, see [BL+04]: $D_2 = 0^*1\{0, 1, 2\}^*$, and for $n \geq 3 D_n$ consists of the words $w$ such that the occurrences of the letters $2n - 3$ and $2n - 2$ in $w$ are considered as markers, and $w$ is in $D_n$ iff the marker after the first factor between two such markers which is in $D_{n-1}$ is $2n-1$. $D_n$ is not only in $\Delta_n$, as it is argued in [BL+04], but even in $\text{DA}^{(n-1)[]}$. And moreover (thanks to Klaus Wagner, Würzburg, for this hint), $D_n$ can be shown to be not in $\text{DD}_n^L$ by the result of [Tr02, BL+04] that $\text{Leaf}^{P}(D_n) = \Delta_n^P$, together with the oracle result separating the levels of PH and the relativizable result that PH collapses if BH collapses.

(c) Part (a) above verifies that each $\text{DA}^{n[]}$, and therefore the limit of this sequence, consists of starfree languages only. On the other hand every starfree language $L$ is covered by some $\text{DA}^{n[]}$: let $\phi$ be a first order formula for $L$, which exists by the classical result starfree = first-order definable of McNaughton & Papert [MP71]. Then the quantifier depth (n.b.: not the quantifier alternation depth) of $\phi$ is such an $n$: each nested quantifier can be simulated by a $\text{DA} [] \ldots$ operation (actually, by a $\text{DD}_1^L \sqcap \ldots$ operation). q.e.d.

Note that by the results of Theorem 1 it still could be the case that for example $\text{DA}^{n[]} = \Delta_{n+1}^L$ for all $n \geq 1$, or that $\text{DA}^{n[]} = \text{a class in between } \Sigma_2^L \text{ and } \Delta_{n+1}^L$, or that a similar close relation to the dot-depth hierarchy holds. In the following section it is shown that this is not the case.

### 3 $\Sigma_2^L$ is not contained in an iterated block product of $\text{DA}$

The following languages $L_n$, for $n \geq 2$, over alphabet $\Sigma_n := \{1, \ldots, n\}$ are from $\Sigma_2^L$ and will be shown to be witnesses for the properness of the inclusion $\text{DA}^{(n-1)[]} \subset \text{DA}^{n[]}$.\[L_2 = \{1, 2\}^*11\{1, 2\}^*, \quad (6)\]
\[ L_{n+1} := \Sigma^*_{n+1} L_n \Sigma^*_{n+1}. \]  

(7)

where \( L_n \) is considered as a language over the larger alphabet \( \Sigma_{n+1} \). For example,

\[ L_3 = \{1, 2, 3\}^*11\{1, 2\}^*11\{1, 2, 3\}^* \]

(because \( \{1, 2, 3\}^*\{1, 2\}^* = \{1, 2, 3\}^* \) etc.), and

\[ L_4 = \{1, 2, 3, 4\}^*11\{1, 2\}^*11\{1, 2, 3\}^*11\{1, 2, 3, 4\}^*. \]

(With some fantasy the reader can see overlapping waves in these languages.) These examples show that \( L_n \) can also described as \( L_n = \Sigma^* M_n \Sigma^*_n \) where \( M_n \) is defined via the following recursion:

\[ M_2 = 11, \]

(8)

\[ M_n = M_{n-1} \Sigma^*_n M_{n-1}. \]

(9)

**Theorem 2 (Main)** For every \( n \geq 2 \) it holds: The language \( L_n \) is an element of \( \Sigma^*_2 \cap \text{DA}^{n\Box} \) but not of \( \text{DA}^{(n-1)\Box} \).

This theorem is the conjunction of the following Lemma 2, Corollary 1, and Lemma 6, which will be proven now, using more sub-lemma.

A **marked product of sub-alphabets** over an alphabet \( A \) is a regular expression

\[ A_0 a_1 A_1 \ldots a_n A_n \]

with \( n \geq 0, a_0, \ldots, a_n \) “markers” = letters from \( A \), and \( A_0, \ldots, A_n \) sub-alphabets, i.e. subsets of \( A \). Example: \( \{0, 1, 2\}^*20^*2\{0, 1, 2\}^* \) expressing “there exists two 2’s with no 1’s between them”. It is easy to see that a language described by a marked product of sub-alphabets is in \( \Sigma^*_2 \), and in fact, by the results of Arfi [Ar87], \( \Sigma^*_2 \) equals the set of all finite unions of them.

**Lemma 2** For every \( n \geq 2 \) it holds: The language \( L_n \) is an element of \( \Sigma^*_2 \).

**Proof.** Every \( L_n \) (for \( n \geq 2 \)) is by the representation \( \Sigma^*_n M_n \Sigma^*_n \) a marked product of sub-alphabets: \( M_2 = 10^*1 \) is a marked product of sub-alphabets with two outmost markers 1, and \( M_{n+1} = M_n \Sigma^*_n M_n \) keeps its two outmost markers 1. q.e.d.

**Lemma 3** For every \( n \geq 1 \) it holds: Any language described by a marked product of sub-alphabets with at most \( 2^n - 1 \) markers is in \( \text{DA}^{n\Box} \).

**Proof.** Induction start \( n = 1 \). A marked product \( A_0 a_1 A_1 \) is in \( \Sigma^*_2 \), see above. On the other hand, \( A_0 a_1 A_1 \) can be expressed by the following \( \Pi_2 \) expression “there exists a position carrying letter \( a_1 \), and all positions carry letters from \( A_0 \cup A_1 \cup \{a_1\} \), and it never occurs that a position has a letter from \( A_1 - (A_0 \cup \{a_1\}) \) and larger position has a letter from \( A_0 - (A_1 \cup \{a_1\}) \), and between every two positions with a letter from \( A_0 - (A_1 \cup \{a_1\}) \) and a letter from \( A_1 - (A_0 \cup \{a_1\}) \) there is a position in between carrying letter \( a_1 \)”. This shows that \( A_0 a_1 A_1 \) is in \( \Sigma^*_2 \cap \Pi_2^* = \Delta^*_2 \). Induction step for \( n \geq 2 \). Given a marked product \( L = A_0 a_1 A_1 \ldots a_m A_m \) over alphabet \( A \) with \( m \leq 2^n - 1 \), let \( a_k \) be the marker in the middle of the expression, i.e. \( k = m/2 \) if \( m \) is odd and \( k =
\[(m + 1)/2 \text{ if } m \text{ is even. Then } L = L_0 a_k L_1 \text{ with } L_0 = A_0 a_1 A_1 \ldots a_{k-1} A_{k-1} \text{ and } L_1 = A_k \ldots a_m A_m,\]
and both \(L_0\) and \(L_1\) are marked products of sub-alphabets with not more than \(2^{n-1} - 1\) markers. Therefore, the induction hypothesis applies to \(L_0\) and \(L_1\), i.e. both \(L_0\) and \(L_1\) are in \(\text{DA}^{(n-1)}\). Let \(P := L_0 \times L_1\) be their product language which is by the variety closure properties still an element of \(\text{DA}^{(n-1)}\). Let \(Q\) be the \(\Sigma^*_1\) language consisting of the union of the languages \(B^*(p, a_k, q)B^*\) on the alphabet \(B = M_P \times A \times M_P\) such that \(p\) stands for acceptance of \(L_0\) and \(q\) for acceptance of \(L_1\). The language \(Q \square P\) is by this representation from \(\text{DA}^{n\square}\) and equals \(L\). q.e.d.

Because \(L_n\) has \(2^{n-1}\) markers (the 1’s) we have the following corollary.

**Corollary 1** For every \(n \geq 2\) it holds: \(L_n\) is in \(\text{DA}^{n\square}\).

It remains to prove that \(L_n\) is not in \(\text{DA}^{(n-1)\square}\). Assume that \(L_n\) equals a language \(K\) from \(\text{DA}^{(n-1)\square}\), i.e.

\[K := K_{n-1} \square (\ldots (K_2 \square K_1))\]  

(10)

where each \(K_i\) is in \(\text{DA}\). We will specify two words \(u_n, v_n\) such that \(u_n \notin L_n\) and \(v_n \in L_n\) but \(u_n\) and \(v_n\) are indistinguishable by \(K\), i.e. \(u_n \in K \iff v_n \in K\).

Define \(u_n\) and \(v_n\) for \(2 \leq n\) by induction:

\[u_2 = (21)^\omega\]  

(11)

\[v_2 = (21)^\omega 1(21)^\omega\]  

(12)

where \(\omega\) is the constant from Lemma 1 for \(K_1\). For \(n \geq 3\) define the abbreviation \(w_n\), and \(u_n, v_n\) the following way:

\[w_n = u_{n-1} n u_{n-1} v_{n-1}\]  

(13)

\[u_n := \begin{array}{cccc}
\omega & w_n & w_n^\omega & w_n^\omega \\
I & II & III & IV
\end{array}\]  

(14)

\[v_n := \begin{array}{cccc}
\omega & w_n & v_{n-1} & w_n^\omega \\
I & II & IIa & III
\end{array}\]  

(15)

where \(\omega\) is the constant from Lemma 1 for \(K_{n-1}\) (no indexing of \(\omega\) necessary, it will be clear from context which one is meant).

We show that \(u_n \notin L_n\) and \(v_n \in L_n\) via the following stronger invariant.

**Lemma 4** Consider a word \(g = g_1 \cdots g_m\) where each \(g_i\) is either \(u_n\) or \(v_n\). The factors of \(g\) which are elements of \(M_n\) are the following: exactly one such factor within each of the \(g_i\) for which \(g_i = v_n\).
Proof. For \( n = 2 \) the lemma can be checked easily. Let \( n \geq 3 \) and consider a word \( g \) from \( \{u_n, v_n\}^* \). Because \( M_n \) does not use the letter \( n \), a potential factor of \( g \) which is in \( M_n \) can only be found in the parts \( u_{n-1}v_{n-1}u_{n-1} \) and \( u_{n-1}v_{n-1}u_{n-1}v_{n-1}u_{n-1} \), the latter occurring within the \( v_n \)'s of \( g \). The parts \( u_{n-1}v_{n-1}u_{n-1} \) contain by induction hypothesis only one factor which is from \( M_{n-1} \). By \( M_n = M_{n-1} \Sigma_n^{-1} M_{n-1} \) we need two factors from \( M_{n-1} \) for a word in \( M_n \). Therefore, these parts \( u_{n-1}v_{n-1}u_{n-1} \) do not contain a factor from \( M_n \), what proves one part of Lemma 4 for this \( n \). The parts \( u_{n-1}v_{n-1}v_{n-1}u_{n-1} \) contain by induction hypothesis exactly 2 factors of a word from \( M_{n-1} \). Therefore these two factors together with the word in between build a factor belonging to \( M_n = M_{n-1} \Sigma_n^{-1} M_{n-1} \), and this is the only such factor. The parts \( u_{n-1}v_{n-1}v_{n-1}u_{n-1} \) are the parts corresponding to the the occurrences of \( v_n \) in \( g \). Therefore, Lemma 4 holds also for this \( n \). \textbf{q.e.d.}

**Corollary 2** For every \( n \geq 2 \) it holds: \( u_n \notin L_n, v_n \in L_n \).

**Proof.** From Lemma 4 it follows that for \( g = g_1 = u_n \) there is no occurrence of a factor from \( M_n \), therefore \( u_n \) is not contained in \( L_n = \Sigma^* M_n \Sigma^* \), while for \( g = g_1 = v_n \) is there an (actually, exactly one) occurrence of a factor from \( M_n \), therefore \( v_n \) is contained in \( L_n = \Sigma^* M_n \Sigma^* \). \textbf{q.e.d.}

We will proof by induction the following crucial invariant.

**Lemma 5** For \( n \geq 2 \) it holds in the syntactic monoid of \( K = K_{n-1} \sqcap \ldots (K_2 \sqcap K_1) \ldots \) the following:

\[
[v_n] = [u_n] = [u_n u_n] = [v_n v_n] = [u_n v_n] = [v_n u_n].
\] (16)

**Proof.** Induction start: In case \( n = 2 \) the block product \( K = K_1 \) is a single DA language. In order to verify the first of the equations in 16 note that \([v_2] = [(21)^\omega 1(21)^\omega] = [(21)^\omega 1(21)^\omega] = [u_2 u_2]\) by equation 4 in Lemma 1 setting \( x := 2, y := 1, z := \varepsilon \). Moreover, \([u_2] = [(21)^\omega] = [(21)^\omega 1(21)^\omega] = [u_2 u_2]\) by equation 3. The other equations follow immediately from these two by equation 2.

Induction step for \( n \geq 3 \) : Define \( J := K_{n-2} \sqcap \ldots (K_2 \sqcap K_1) \ldots \), this way \( K = K_{n-1} \sqcap J \). We go to the definition of the block product \( K_{n-1} \sqcap J \), and will analyze the words \( \tau(z u_n z') \) and \( \tau(z v_n z') \), see equation 5 in Definition 1. \( z \) and \( z' \) are two arbitrary words from \( \Sigma_n \), we need them later in order to show that from \( [\tau(z u_n z')] = [\tau(z v_n z')] \) in the syntactic monoid of \( K_{n-1} \) it follows \([u_n] = [v_n]\) in the syntactic monoid of \( K_{n-1} \sqcap J \). Note that \( \tau(z u_n z') \) and \( \tau(z v_n z') \) are words on alphabet \( M_J \times \Sigma \times M_J \) which have the same length as \( z u_n z' \) and \( z v_n z' \), respectively, so we can keep the partition of the positions of \( u_n \) and \( v_n \) into the parts I to IV, as in equations 14 and 15, plus two parts 0 and V for the positions of \( z \) and \( z' \), respectively. We will show that there exist words \( p_0, p, x, y, s, s_0 \) over alphabet \( M_J \times \Sigma \times M_J \) such that \( \tau(z u_n z') \) and \( \tau(z v_n z') \) can be written the following way:

\[
\tau(z u_n z') = \tau(0 \ W_{n}^{w} \ W_{n}^{w} \ W_{n}^{w} \ W_{n}^{w} \ z') = p_0 \ p_0^{x} \ (xy)^{y} \ s \ s_0
\] (17)

\[
\tau(z v_n z') = \tau(0 \ W_{n}^{w} \ W_{n}^{w} \ v_{n}^{w} \ W_{n}^{w} \ W_{n}^{w} \ z') = p_0 \ p_0^{x} \ (xy)^{y} \ s \ s_0
\] (18)

To verify the above three equations 17 and 18 we have to show the following:
(a) \( \tau(zu_nz') \) and \( \tau(zw_nz') \) coincide on parts 0, I, II, III, IV and V.

(b) There exists a word \( h (= xy) \) such that the two restrictions of \( \tau(zu_nz') \) to parts II and III are of the form \( h^w \).

(c) This periodic pattern \( h \) from (b) has a suffix \( y \) which equals \( \tau(zw_nz') \) restricted to part IIa.

ad (a): We show that the words \( \tau(zu_nz') \) and \( \tau(zw_nz') \) coincide on parts 0, I, II, III, IV, and V: Let \( i \) be a position in part 0, I, or II of the words \( zu_nz' = b_1 \ldots b_m \) and \( zw_nz' = b'_1 \ldots b'_m. \) The two triples \( ([b_1 \ldots b_i], [b_i \ldots b_m]) \) at position \( i \) of \( \tau(zu_nz') \) and \( ([b'_1 \ldots b'_i], [b'_i \ldots b'_m]) \) at position \( i \) of \( \tau(zw_nz') \) will of course coincide on their left and middle component because \( zu_nz' \) and \( zw_nz' \) are identical up to that position. But moreover they also coincide on the right component of the triple: The two words \( b_{i+1} \ldots b_m \) and \( b'_{i+1} \ldots b'_m \) only differ by the extra factor \( v_{n-1} \) in \( b'_{i+1} \ldots b'_m \) from part IIa. But this \( v_{n-1} \) is immediately left to a \( u_{n-1} \) \( (u_{n-1} \) is a prefix of part III), and by induction hypothesis we have \( [v_{n-1}u_{n-1}] = [u_{n-1}] \) in the syntactic monoid of \( J \). Therefore, by equation 2, \( [b_{i+1} \ldots b_m] = [b'_{i+1} \ldots b'_m], \) i.e. the third components of the two tripels are also equal. By symmetrical arguments and \( [v_{n-1}v_{n-1}] = [v_{n-1}] \) by induction hypothesis we have that \( \tau(zu_nz') \) and \( \tau(zw_nz') \) also coincide on parts III, IV, and V.

ad (b): Let \( i \) be a position in the \( j \)-th factor \( w_n \) \((1 \leq j \leq \omega) \) of part II of \( zu_nz' \). Then the triple of \( \tau(zu_nz') \) at that position \( i \) has the form

\[
([w_n^j w_n^{j-1} f], a,[g w_n^{\omega-j} w_n^\omega z'])
\]

where \( f \) and \( g \) are the prefix and the suffix of the factor \( w_n \) left and right of that position \( i \), respectively, i.e. \( f ag = w_n \). Note that by equation 3 it holds \( [w_n^j w_n^{j-1} f] = [w_n^\omega] \) in the syntactic monoid of \( J \), so we can by equation 2 rewrite the left component as \( [w_n^\omega f] \). Likewise (now via adding \( w_n^{j-1} \) instead of dropping it) the right component can be rewritten as \( [g w_n^{\omega-j} w_n^\omega z'] \). This way we have at the position \( i \) in the \( j \)-th factor \( w_n \) of part II of \( \tau(zu_nz') \) the triple

\[
([w_n^\omega f], a,[g w_n^{\omega-j} w_n^\omega z']).
\]

But this is exactly the same triple as the triple at the \( i \)-th position of the first factor \( w_n \) in part II of \( \tau(zu_nz') \). By setting \( h \) to be the suffix of length \( |w_n| \) of part II of \( \tau(z' w_n z) \) we get the desired property (b) for part II. By symmetrical arguments (b) also holds for part III.

ad (c): Consider a position \( i \) in part IIa, i.e. \( v_n = b_1 \cdots b_{i-1} b_i b_{i+1} \cdots b_m \). The triple at the \( i \)-th position in part IIa of \( \tau(zw_nz') \) will be

\[
([w_n^\omega w_n^{\omega-1} u_{n-1} v_{n-1} b_1 \cdots b_{i-1}], [b_{i+1} \cdots b_m u_{n-1} v_{n-1} w_n^{1-w_n} w_n^\omega]),
\]

By induction hypothesis it holds \( [u_{n-1} v_{n-1}] = [v_{n-1}] \) in the syntactic monoid of \( J \), therefore the first component the factor \( u_{n-1} v_{n-1} \) left of \( b_1 \) can be rewritten by \( u_{n-1} \), and likewise in the third component the factor \( u_{n-1} \) right of \( b_m \) can be rewritten by \( v_{n-1} u_{n-1} \), as this is indicated by the underlinings in the triples above and below. This way the above triple equals

\[
([w_n^\omega w_n^{\omega-1} u_{n-1} v_{n-1} b_1 \cdots b_{i-1}], [b_{i+1} \cdots b_m v_{n-1} u_{n-1} v_{n-1} w_n^{1-w_n} w_n^\omega z']).
\]

But this is exactly the triple which one gets by looking at the \( i \)-th position in the suffix \( v_{n-1} \) of part II of the word \( \tau(zv_nz') \).
We have shown (a), (b), and (c), i.e. $\tau(zu_n z')$ and $\tau(zv_n z')$ can be written in the form of equations 17
and 18. This gives the following equation 19 in the syntactic monoid of $K_{n-1}$:

$$\tau(zu_n z') = \left[ \begin{array}{cccccc}
0 & p & (xy)^\omega & (xy)^\omega & s & s_0 \\
\phantom{0} & \phantom{p} & \phantom{(xy)^\omega} & \phantom{(xy)^\omega} & \phantom{s} & \phantom{s_0} \\
0 & I & I & I & IIa & III \\
\end{array} \right]$$

(19)

The middle equation symbol above holds by the following equality in the syntactic monoid of $K_{n-1}$
which is a case of equation 4 (no renaming of the variables $x, y$ necessary, $z := \varepsilon$):

$$[\begin{array}{cc}
II & III \\
\end{array} (xy)^\omega (xy)^\omega] = [\begin{array}{cc}
II & III \\
\end{array} (xy)^\omega (xy)^\omega]$$

(20)

We have shown $[\tau(zu_n z')] = [\tau(zv_n z')]$ in the syntactic monoid of $K_{n-1}$ for all words $z, z' \in \Sigma_n^*$. From this it follows $\tau(zu_n z') \in K_{n-1} \iff \tau(zv_n z') \in K_{n-1}$ for all $z, z' \in \Sigma_n^*$. This means, by the
definition of block product: $zu_n z' \in K_{n-1} \square J \iff zv_n z' \in K_{n-1} \square J$ for all $z, z' \in \Sigma_n^*$. By the
definition of the elements of the syntactic monoid we have the equality

$$[u_n] = [v_n]$$

(21)

in the syntactic monoid of $K_{n-1} \square J$.

This shows that the first equation in Lemma 5 holds. Now we show the second equation $[u_n u_n] = [u_n]$. Let $z, z'$ be again some words from $\Sigma_n^*$. Let $\tau$ again be the function in equation 5 in the definition
of block product. It holds for $\tau(zu_n u_n z')$ the following:

$$\tau(zu_n u_n z') = \tau(z u_n^{\omega_1} u_n^{\omega_2} u_n^{\omega_3} u_n^{z'} = \left[ \begin{array}{cccccc}
0 & p & (xy)^{3\omega} & (xy)^{3\omega} & s & s_0 \\
\phantom{0} & \phantom{p} & \phantom{(xy)^{3\omega}} & \phantom{(xy)^{3\omega}} & \phantom{s} & \phantom{s_0} \\
0 & I & I & I & I & IVa \\
\end{array} \right]$$

(22)

The first equality is the definition of $u_n$, the second equality holds by the same argumentation like for claim (a) above. In the syntactic monoid of $K_{n-1}$ it holds by equation 3 $[(xy)^{3\omega}] = [(xy)^{\omega}]$. Therefore, and by equations 22 and 17 together with equation 2, it holds in the syntactic monoid of $K_{n-1}$:

$$\tau(zu_n u_n z') = \left[ \begin{array}{cccccc}
0 & p & (xy)^{3\omega} & (xy)^{3\omega} & s & s_0 \\
\phantom{0} & \phantom{p} & \phantom{(xy)^{3\omega}} & \phantom{(xy)^{3\omega}} & \phantom{s} & \phantom{s_0} \\
0 & I & I & I & I & IVa \\
\end{array} \right]$$

(23)

From $[\tau(zu_n u_n z')] = [\tau(zu_n z')]$ in the syntactic monoid of $K_{n-1}$ for all $z, z' \in \Sigma_n^*$ we can like above conclude that in the syntactic monoid of $K_{n-1} \square J$ it holds:

$$[u_n u_n] = [u_n]$$

(24)

We have shown $[u_n] = [v_n]$ and $[u_n] = [u_n u_n]$ in the syntactic monoid of $K_{n-1} \square J$. The other equations follow immediately from these two by equation 2. q.e.d.

**Lemma 6** For every $n \geq 2$ it holds: $L_n$ is not an element of $DA^{(n-1)\Box}$. 

9
Proof. Let $n \geq 2$ and consider $L_n$ as a language over alphabet $\Sigma_n$. Assume that $L_n$ is in $\text{DA}^{(n-1)\Box}$. Then there exist $n-1$ languages $K_{n-1}, \ldots, K_1$ all of them from DA such that for $K = K_{n-1} \Box (\ldots (K_2 \Box K_1) \ldots)$ it holds $L_n = K$. By Corollary 2, $u_n \in L_n$ and $v_n \notin L_n$. But on the other hand, by Lemma 4, it holds $[u_n] = [v_n]$ in the syntactic monoid of $K$, from which it follows $u_n \in K \iff v_n \in K$, i.e., $u_n$ and $v_n$ are indistinguishable in $K$. Therefore, $L_n$ cannot be equal to $K$. It follows that $L_n$ cannot be from $\text{DA}^{(n-1)\Box}$. q.e.d.

From Theorems 1 and 2 we can conclude:

**Corollary 3** Let $n \geq 1$ and $k \geq 2$. If $n < k$ then each of the four classes $\Sigma^L_k$, $\Pi^L_k$, $\text{DD}^L_k$, and $\Delta^L_{k+1}$ contains $\text{DA}^{n\Box}$ properly. If $n \geq k$ then each of these four classes is incomparable with $\text{DA}^{n\Box}$.

Figure 2 gives a visual summary of the results in Theorems 1 and 2, and Corollary 3.

4 Open Questions and Acknowledgements

A problem left open is whether the weakly and the strongly bracketed $n$-fold iterated block product of DA coincide. Another interesting question is whether the class $\text{DA} \Box \text{DA}$ or at least $(\text{DA} \Box \text{DA}) \cap \Sigma^L_2$ is decidable. By the results of Arfi [Ar87] the latter question can be reduced to the decidability of the following computational problem: Given a marked product $A_0a_1A_1\ldots a_nA_n$ of sub-alphabets, does it belong to $\text{DA} \Box \text{DA}$?

The author is grateful to Pascal Tesson for many discussions on the subject.

**References**


Figure 2: $\Sigma_2^L$ v. iterated block products of DA.

$DA = \Delta_2^L$


