A Note on Multidimensional Dyck Languages

Michael Moortgat

Utrecht Institute of Linguistics
M.J.Moortgat@uu.nl

Abstract. Multidimensional Dyck languages generalize the language of balanced brackets to alphabets of size > 2. Words contain each alphabet symbol with the same multiplicity. In addition, reading a word from left to right, there are always at least as many a_i as $a_{(i+1)}$, where a_j is the jth alphabet symbol in the lexicographic ordering.

We compare the Dyck languages with MIX languages, where the multiplicity constraint is respected, but the order of the symbols is free. To understand the combinatorics of the Dyck languages, we study the bijection with standard Young tableaux of rectangular shape, and, for the three-dimensional case, with planar webs for combinatorial spider categories. We present a typelogical analysis of Dyck languages with an alphabet of size d in terms of a construction that aligns $(d-1)$ grammars for the two-symbol case.

This paper is dedicated to Jim Lambek on the occasion of his 90th birthday. More than 50 years ago, the deductive method of his 'Mathematics of Sentence Structure' raised the standards for doing computational linguistics to a new level. His work has been a continuing source of inspiration.

1 Introduction

Typelogical grammar, ever since the rediscovery of Lambek's Syntactic Calculus in the 1980s, has been struggling to come to grips with *discontinuity*: information flow between physically detached parts of an utterance. Emmon Bach, in a number of papers [1,2], discussed generalizations of categorial grammar that would be able to deal with 'scramble' languages: languages that are the free permutations of arbitrary context-free languages. As an abstract example of a language with total word order freedom, Bach proposed MIX: the words of this language are strings over a three-letter alphabet with an equal number of occurrences of each letter but no constraints on their order. MIX, in other words, is the scramble of context-free (actually, regular) $(abc)^+$.

A natural generalization of MIX allows variation in the size of the alphabet, keeping the multiplicity constraint on the letters. Let us call the size of the alphabet the *dimension* of the language. For positive d and n, let us write M_n^d for the (finite) language consisting of words of length dn over an alphabet with d symbols where each symbol occurs with multiplicity n.

$$M_n^d = \{w \in \{a_1, \dots, a_d\}^+ \mid |w|_{a_1} = \dots = |w|_{a_d} = n\}$$

Dropping the subscript, we write M^d for $\bigcup_{0<n} M^d_n$.

The restriction to positive multiplicity is characteristic of a categorial view on these languages: categorial grammars, being fully lexicalized, do not recognize the empty word. In the context of rewriting grammars, it is natural to also allow $n = 0$. In the degenerate case of $d = 1$, the language consists of the nth powers of the single alphabet symbol; there is nothing to scramble then.

I this paper, I consider a similar multidimensional generalization for Dyck languages. These are d-dimensional MIX languages with an extra prefix constraint: reading a word from left to right, there are always at least as many letters a_i as a_{i+1}, where a_j is the j-th symbol in the lexicographic order of the alphabet[1]

$$D^d_n = \{w \in M^d_n \mid \text{ for every prefix } u \text{ of } w, |u|_{a_i} \geq |u|_{a_{i+1}}, 1 \leq i < d\} \qquad (1)$$

As before, we write D^d for $\bigcup_{0<n} D^d_n$. In the two-dimensional case D^2, we have the familiar language of well-balanced brackets consisting of words over an alphabet $\{\mathsf{a}, \mathsf{b}\}$ (with $\mathsf{a} < \mathsf{b}$) reading a as the opening and b as the closing bracket symbol. Whereas MIX stands for total word order freedom, the generalized Dyck languages represent discontinuity of the *interleaving* type: words of D^d_n merge n copies of the word $a_1 \cdots a_d$ in such a way that the order of the letters is respected. Interleaving (or shuffling) has been proposed for the analysis of semifree word order and bounded discontinuous constituency, for example in the 'sequence union' operation of [21]. Complexity of parsing of extended context-free formalisms with interleaving and concatenation operations has been studied in [18]. For applications of interleaving in various areas of computer science, see [3].

As for the position of M^d and D^d in the extended Chomsky hierarchy, much remains to be settled. Both M^2 and D^2 are context-free. From $d > 2$ on, one goes beyond the context-free languages. In the 3D case, for example, intersection with regular $a^+b^+c^+$ produces $a^nb^nc^n$, which is not context-free. Recent work by Salvati and Kanazawa [22,12] shows that M^3 is a 2-MCFG (Multiple Context Free Language [24]), but a non-wellnested one, which means M^3 is not a TAL (Tree Adjoining Language [10]). For higher dimensions, the challenge, for the MIX and the Dyck languages alike, would be to relate the size of their alphabet to the minimal degree (or fan-out) of the k-MCFG required for their recognition. Similarly, from the categorial point of view, one would like to see a hierarchy of stepwise generalizations of Lambek's Syntactic Calculus in correspondence with the rise in dimensionality.

This short note does not aim to answer these foundational questions in their full generality. We *do* hope to clarify the nature of the discontinuity exhibited by multidimensional Dyck languages by studying their correspondence with some well-understood combinatorial objects (sections §3 to §5). In §6 we present a typelogical anaysis of d-symbol Dyck languages in terms of an alignment of $(d-1)$ grammars for the two-symbol case. In §2, we set the stage with a comparison of the cardinalities of the Dyck and MIX languages.

[1] D^d should not be confused with the generalization that considers k different *pairs* of opening/closing parentheses $[_1,]_1, \ldots, [_k,]_k$.

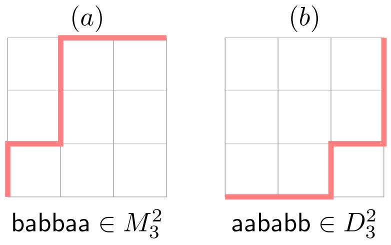
2 Counting Words

Table 1 compares the size of M_n^d and D_n^d for small values of $d \geq 2, n \geq 0$, and gives the identifiers of these number sequences in $OEIS$, the Online Encyclopedia of Integer Sequences.

Table 1. Cardinality of M_n^d and D_n^d (grayscale) for small values of $d \geq 2, n \geq 0$

$d \backslash n$	0	1	2	3	4	5	6	*OEIS*
2	1	2	6	20	70	252	924	*A000984*
	1	1	2	5	14	42	132	*A000108*
3	1	6	90	1680	34650	756756	17153136	*A006480*
	1	1	5	42	462	6006	87516	*A005789*
4	1	24	2520	369600	63063000	11732745024	2308743493056	*A008977*
	1	1	14	462	24024	1662804	140229804	*A005790*

A useful graphical representation highlighting the differences between M_n^d and D_n^d pictures the words of these languages as *monotone paths* along the edges of a d-dimensional grid, with sides measuring n unit steps: these are paths starting at the origin $(0, \ldots, 0)$ and ending at $(n, \ldots, n)$, and consisting entirely of forward steps along one of the dimensions. All such paths are legitimate for the MIX words. For Dyck words, the prefix constraint translates into the requirement that the coordinates of the points visited are weakly decreasing $x_1 \geq x_2 \geq \cdots \geq x_d$. In the two-dimensional case, this means a path can touch but not cross the diagonal. Compare the two words over the alphabet $\{\mathsf{a}, \mathsf{b}\}$ with letter multiplicity 3 in (2). Paths start at the lower left corner and end at the upper right corner. The letter a is interpreted as a step to the right, the letter b as a step up. The path in (b) respects the prefix constraint; the path in (a) does not.

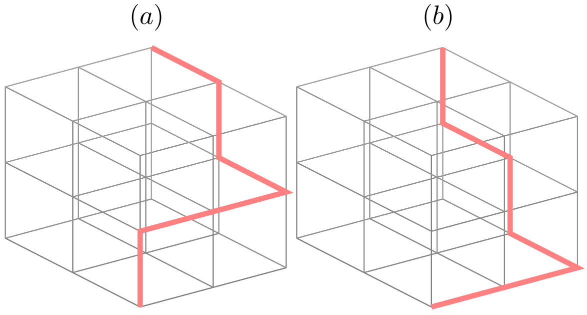


(2)

A comparable contrast for the 3-dimensional case is given in (3). All points visited by the path in (3b) have coordinates $x \geq y \geq z$:

$$(0,0,0) \xrightarrow{\mathsf{a}} (1,0,0) \xrightarrow{\mathsf{a}} (2,0,0) \xrightarrow{\mathsf{b}} (2,1,0) \xrightarrow{\mathsf{c}} (2,1,1) \xrightarrow{\mathsf{b}} (2,2,1) \xrightarrow{\mathsf{c}} (2,2,2)$$

whereas the path in (3b) violates the Dyck prefix constraint at the first step.



(a) caabcb $\in M_2^3$ (b) aabcbc $\in D_2^3$ (3)

The cardinality of the MIX languages is easily determined, see (4). There are $\binom{dn}{n}$ possible choices for the position of the n copies of the first alphabet symbol among dn candidate positions, then $\binom{(d-1)n}{n}$ possibilities for the n copies of the second alphabet symbol among the remaining $(d-1)n$ positions, and so on until one reaches the dth symbol for which no alternatives remain. In other words, (4) counts the number of distinct permutations of a multiset with d distinct elements each occurring with multiplicity n.

$$|M_n^d| \quad = \quad \prod_{k=1}^{d} \binom{kn}{n} \quad = \quad \frac{(dn)!}{(n!)^d} \tag{4}$$

The cardinality of the Dyck languages is given by the multidimensional Catalan numbers C_n^d which are given with the formula (5) in [7]. We will present an alternative formula with a direct combinatorial interpretation once we have discussed the connection between Dyck words and rectangular standard Young tableaux.

$$|D_n^d| = C_n^d = (dn)! \times \prod_{k=0}^{d-1} \frac{k!}{(n+k)!} \tag{5}$$

In the two-dimensional case, (5) gives the Catalan numbers $1, 1, 2, 5, 14, 42, 132, \ldots$ Note also that $|D_n^d| = |D_d^n|$; the triangular arrangement for D_n^d brings this out clearly. (Values for $n \leq 2, d \leq 2$. Multiplicity n increases along the $\searrow$ diagonal, dimensionality d along the $\swarrow$ diagonal.)

$$\begin{array}{ccccccccc}
 & & & & 2 & & & & \\
 & & & 5 & & 5 & & & \\
 & & 14 & & 42 & & 14 & & \\
 & 42 & & 462 & & 462 & & 42 & \\
132 & & 6006 & & 24024 & & 6006 & & 132
\end{array}$$

3 Dyck Words and Rectangular Standard Young Tableaux

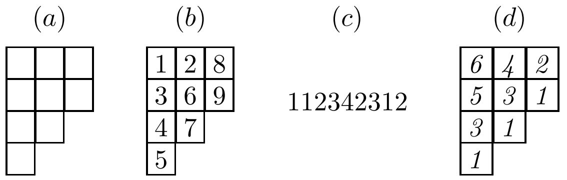
Young tableaux are rich combinatorial objects, widely used in representation theory of the symmetric and general linear groups and in algebraic geometry.

They are less known in computational linguistics, so let us start with some definitions, following [5]. Let $\lambda = (\lambda_1, \ldots, \lambda_k)$ be a partition of an integer n , i.e. a multiset of positive integers the sum of which is n, and let us list the k parts in weakly decreasing order. With such a partition, we associate a *Young diagram*: an arrangement of n boxes into left-aligned rows of length $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$. A *standard Young tableau* of shape λ is obtained by placing the integers 1 through n in these boxes in such a way that the entries are strictly increasing from left to right in the rows and from top to bottom in the columns. Henceforth, when we say tableau, we mean standard Young tableau.

Given a partition of n with a diagram of shape λ, the number of possible tableaux with that shape is given by the Frame-Robinson-Thrall [4] *hook length formula*, which is $n!$ divided by the product of the *hook lengths* of the n boxes; the hook length of a box is the number of boxes to the right in its row and below it in its column, plus one for the box itself. Given a tableau T, let T' be the tableau obtained from T by a reflection through the main diagonal, taking rows to columns and columns to rows. The diagrams for T and T', clearly, produce the same result for the hook length formula.

Given a standard Young tableau one can read off its *Yamanouchi word.* The Yamanouchi word for a tableau T of shape $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a word $w = w_1 \cdots w_n$ over a k-symbol alphabet $\{1, 2, \ldots, k\}$ such that w_i is the row that contains the integer i in T. Conversely, given a word $w = w_1 \cdots w_n$ over a k-symbol alphabet $\{1, 2, \ldots, k\}$ with the property that, reading w from left to right, there are never fewer letters i than letters $(i+1)$, we can recover a standard Young tableau with k rows.

We illustrate these definitions in (6) below. The partition $(3, 3, 2, 1)$ of the integer 9 has the Young diagram in (a). Distributing the integers $1, \ldots, 9$ over the nine boxes as in (b) yields a well-formed standard Young tableau with strictly increasing rows and columns. The Yamanouchi word for this tableau is given in (c). The fifth letter of the word is 4, because the integer 5 appears on the fourth row in the tableau. In (d), we give the hook lengths of the boxes of diagram (a). The hook length formula then says there are 168 standard Young tableaux of that shape: 9! divided by $(6 \cdot 4 \cdot 2 \cdot 5 \cdot 3 \cdot 1 \cdot 3 \cdot 1 \cdot 1)$.



(6)

The definitions above apply to standard Young tableaux in general. To obtain the multidimensional Dyck words we are interested in, we restrict to *rectangular* tableaux of shape $d \times n$ (d rows by n columns). These tableaux correspond to *balanced* Yamanouchi words of length dn— words in which the constituting letters appear with equal multiplicity. The prefix constraint of the generalized Dyck languages is captured by the tableau constraint that the box entries are strictly increasing from top to bottom in the columns; the restriction to rows of equal length corresponds to the letter multiplicity constraint on Dyck (and MIX) words. Note that, in accordance with the practice in formal linguistics, we use the alphabet $\mathsf{a} < \mathsf{b} < \mathsf{c} < \cdots$, rather than the alphabet $1 < 2 < 3 < \cdots$ of (6c).

For rectangular tableaux, the hook length formula takes on the simple form of (7) below.

$$\frac{dn!}{\prod_{k=1}^{n} k^{\overline{d}}} \tag{7}$$

(writing $n^{\overline{m}}$ for rising factorial powers $n(n+1)\cdots(n+m-1)$) which then counts the words of D_n^d for positive n. We saw in Table 1 that D_n^d and D_d^n have the same cardinality; indeed, the hook length formula for the D_n^d and for the D_d^n diagrams produces the same result. On the level of individual words, we have a bijection between words of D_n^d and words of D_d^n, reading them as the Yamanouchi words of a $d \times n$ tableau T and of the transposed $n \times d$ tableau T' respectively.

We illustrate with D_2^3, words over a three-letter alphabet with letter multiplicity 2. On the left are the hook lengths of the boxes in a diagram of shape $(3,2)$. By the hook length formula (7), there are $6!/(1 \cdot 2^2 \cdot 3^2 \cdot 4) = 720/144 = 5$ standard Young tableaux of that shape. They are listed in (a) through (e) together with their Yamanouchi words $y(T)$. Finally, $y(T')$ gives the words corresponding to the transposed tableaux T'. These are the five words of D_3^2.

$$\begin{array}{|c|c|}\hline \mathit{4} & \mathit{3} \\ \hline \mathit{3} & \mathit{2} \\ \hline \mathit{2} & \mathit{1} \\ \hline \end{array} \qquad \begin{array}{rccccc} & (a) & (b) & (c) & (d) & (e) \\ T: & \begin{array}{|c|c|}\hline 1&4\\\hline 2&5\\\hline 3&6\\\hline\end{array} & \begin{array}{|c|c|}\hline 1&3\\\hline 2&4\\\hline 5&6\\\hline\end{array} & \begin{array}{|c|c|}\hline 1&2\\\hline 3&5\\\hline 4&6\\\hline\end{array} & \begin{array}{|c|c|}\hline 1&3\\\hline 2&5\\\hline 4&6\\\hline\end{array} & \begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline 5&6\\\hline\end{array} \\ y(T) \in D_2^3: & \mathsf{abcabc} & \mathsf{ababcc} & \mathsf{aabcbc} & \mathsf{abacbc} & \mathsf{aabbcc} \\ y(T') \in D_3^2: & \mathsf{aaabbb} & \mathsf{aabbab} & \mathsf{abaabb} & \mathsf{aababb} & \mathsf{ababab} \end{array} \tag{8}$$

4 Dyck Words as Iterated Shuffles

Can we give an algorithm that generates the words of D_n^d in a unique way, and that provides the combinatorial *raison d'être* for the hook length formula for rectangular tableaux?

We approach this question in two steps. First we model the interleaving of n copies of the word $a_1 \cdots a_d$ as an iterated *shuffle* [6], where for strings w_1, w_2 the shuffle operation gives all possibilities of interleaving them in such a way that the order of the letters in w_1 and in w_2 is preserved. The operation takes its name from the familiar way of shuffling a deck of cards: the deck is cut in two packets, which are then merged by releasing the cards with the thumb of the left and of the right hand.

When shuffling a deck of cards, all cards are distinct. When interleaving copies of a Dyck word $a_1 \cdots a_d$, there are multiple occurrences of the letters, hence duplicate outcomes of their interleaving.

As a second step, then, we break the symmetry, and introduce a biased form of shuffling that generates the words of D^d_n without duplicates. We give a redundancy-free representation of these unique outputs in the form of a deterministic acyclic word automaton; the size (number of states) of this automaton directly correlates with the cardinality of D^d_n as given by the hook length formula (7).

Formally, shuffling is an operation $\odot : \Sigma^* \times \Sigma^* \to \mathcal{P}(\Sigma^*)$ defined as in (9) below (for $\alpha_1, \alpha_2 \in \Sigma$, and $u, u_1, u_2 \in \Sigma^*$):

$$\begin{array}{ll} (i) & u \odot \epsilon = \epsilon \odot u = \{u\} \\ (ii) & \alpha_1 u_1 \odot \alpha_2 u_2 = \{\alpha_1 w \mid w \in u_1 \odot \alpha_2 u_2\} \cup \{\alpha_2 w \mid w \in \alpha_1 u_1 \odot u_2\} \end{array} \tag{9}$$

The $\odot$ operation can be lifted to the level of languages in the usual way (we use the same symbol for the lifted operation): for languages L_1, L_2,

$$L_1 \odot L_2 = \bigcup_{\substack{w_1 \in L_1 \\ w_2 \in L_2}} w_1 \odot w_2$$

Analogous to the Kleene star, we have the notion of the *shuffle closure* of a language, $L^{(*)}$, defined in terms of shuffle iteration $L^{(n)}$:

$$L^{(0)} = \{\epsilon\}, \qquad L^{(n)} = L \odot L^{(n-1)}, \qquad L^{(*)} = \bigcup_{0 \leq i} L^{(i)}$$

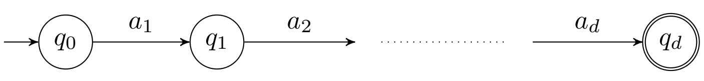
With these definitions in place, we see that D^d_n is the n-th shuffle interation of the language that has the concatenation of the d alphabet symbols in their lexicographic order as its single word. D^d is the shuffle closure of that language.

$$D^d_n = \{a_1 \cdots a_d\}^{(n)}, \qquad D^d = \{a_1 \cdots a_d\}^{(*)} \tag{10}$$

As noted, (10) correctly describes D^d_n as a *set* of words. But if, in the definition of shuffling in (9), we would substitute *lists* for sets and list concatenation for set union, we unfortunately would not obtain the desired procedure to produce the outcomes of shuffling in a repetition-free way.

To generate D^d_n without repetitions, we build a deterministic acyclic finite state automaton (DAFSA) representing the words of D^d_n in a redundancy-free way. The construction adapts a technique used by Warmuth and Haussler to prove that for any regular language R, its shuffle closure $R^{(*)}$ can be recognized in deterministic polynomial time [26, Theorem 5.1]. Informally, the idea is to decorate the states of a finite automaton with 'pebbles'; these pebbles control when and how often a transition for a given alphabet symbol can be made.

The construction of DAFSA(D^d_n) proceeds as follows. Let (11) below be the $(d + 1)$-state finite automaton for $\{a_1 \cdots a_d\}$, i.e. the singleton language that has the concatenation of the alphabet symbols in their lexicographic order as its single word.



(11)

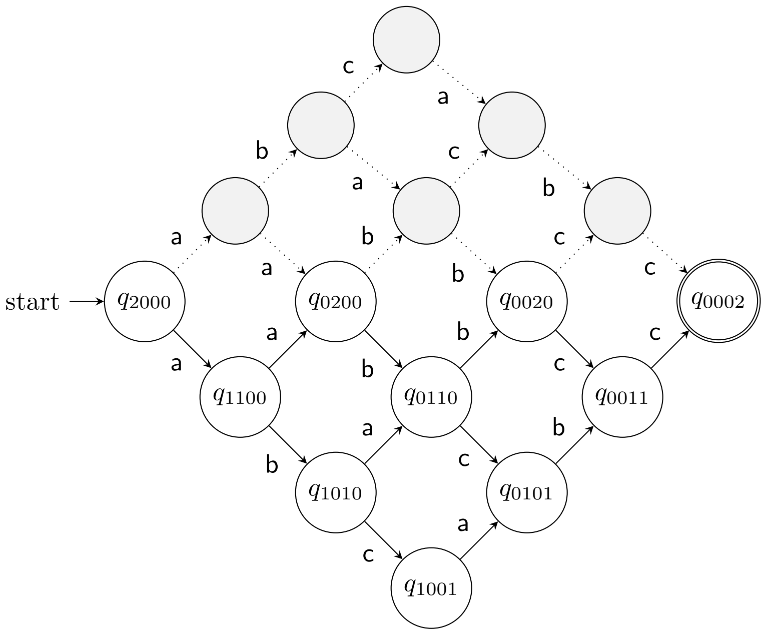
Then DAFSA$(D^d_n) = (Q, \Sigma, q_{start}, \delta, \{q_f\})$ is the following machine.

- ALPHABET $\Sigma = \{a_1, \dots, a_d\}$
- STATES Q: all $(d+1)$-tuples of non-negative integers $(p_0, \dots, p_d)$ such that $p_0 + \cdots + p_d = n$.
- INITIAL STATE q_{start}: the tuple that has $p_0 = n$ and $p_i = 0$ for $0 < i \leq d$
- FINAL STATE q_f: the tuple that has $p_d = n$ and $p_i = 0$ for $0 \leq i < d$
- TRANSITION FUNCTION δ: for states $q = (p_0, \dots, p_d)$ and $q' = (p'_0, \dots, p'_d)$ and alphabet symbols a_i $(1 \leq i \leq d)$, we have a transition $\delta(q, a_i) = q'$ whenever $p_{i-1} > 0$, $p'_{i-1} = p_{i-1} - 1$, $p'_i = p_i + 1$, and $p'_j = p_j$ for the other elements of the state tuples.

Note that the *size* (number of states) of the constructed DAFSA(D^d_n) is given by the number of possible distributions of n pebbles over the $(d+1)$ states of (11), i.e. the number of weak compositions of n in $d+1$ parts which is $\binom{n+d}{d}$.

Allowing the case of $n = 0$, there is one state, $q_{start} = q_f = (0, \dots, 0)$, and no transitions: this automaton accepts only the empty word.

The graph in (12) illustrates the construction with D^3_2, Dyck words over a three-letter alphabet with letter multiplicity 2. The diagram as a whole represents *all* ways of shuffling two copies of the word abc. An upward step releases a card from the left hand packet, a step downward from the right hand packet. There are twenty walks through this graph, enumerating the five words of D^3_2 in a non-unique way.

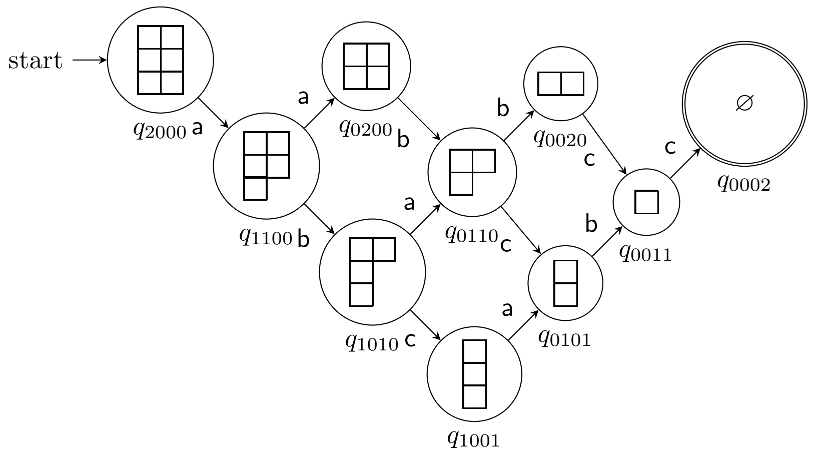


(12)

The state diagram for the constructed DAFSA(D^3_2) restricts the graph to the lower half, with the ten named states for the weak compositions of 2 in 4 parts.

One can picture this as a *biased* form of shuffling: the dealer releases a card/letter from the packet in the left hand (move upwards) only if in the lexicographic order it is strictly smaller than the competing card/letter in the right hand packet.

Under the above interpretation, a walk through the graph spells the Yamanouchi word of a rectangular tableau. The DAFSA(D^d_n) word graph allows another interpretation, discussed in [19], where a walk through the graph now provides a recipe for constructing a tableau (rather than its Yamanouchi word).



(13)

States, under this second interpretation, are Young diagrams for all possible partitions of integers 1 to $d \times n$ into maximally d parts of maximal size n. There is a directed edge from partition λ to λ' if λ' is the result of breaking off a *corner* of λ. A corner is a box that appears at the end of a row and a column. We read off the tableau from a walk in this graph as follows. Start at the d-part partition $\lambda_{start} = (n, \ldots, n)$ and finish at the empty partition $\lambda_{\varnothing}$. At each step, insert the highest integer from $d \times n$ not yet inserted into the corner that is deleted at that edge.

In (13), the Young diagrams corresponding to the states of DAFSA(D^3_2) have been added. As an example, take the path for the word abacbc $\in D^3_2$. The corresponding tableau is obtained by successively inserting 6 in the corner of the bottom row, 5 in the corner of the middle row, 4 in the corner of the first column, etc, with as final result the tableau

1	3
2	5
4	6

We have the ingredients in place now for a combinatorial interpretation of the hook-length formula (7) counting the rectangular standard Young tableaux of shape $d \times n$ (n positive) or equivalently the words of D^d_n they are in bijection with. It is convenient to model the situation with an n-handed card dealer. We can then represent the shuffle of n copies of the word $a_1 \cdots a_d$ as an n-dimensional diagram with sides of length d; steps forward in each direction are labeled with the alphabet symbols $a_1, \ldots, a_d$ in order, as in the two-dimensional case depicted in (12) above. Now consider the formula in (14) below.

$$\prod_{k=1}^{n} \left[\binom{kd}{d} \Big/ \binom{k+d-1}{d} \right] \tag{14}$$

The product of the numerators counts all outcomes of shuffling, including duplicates: in other words, all monotone paths from the origin $(0, \ldots, 0)$ to $(n, \ldots, n)$. The product of the denominators divides this by the size (number of states) of the successive $\text{DAFSA}(D^d_{(k-1)})$ word graphs. The result of the division counts the *distinct* outcomes of shuffling, i.e. gives the cardinality of D^d_n.

It is not hard to see that (14) is equivalent to (7). Indeed, expressing the binomial coefficients $\binom{n}{m}$ in factorial form $\frac{n^{\underline{m}}}{m!}$, with $n^{\underline{m}}$ for the falling factorial powers $n(n-1)\cdots(n-m+1)$, and noticing that $(k+d-1)^{\underline{d}} = k^{\overline{d}}$, we have

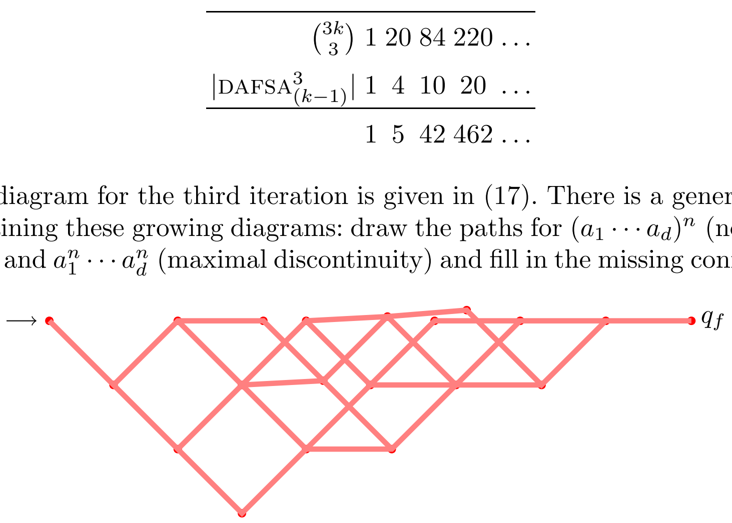
$$(14) \quad = \quad \prod_{k=1}^{n} \frac{kd^{\underline{d}}}{(k+d-1)^{\underline{d}}} \quad = \quad \prod_{k=1}^{n} \frac{kd^{\underline{d}}}{k^{\overline{d}}} \quad = \quad \frac{dn!}{\prod_{k=1}^{n} k^{\overline{d}}} \quad = \quad (7) \tag{15}$$

The table below illustrates with the first iterations of the shuffle of the word abc. For increasing k, the first row gives the number of possible positions for the letters of the kth copy of abc among $3k$ available positions. The second row gives the size of the word automaton for $(k-1)$ iterations (pebbles). In the bottom row, the result of the divisions: $(1 \times 20)/4 = 5$, $(5 \times 84)/10 = 42$, etc, i.e. *OEIS* sequence *A005789* of Table 1.

$k:$	1	2	3	4	...
$\binom{3k}{3}$	1	20	84	220	...
$\lvert\text{DAFSA}^3_{(k-1)}\rvert$	1	4	10	20	...
	1	5	42	462	...

(16)

The diagram for the third iteration is given in (17). There is a general recipe for obtaining these growing diagrams: draw the paths for $(a_1 \cdots a_d)^n$ (no discontinuity) and $a_1^n \cdots a_d^n$ (maximal discontinuity) and fill in the missing connections.



(17)

The word graph construction discussed here gives a non-redundant representation of the *finite* sets of elements of D^d_n, given symbol multiplicity n. For the *infinite* languages D^d, the transition function of the DAFSA word graph suggests a natural automaton model in the form of Greibach's [8] partially blind one-way multicounter machines. In these automata (see [9] for discussion) the stack holds a representation of a natural number, which can be incremented, decremented, or tested for zero. A counter is *blind* if it cannot be tested for zero except at the end of the computation as part of the acceptance condition. A *partially* blind counter cannot hold a negative value: the machine blocks on decrementing zero.

D^d then is accepted by a partially blind $(d-1)$-counter machine: on reading symbol a_1, the first counter is incremented; on reading symbol a_d, the last counter is decremented; on reading intermediate symbols a_i $(1 < i < d)$, the i-th counter is incremented and the $(i-1)$-th counter decremented. Failure on decrementing zero captures the prefix constraint of Equation (1) on Dyck words.

5 Dyck Words and Irreducible Webs for $\mathfrak{sl}_3$

In §3, we discussed the correspondence between multidimentional Dyck languages and rectangular standard Young tableaux for arbitrary dimension (number of rows) d. In the three-dimensional case, there is a further bijection between $3 \times n$ tableaux (hence: D^3_n Dyck words) and a class of combinatorial graphs ('webs') underlying Khovanov and Kuperberg's work [13] on combinatorial *spiders*, planar categories describing the invariant space of a tensor product of irreducible representations of a Lie algebra of given rank. The correspondence between webs for rank 3 and D^3_n Dyck words throws a surprising light on the nature of the discontinuity involved: the $\mathfrak{sl}_3$ webs are in fact *planar* graphs.

Our presentation essentially follows [25] except that where Tymoczko formulates the bijection in terms of tableaux, we rely on §3 and phrase the map directly in terms of the 3D Dyck words corresponding to these tableaux.

A web for the $\mathfrak{sl}_3$ spider category is a planar directed graph embedded in a disk where (i) internal vertices are of degree 3, boundary vertices are of degree 1; (ii) each vertex is either a source (all edges outgoing) or a sink (all edges incoming). A web is irreducible (or non-elliptic) if all internal faces have at least 6 sides. For the webs considered here all boundary vertices are sources.

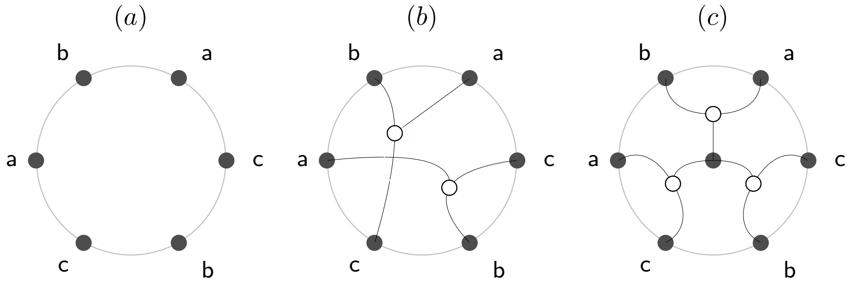
Let w be a word of D^3_n, i.e. the Yamanouchi word of a $3 \times n$ tableau. To build the web corresponding to w, take the following steps.

(i) Place the letters of $w = w_1, \ldots, w_{3n}$ counterclockwise on the boundary of a disk.
(ii) Create n internal sink vertices; each of these has incoming edges from boundary vertices labeled a, b and c. To determine how these triples are grouped, read w from left to right; for each b, find the a that precedes it most closely and that is not already linked, and direct the outgoing edges for this a and b pair to the same internal sink; then match c's and b's in the same way: for each c, find the b that precedes it most closely and that is not already co-linked with a c and connect them to the same internal sink.
(iii) Step (ii) may lead to crossing edges. If not, we are done. If there are crossing edges, restore planarity by replacing the degree 4 vertex v at their intersection by two degree 3 vertices: a sink v_1 and a source v_2; v_1 receives the

two incoming arrows that meet at the intersection point v, v_2 emits the two outgoing edges and has an arrow to v_1.

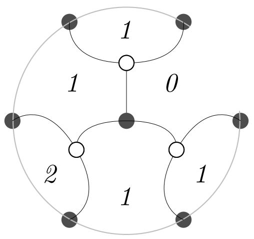
We refer the reader to [25] for proofs that the outcome of removing crossings is uniquely determined, and independent of the order of their removal, in the case of multiple crossings. Note that the grouping algorithm of step (ii) easily generalizes to alphabets of cardinality greater than three, a point we will return to in §6.

We illustrate with the word abacbc $\in D^3_2$. To avoid a clutter of arrows, we represent sources with dark, sinks with open circles. For step (a), start reading at the one o'clock vertex. Step (ii) here leads to the crossing shown in (b): the edge that connects the first c to the internal vertex that has an incoming edge from the closest b preceding it intersects an edge that was created in linking the second a and b to the same internal sink. Resolving the crossing with the recipe in (iii) leads to the planar web in (c).



(18)

For the opposite direction, from web to 3D Dyck word or tableau, we cut the disk, creating an unbounded face f_∞. All faces f are associated with a *depth*, $d(f)$, defined as the minimal number of edges one has to cross to reach the unbounded face, which then has $d(f_\infty) = 0$. Visit the boundary vertices counterclockwise and calculate $d(f_l) - d(f_r)$, the difference between the depth of the face to its left and to its right, looking in the direction of the outgoing edge. These differences have $\{1, 0, -1\}$ as possible outcomes. Interpret these as the three alphabet symbols of D^3_n: read 1 as a, 0 as b and -1 as c.

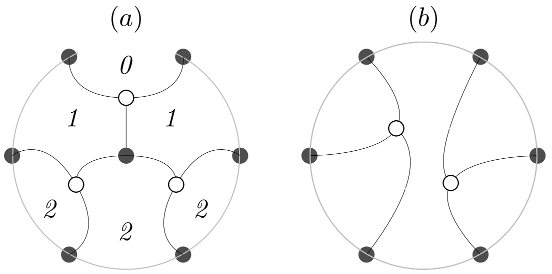


(19)

In (19), we illustrate with the web we obtained in (18)(c). We have cut the disk between the one o'clock and the three o'clock vertices, so that the one o'clock vertex is the starting point for the computation of the corresponding

Dyck word, as it was for (18)(c). Faces are annotated with the depth values resulting from that cut. Visiting the vertices counterclockwise, starting from one o'clock, and computing $d(f_l) - d(f_r)$, gives rise to the sequence $1, 0, 1, -1, 0, -1$, which translates as abacbc.

In general, for a word of length dn, there are dn possibilities of cutting the disk, with a maximum of dn distinct words to be read off from these cuts. In (20a) below, the cut is between the vertices at 11 o'clock and one o'clock, making the 11 o'clock vertex the starting point for the computation of the associated Dyck word. Another way of picturing this is as a counterclockwise *rotation* of the boundary vertices. The calculation of $d(f_l) - d(f_r)$ for the boundary vertices starting from 11 o'clock now gives rise to the sequence $1, 1, 0, 0, -1, -1$, which translates as the word aabbcc. We leave it to the reader to verify that the six possible cuts of the web of (19)/(20a) give rise to the two words $\mathsf{abacbc}, \mathsf{aabbcc}$. In the case of (20b), a web that doesn't involve any crossing dependencies, the six possible cuts give rise to three words: $\mathsf{abcabc}, \mathsf{ababcc}, \mathsf{aabcbc}$. The five words of D^3_2, then, are read off from the rotations of the two webs in (20).



(20)

In [25,20] it is shown that web rotation corresponds to Schützenberger's [23] *jeu de taquin* promotion operation on tableaux. The promotion operation (web rotation in the 3D case) induces an equivalence relation on multidimensional Dyck words that further reduces the apparent diversity of surface forms.

6 Grammars

After this exploration of the combinatorics of multidimensional Dyck languages, let us turn now to their analysis in terms of categorial grammar.

As we saw in the Introduction, D^2, the Dyck language over a two-symbol alphabet, is the language consisting of balanced strings of parentheses, one of the archetypical examples of a context-free language. A grammar with start symbol S and rules $S \longrightarrow \mathsf{a}\, S\, \mathsf{b}\, S$, $S \longrightarrow \epsilon$ generates D^2 with the empty word included in the language. A *lexicalized* version of this grammar excludes the empty word; each rule is required to introduce a single terminal element, the lexical anchor. The grammar with alphabet $\{\mathsf{a}, \mathsf{b}\}$, non-terminals $\{S, B\}$, start symbol S and the set of productions P of (21)

$$P: \qquad S \longrightarrow \mathsf{a}\, B \mid \mathsf{a}\, S\, B \mid \mathsf{a}\, B\, S \mid \mathsf{a}\, S\, B\, S \qquad ; \qquad B \longrightarrow \mathsf{b} \tag{21}$$

is in Greibach normal form (GNF), with the lexical anchor as the leftmost element in a rule's right-hand side. As is well-known, a grammar in GNF format can be directly transformed in a categorial type lexicon associating each terminal symbol with a finite number of types. For the resulting categorial grammar, atomic types are the non-terminals of the rewrite grammar; the only type constructor is '/' which we read left-associatively to economize on parentheses; the rules in P are turned into first-order type assignments for the lexical anchors according to the recipe in (22).

$$\textsc{lex}(a) = \{A_0/A_n/\cdots/A_1 \mid (A_0 \longrightarrow aA_1 \ldots A_n) \in P\} \tag{22}$$

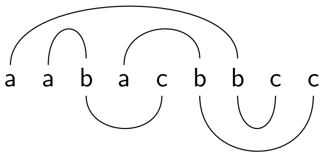
(or $A_0\, A_n^l \,\cdots\, A_1^l$, in the pregroup type format). In (23) one finds the lexicon for D^2 that results from the transformation of the rules in (21) into categorial types.

$$\textsc{lex}(\mathsf{a}) = \{S/B,\ S/B/S,\ S/S/B,\ S/S/B/S\} \qquad \textsc{lex}(\mathsf{b}) = \{B\} \tag{23}$$

For Dyck languages D^d with dimensionality $d > 2$, context-free expressivity is no longer sufficient, so one has to turn to generalizations of Lambek's Syntactic Calculus or its simplified pregroup incarnation. There is a choice of options here. The *multimodal* typelogical grammars of the Barcelona School (see Morrill and Valentín, this volume) combine residuated families of concatenation operations $(/, \otimes, \backslash)$ with *wrapping* operations $(\uparrow, \odot, \downarrow)$ for the combination of discontinuous expressions consisting of detached parts. Multimodal typelogical grammar, Utrecht style, extends the language of the Syntactic Calculus with *control features* $\Diamond, \Box$: a residuated pair of *unary* multiplicative operations licensing or blocking structural rules of inference. Bi-Lambek (or Lambek-Grishin) calculus (see [16]) adds duals for the $/, \otimes, \backslash$ operations: a multiplicative *sum* $\oplus$ together with left and right *difference* operations $\obslash, \oslash$; discontinuous dependencies then arise out of the *linear distributivity* principles relating the product and sum operations. Finally, one can move beyond the propositional language of the Syntactic Calculus, and handle the composition of discontinuous expressions in terms of structure variables and first-order quantification over them (as in [17] and Moot, this volume).

The recipient of this volume has shown a certain reluctance to consider any of these extended categorial formalisms (indeed, the title of [15] 'Should Pregroup Grammars be Adorned with Additional Operations?' can be taken as a *rhetorical* question), except for extensions of the formula language with meet and/or join operations, and for grammars based on products of pregroups (see for example Béchet, this volume). So let us here explore these strategies in the case of generalized Dyck languages of higher dimension.

In §5 we discussed a method for grouping the $(\mathsf{a},\mathsf{b},\mathsf{c})$ triples that are interleaved to make up the words of D^3 — a method that generalizes to alphabets of greater cardinality. The crucial observation about these d-symbol alphabets is that for each pair of successive symbols a_i, a_{i+1} $(1 \leq i < d)$, the projections $\pi_{a_i,a_{i+1}}(D^d)$ constitute D^2 Dyck languages over the two-letter alphabets $\{a_i, a_{i+1}\}$. As an example, take the word $w = \mathsf{aabacbbcc} \in D^3$. The projection $\pi_{\mathsf{a},\mathsf{b}}(w)$ removes all occurrences of c, producing $\mathsf{aababb} \in D^2$. Similarly, $\pi_{\mathsf{b},\mathsf{c}}(w) = \mathsf{bcbbcc}$ removes all occurrences of a, resulting in a D^2 Dyck word over the alphabet $\{\mathsf{b},\mathsf{c}\}$.



(24)

The D^d ($d > 2$) languages, then, can be obtained through the *intersection* of $d-1$ context-free languages, where each of the constituting languages respects well-nestedness for a pair of successive alphabet symbols and ignores intervening symbols from the rest of the alphabet.

We work this out for $d = 3$; the generalization to alphabets of greater size is straightforward. Consider $G = (\{\mathsf{a},\mathsf{b},\mathsf{c}\}, \{S, B, C\}, S, P)$ with the productions P in (25). It recognizes the same a,b patterns as the D^2 grammar with productions (21), but allows each occurrence of the letters a and b to be followed by an arbitrary number of c's.

$$P: \quad \begin{array}{l} S \longrightarrow \mathsf{a}\,B \mid \mathsf{a}\,S\,B \mid \mathsf{a}\,B\,S \mid \mathsf{a}\,S\,B\,S \\ \qquad \mid \mathsf{a}\,C\,B \mid \mathsf{a}\,C\,S\,B \mid \mathsf{a}\,C\,B\,S \mid \mathsf{a}\,C\,S\,B\,S \\ B \longrightarrow \mathsf{b} \mid \mathsf{b}\,C \\ C \longrightarrow \mathsf{c} \mid \mathsf{c}\,C \end{array} \tag{25}$$

Similarly, $G' = (\{\mathsf{a},\mathsf{b},\mathsf{c}\}, \{S', B', C'\}, S', P')$ with the productions P' in (26) recognizes balanced subsequences of a bracket pair $\{\mathsf{b},\mathsf{c}\}$, but allows occurrences of the letters b and c to be preceded by an arbitrary number of a's.

$$P': \quad \begin{array}{l} S' \longrightarrow \mathsf{b}\,C' \mid \mathsf{b}\,C'\,S' \mid \mathsf{b}\,S'\,C' \mid \mathsf{b}\,S'\,C'\,S' \\ \qquad \mid \mathsf{a}\,B'\,C' \mid \mathsf{a}\,B'\,C'\,S' \mid \mathsf{a}\,B'\,S'\,C' \mid \mathsf{a}\,B'\,S'\,C'\,S' \\ B' \longrightarrow \mathsf{b} \mid \mathsf{a}\,B' \\ C' \longrightarrow \mathsf{c} \mid \mathsf{a}\,C' \end{array} \tag{26}$$

We now have $D^3 = L(G) \cap L(G')$: the words of $L(G)$ have a and b with equal multiplicity, and at least as many a's as b's in every prefix; the words of $L(G')$ have an equal number of occurrences of b and c, and every prefix counts at least as many b's as c's.

The grammars G and G' are in Greibach normal form, so again, the transformation to categorial lexica is immediate. Intersection of context-free languages can be expressed categorially in a number of ways. One option, investigated in [11], is to enrich the type language with an intersective conjunction $\cap$, as originally proposed in [14]. Sequent rules for $\cap$ are given below.

$$\frac{\Gamma, A, \Gamma' \Rightarrow C}{\Gamma, A \cap B, \Gamma' \Rightarrow C}\cap L \qquad \frac{\Gamma, A, \Gamma' \Rightarrow C}{\Gamma, A \cap B, \Gamma' \Rightarrow C}\cap L \qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \cap B}\cap R$$

Under this approach, the lexicon for D^3 assigns type $A \cap B$ to an alphabet symbol a iff the lexicon obtained from G by the rule-to-type map (22) assigns A to a

and the lexicon obtained from G' assigns B; the distinguished symbol for the D^3 categorial grammar then becomes $S \cap S'$.

Alternatively, one can work in a product pregroup, where alphabet symbols are associated with a *pair* of pregroup types $\langle p, q \rangle$, with p obtained from G and q from G', and distinguished symbol $\langle S, S' \rangle$. Below in (27) is the computation that accepts aabacbbcc as a word of D^3.

$$\begin{array}{ccccccccc} \mathsf{a} & \mathsf{a} & \mathsf{b} & \mathsf{a} & \mathsf{c} & \mathsf{b} & \mathsf{b} & \mathsf{c} & \mathsf{c} \\ \begin{pmatrix} SB^lS^l \\ S'S'^{l}C'^{l}B'^{l} \end{pmatrix} & \begin{pmatrix} SS^lB^l \\ B'B'^{l} \end{pmatrix} & \begin{pmatrix} B \\ B' \end{pmatrix} & \begin{pmatrix} SB^lC^l \\ C'C'^{l} \end{pmatrix} & \begin{pmatrix} C \\ C' \end{pmatrix} & \begin{pmatrix} B \\ S'C'^{l}S'^{l} \end{pmatrix} & \begin{pmatrix} BC^l \\ S'C'^{l} \end{pmatrix} & \begin{pmatrix} CC^l \\ C' \end{pmatrix} & \begin{pmatrix} C \\ C' \end{pmatrix} \end{array} \tag{27}$$
$$= \begin{pmatrix} SB^lS^lSS^lB^lBSB^lC^lCBBC^lCC^lC \\ S'S'^{l}C'^{l}B'^{l}B'B'^{l}B'C'C'^{l}C'S'C'^{l}S'^{l}S'C'^{l}C'C' \end{pmatrix} \leq \begin{pmatrix} S \\ S' \end{pmatrix}$$

An even simpler pregroup construction obtains if we use the *multiplicative identity* to handle the letters that have to be ignored in checking balance for pairs of alphabet symbols. The type-assignment schema in (28) is the pregroup version of (23), handling successive pairs of alphabet symbols $\{a_i, a_{i+1}\}$ $(1 \leq i < d)$. The non-terminals B_i here introduce the 'closing bracket' a_{i+1} matching the 'opening bracket' a_i.

$$\textsc{lex}_i(a_i) = \{S_i\, B_i^l,\ S_i\, B_i^l\, S_i^l,\ S_i\, S_i^l\, B_i^l,\ S_i\, S_i^l\, B_i^l\, S_i^l\} \qquad \textsc{lex}_i(a_{i+1}) = \{B_i\} \tag{28}$$

The construction for the D^3 lexicon $\textsc{lex}_{1,2}$ now associates alphabet symbols with pairs of types. The first coordinate checks balance for the $\pi_{\mathsf{a},\mathsf{b}}$ projection; the letter that is to be ignored for this check (c) has the multiplicative identity in this position. The second coordinate, similarly, checks balance for the $\pi_{\mathsf{b},\mathsf{c}}$ projection; the letter a can be ignored in this case.

$$\begin{aligned} \textsc{lex}_{1,2}(\mathsf{a}) &= \{\langle p, 1\rangle \mid p \in \textsc{lex}_1(\mathsf{a})\} \\ \textsc{lex}_{1,2}(\mathsf{b}) &= \{\langle p, q\rangle \mid p \in \textsc{lex}_1(\mathsf{b}) \wedge q \in \textsc{lex}_2(\mathsf{b})\} \\ \textsc{lex}_{1,2}(\mathsf{c}) &= \{\langle 1, q\rangle \mid q \in \textsc{lex}_2(\mathsf{c})\} \end{aligned} \tag{29}$$

The computation for aabacbbcc $\in D^3$ now takes the form of (30) (distinguished type: $\langle S_1, S_2 \rangle$).

$$\begin{array}{ccccccccc} \mathsf{a} & \mathsf{a} & \mathsf{b} & \mathsf{a} & \mathsf{c} & \mathsf{b} & \mathsf{b} & \mathsf{c} & \mathsf{c} \\ \begin{pmatrix} S_1\, B_1^l\, S_1^l \\ 1 \end{pmatrix} & \begin{pmatrix} S_1\, S_1^l\, B_1^l \\ 1 \end{pmatrix} & \begin{pmatrix} B_1 \\ S_2\, S_2^l\, B_2^l \end{pmatrix} & \begin{pmatrix} S_1\, B_1^l \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ B_2 \end{pmatrix} & \begin{pmatrix} B_1 \\ S_2\, B_2^l\, S_2^l \end{pmatrix} & \begin{pmatrix} B_1 \\ S_2\, B_2^l \end{pmatrix} & \begin{pmatrix} 1 \\ B_2 \end{pmatrix} & \begin{pmatrix} 1 \\ B_2 \end{pmatrix} \end{array} \tag{30}$$

Again, the generalization to alphabets $\{a_1, \ldots, a_d\}$ of size $d > 3$ will be clear. Type assignments are $d - 1$ tuples. The symbol a_1 has 1 in all coordinates except the first, which has the $\textsc{lex}_1$ values for a_1 ; the symbol a_d has 1 in all coordinates except the last, which has the $\textsc{lex}_{d-1}$ value for a_d; intermediate letters a_j $(1 < j < d)$ serve simultaneously as closing bracket for a_{j-1} and as opening bracket for a_{j+1}: they have $\textsc{lex}_{j-1}$ and $\textsc{lex}_j$ values in the relevant coordinates, and 1 elsewhere.

Notice that with this construction, the first and the last alphabet symbols commute. In the case of the $\textsc{lex}_{1,2}$ type assignments, for example, we have $\langle p, 1\rangle \circ \langle 1, q\rangle = \langle p, q\rangle = \langle 1, q\rangle \circ \langle p, 1\rangle$, and indeed substrings ac of D^3 words commute: $u\,\mathsf{ac}\,v$ is in D^3 iff $u\,\mathsf{ca}\,v$ is.

7 Discussion, Conclusion

For discontinuous dependencies of the interleaving type discussed here, Bach [1,2] proposed action-at-a-distance function types $A /\!\!/ B$ ($B \backslash\!\!\backslash A$): functions producing a result of type A when provided with a B argument *somewhere* to their right (left), ignoring material that might intervene. The constructions of the previous section implement this idea of 'delayed' function application in a rather direct way. They apply to multidimensional Dyck languages with alphabets of arbitrary size, handling d-symbol alphabets by means of type tuples of size $(d-1)$ (or $(d-1)$ counter machines, if one prefers the automaton model discussed at the end of §4).

In §5 we briefly discussed web rotations and the *jeu de taquin* promotion operation on the $3 \times n$ tableaux corresponding to these webs. The promotion operation is not restricted to the three-dimensional case; it applies to tableaux of arbitrary dimension. This opens up the possibility of viewing Dyck words related by *jeu de taquin* promotion as different surface realisations that can be read off from a more abstract underlying representation. We leave this as a subject for further research.

References

1. Bach, E.: Discontinuous constituents in generalized categorial grammars. In: Proceedings of the 11th Annual Meeting of the North Eastern Linguistics Society, New York, pp. 1–12 (1981)
2. Bach, E.: Some generalizations of categorial grammar. In: Landman, F., Veltman, F. (eds.) Varieties of Formal Semantics: Proceedings of the Fourth Amsterdam Colloquium, September 1982. Groningen-Amsterdam Studies in Semantics, pp. 1–23. Walter de Gruyter (1984)
3. Berglund, M., Björklund, H., Högberg, J.: Recognizing shuffled languages. In: Dediu, A.-H., Inenaga, S., Martín-Vide, C. (eds.) LATA 2011. LNCS, vol. 6638, pp. 142–154. Springer, Heidelberg (2011)
4. Frame, J.S., de, G., Robinson, B., Thrall, R.M.: The hook graphs of the symmetric group. Canad. J. Math 6, 316–325 (1954)
5. Fulton, W.: Young Tableaux: With Applications to Representation Theory and Geometry. London Mathematical Society Student Texts, vol. 35. Cambridge University Press (1997)
6. Gischer, J.: Shuffle languages, petri nets, and context-sensitive grammars. Communications of the ACM 24(9), 597–605 (1981)
7. Gorska, K., Penson, K.A.: Multidimensional Catalan and related numbers as Hausdorff moments. ArXiv e-prints (April 2013)
8. Greibach, S.A.: Remarks on blind and partially blind one-way multicounter machines. Theoretical Computer Science 7(3), 311–324 (1978)

9. Hoogeboom, H., Engelfriet, J.: Pushdown automata. In: Martín-Vide, C., Mitrana, V., Păun, G. (eds.) Formal Languages and Applications. STUDFUZZ, vol. 148, pp. 117–138. Springer, Heidelberg (2004)
10. Joshi, A., Schabes, Y.: Tree-adjoining grammars. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 69–123. Springer, Heidelberg (1997)
11. Kanazawa, M.: The Lambek calculus enriched with additional connectives. Journal of Logic, Language and Information 1(2), 141–171 (1992)
12. Kanazawa, M., Salvati, S.: MIX is not a tree-adjoining language. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 666–674. Association for Computational Linguistics (2012)
13. Khovanov, M., Kuperberg, G.: Web bases for $\mathfrak{sl}_3$ are not dual canonical. Pacific J. Math 188(1), 129–153 (1999)
14. Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics XII, pp. 166–178. American Mathematical Society (1961)
15. Lambek, J.: Should pregroup grammars be adorned with additional operations? Studia Logica 87(2-3), 343–358 (2007)
16. Moortgat, M.: Symmetric categorial grammar. Journal of Philosophical Logic 38(6), 681–710 (2009)
17. Moot, R., Piazza, M.: Linguistic applications of first order intuitionistic linear logic. Journal of Logic, Language and Information 10(2), 211–232 (2001)
18. Nederhof, M.-J., Satta, G., Shieber, S.M.: Partially ordered multiset context-free grammars and ID/LP parsing. In: Proceedings of the Eighth International Workshop on Parsing Technologies, Nancy, France, pp. 171–182 (April 2003)
19. Nijenhuis, A., Wilf, H.S.: Combinatorial algorithms for computers and calculators. Computer science and applied mathematics. Academic Press, New York (1978); First ed. published in 1975 under title: Combinatorial algorithms
20. Petersen, T.K., Pylyavskyy, P., Rhoades, B.: Promotion and cyclic sieving via webs. Journal of Algebraic Combinatorics 30(1), 19–41 (2009)
21. Reape, M.: A Logical Treatment of Semi-free Word Order and Bounded Discontinuous Constituency. In: Proceedings of the Fourth Conference on European Chapter of the Association for Computational Linguistics, EACL 1989, Manchester, England, pp. 103–110. Association for Computational Linguistics, Stroudsburg (1989)
22. Salvati, S.: MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is solved by a third-order collapsible pushdown automaton. Rapport de recherche (February 2011)
23. Schützenberger, M.P.: Promotion des morphismes d'ensembles ordonnés. Discrete Mathematics 2(1), 73–94 (1972)
24. Seki, H., Matsumura, T., Fujii, M., Kasami, T.: On multiple context-free grammars. Theoretical Computer Science 88(2), 191–229 (1991)
25. Tymoczko, J.: A simple bijection between standard $3 \times n$ tableaux and irreducible webs for $\mathfrak{sl}_3$. Journal of Algebraic Combinatorics 35(4), 611–632 (2012)
26. Warmuth, M.K., Haussler, D.: On the complexity of iterated shuffle. Journal of Computer and System Sciences 28(3), 345–358 (1984)